



Bunchball Traditional APIs

Traditional API information is provided for legacy support. RESTful APIs should be used rather than the Traditional APIs documented here.

Contents

| | |
|---|----------|
| TRADITIONAL APIS..... | 5 |
| TRADITIONAL METHODS BY TYPE..... | 5 |
| USER.AWARDCHALLENGE METHOD | 7 |
| USER.CLIENTLOGACTION METHOD | 9 |
| USER.CREDITPOINTS METHOD | 12 |
| USER.CREDITPOINTS2 METHOD | 14 |
| USER.DEBITPOINTS METHOD | 16 |
| USER.DEBITPOINTS2 METHOD | 18 |
| USER.EXISTS METHOD | 20 |
| USER.GETACTIONHISTORY METHOD..... | 21 |
| USER.GETCHALLENGEPROGRESS METHOD..... | 23 |
| USER.GETGROUPS METHOD | 30 |
| USER.GETLEVEL METHOD | 31 |
| USER.GETNEXTCHALLENGE METHOD | 32 |
| USER.GETNEXTLEVEL METHOD | 36 |
| USER.GETOWNEDITEMS METHOD | 37 |
| USER.GETPENDINGNOTIFICATIONS METHOD | 40 |
| USER.GETPOINTSBALANCE METHOD | 44 |
| USER.GETPOINTSHISTORY METHOD..... | 47 |
| USER.GETPREFERENCE METHOD | 49 |
| USER.GETPREFERENCES METHOD..... | 51 |
| USER.GETRESPONSES METHOD..... | 52 |
| USER.JOINGROUP METHOD | 54 |
| USER.LEAVEALLGROUPS METHOD..... | 55 |
| USER.LEAVEGROUP METHOD | 56 |
| USER.LOGACTION METHOD..... | 58 |
| USER.LOGIN METHOD | 61 |
| USER.MODIFYUSERID METHOD..... | 65 |
| USER.PURCHASEITEM METHOD | 67 |
| USER.REMOVEPREFERENCE METHOD | 70 |
| USER.RESETLEVEL METHOD | 71 |
| USER.SELLBACKITEM METHOD | 73 |
| USER.SETLEVEL METHOD..... | 75 |
| USER.SETPREFERENCE METHOD | 76 |
| USER.SETPREFERENCES METHOD | 78 |
| USER.STORENOTIFICATIONS METHOD | 79 |
| USER.TRANSFERPOINTS METHOD..... | 81 |
| GROUP.GETCHALLENGEPROGRESS METHOD..... | 82 |
| GROUP.GETPREFERENCE METHOD | 86 |
| GROUP.GETPREFERENCES METHOD..... | 87 |
| GROUP.GETUSERS METHOD | 89 |
| GROUP.REMOVEPREFERENCE METHOD..... | 90 |
| GROUP.SETPREFERENCE METHOD | 91 |
| GROUP.SETPREFERENCES METHOD | 92 |

| | |
|--|-----|
| SITE.ADDUSERSTOGROUP METHOD..... | 93 |
| SITE.GETACTIONFEED METHOD | 95 |
| SITE.GETACTIONLEADERS METHOD | 99 |
| SITE.GETACTIONTARGETLEADERS METHOD..... | 103 |
| SITE.GETCATALOG METHOD | 105 |
| SITE.GETCATALOGITEM METHOD | 107 |
| SITE.GETCHALLENGELEADERS METHOD | 109 |
| SITE.GETGROUPACTIONLEADERS METHOD | 113 |
| SITE.GETGROUPPOINTSLEADERS METHOD | 115 |
| SITE.GETGROUPS METHOD | 117 |
| SITE.GETLEVELS METHOD | 119 |
| SITE.GETPOINTSLEADERS METHOD | 121 |
| SITE.GETRECENTACTIONS METHOD..... | 125 |
| SITE.GETRECENTCHALLENGES METHOD..... | 127 |
| SITE.GETRECENTUPDATES METHOD | 129 |
| SITE.REMOVEUSERSFROMGROUP METHOD | 130 |
| ADMIN.BACKUPSITECONFIG METHOD | 131 |
| ADMIN.CREATEACTION METHOD..... | 132 |
| ADMIN.CREATECATALOG METHOD | 135 |
| ADMIN.CREATECATALOGCATEGORY METHOD | 136 |
| ADMIN.CREATECATALOGITEM METHOD | 138 |
| ADMIN.CREATECHALLENGE METHOD..... | 141 |
| ADMIN.CREATEFOLDER METHOD..... | 148 |
| ADMIN.CREATEMETADATUM METHOD..... | 149 |
| ADMIN.CREATERULE METHOD | 154 |
| ADMIN.CREATESITEPOINTCATEGORY METHOD..... | 159 |
| ADMIN.DELETEMETADATUM METHOD | 162 |
| ADMIN.DELETEUSER METHOD | 163 |
| ADMIN.EXPORTLOCALETRANSLATIONS METHOD | 164 |
| ADMIN.GETACTIONTAGS METHOD | 166 |
| ADMIN.GETADMINLOG METHOD | 167 |
| ADMIN.GETADMINUSERS METHOD | 169 |
| ADMIN.GETALLCATALOGCONTENTS METHOD..... | 170 |
| ADMIN.GETCATALOG METHOD | 173 |
| ADMIN.GETCATALOGS METHOD..... | 176 |
| ADMIN.GETCHALLENGEDEPENDENCYTREE METHOD..... | 179 |
| ADMIN.GETCHALLENGES METHOD | 180 |
| ADMIN.GETFOLDERS METHOD | 185 |
| ADMIN.GETGROUPS METHOD..... | 186 |
| ADMIN.GETGROUPS2 METHOD | 188 |
| ADMIN.GETNOTIFICATIONS METHOD..... | 189 |
| ADMIN.GETNOTIFICATIONSTYLES METHOD | 192 |
| ADMIN.GETSITELEVELS METHOD..... | 195 |
| ADMIN.GETSITELOCALES METHOD | 196 |
| ADMIN.GETSITEPOINTCATEGORIES METHOD | 197 |
| ADMIN.LOGIN AND ADMIN.LOGINADMIN METHODS | 199 |
| ADMIN.LOGOUTADMIN METHOD | 201 |

| | |
|--|------------|
| ADMIN.REMOVECHALLENGEFROMUSER METHOD | 202 |
| ADMIN.RESTORESITECONFIG METHOD | 203 |
| ADMIN.SETPOINTSBALANCE METHOD | 204 |
| ADMIN.UPDATECHALLENGE METHOD | 206 |
| ADMIN.UPDATEMETADATUM METHOD | 213 |
| ADMIN.UPDATERULE METHOD..... | 218 |
| ADMIN.UPDATESITEPOINTCATEGORY METHOD..... | 221 |
| BATCH.RUN METHOD..... | 224 |
| HELLO METHOD | 225 |
| SERVER.GETERRORCODES METHOD | 226 |
| SERVER.STATUS METHOD..... | 226 |
| NITRO ERROR CODES..... | 227 |
| ARGUMENT TYPES..... | 234 |
| OTHER PARAMETER CONSTRAINTS AND FLAGS..... | 235 |
| DISCLAIMER..... | 237 |

Traditional APIs

Nitro's traditional APIs are HTTPS-based requests that can be called from any programming language supporting HTTPS. Traditional APIs have been replaced with RESTful APIs. As you manage your program, [RESTful APIs](#) should be used before traditional APIs. Use Traditional APIs only when a REST version is unavailable.


Bunchball strongly recommends using JSON3 for all Traditional API calls.

Traditional Methods By Type


| User | Site | Group | Admin* | Utility |
|---|---|--|--|--------------------------------------|
| user.awardChallenge | site.addUsersToGroup | group.getChallengeProgress | admin.createAction | batch.run |
| user.clientLogAction | site.getActionFeed | group.getPreference | admin.createCatalog | hello |
| user.creditPoints | site.getActionLeaders | group.getPreferences | admin.createCatalogCategory | server.getErrorCodes |
| user.creditPoints2 | site.getActionTargetLeaders | group.getUsers | admin.createCatalogItem | |
| user.debitPoints | site.getCatalog | group.removePreference | admin.createChallenge | |
| user.debitPoints2 | site.getCatalogItem | group.setPreference | admin.createFolder | |
| user.exists | site.getChallengeLeaders | group.setPreferences | admin.createMetadata | |
| user.getActionHistory | site.getGroupActionLeaders | | admin.createRule | |
| user.getChallengeProgress | site.getGroupPointsLeaders | | admin.createSitePointCategory | |
| user.getGroups | site.getGroups | | admin.deleteMetadata | |
| user.getLevel | site.getLevels | | admin.deleteUser | |
| user.getNextChallenge | site.getPointsLeaders | | admin.exportLocaleTranslations | |
| user.getNextLevel | site.getRecentActions | | admin.getActionTags | |

| User | Site | Group | Admin* | Utility |
|--|---|-------|--|---------|
| user.getOwnedItems | site.getRecentChallenges | | admin.getAdminLog | |
| user.getPendingNotifications | site.getRecentUpdates | | admin.getAdminUsers | |
| user.getPointsBalance | site.removeUsersFromGroup | | admin.getAllCatalogContents | |
| user.getPointsHistory | | | admin.getCatalog | |
| user.getPreference | | | admin.getCatalogs | |
| user.getPreferences | | | admin.getChallengeDependencyTree | |
| user.getResponses | | | admin.getChallenges | |
| user.joinGroup | | | admin.getFolders | |
| user.leaveAllGroups | | | admin.getGroups | |
| user.leaveGroup | | | admin.getGroups2 | |
| user.logAction | | | admin.getNotifications | |
| user.login | | | admin.getNotificationStyles | |
| user.modifyUserId | | | admin.getSiteLevels | |
| user.purchaseItem | | | admin.getSiteLocales | |
| user.removePreference | | | admin.getSitePointCategories | |
| user.resetLevel | | | admin.login | |
| user.sellbackItem | | | admin.logoutAdmin | |
| user.setLevel | | | admin.removeChallengeFromUser | |
| user.setPreference | | | admin.setPointsBalance | |
| user.setPreferences | | | admin.updateChallenge | |
| user.storeNotifications | | | admin.updateMetadatum | |
| user.transferPoints | | | admin.updateRule | |

| User | Site | Group | Admin* | Utility |
|------|------|-------|---|---------|
| | | | admin.updateSitePoint Category | |

*  Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

user.awardChallenge Method

 As you manage your program, [RESTFul APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Instantly awards a challenge to a user by bypassing all rules required for the challenge.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|---------|---------------|------------------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| challenge | YES | String | None | 0 - Unlimited chars | The challenge name. |
| userId | YES | String | None | 1 - 255 chars | The user ID of the user to be awarded. |
| value | No | Integer | 1 | 0 - 2 ³¹ -1 | Multiplies the challenge point reward by the factor if the challenge is set to apply multipliers. Mimics the multiplier / action value behavior seen when typically completing a challenge. Only applies to challenges that have the multiplier enabled (Nitro Studio > Reward & Motivate > Challenges > challenge name > Rewards tab > Points). A multiplier is only available if there is one rule in the challenge |

| Name | Required | Type* | Default Value | Range | Comments |
|------|----------|-------|---------------|-------|---|
| | | | | | and that rule has an Action goal of Action value. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------------------|---|
| actionPhrase | The challenge's action phrase. |
| catalogItems | The catalog item awarded for completing the challenge: <ul style="list-style-type: none"> ▪ description - The item's description. ▪ name - The item name. |
| description | The challenge's description. |
| fullUrl | The URL to the full size image awarded as a badge. |
| name | The challenge's name. |
| points | The number of points awarded. |
| pointCategory | The category the points were awarded from. |
| repeatable | Indicates if the challenge is repeatable: <ul style="list-style-type: none"> ▪ 0 - Not repeatable ▪ 1 - Repeatable |
| tags | The tags associated with the challenge. |
| thumbUrl | The URL to the thumbnail image awarded as a badge. |
| value | The challenge's multiplier value. |

Sample Request

```
method=user.awardChallenge&asyncToken=&challenge=ReceiveKudos&userId=4000&value=1
```


Sample Response

```
{ "response": { "achievements": { "challengesAchieved": {
"challengeAchieved": [ { "catalogItems": { "catalogItem": [ {
"description": "Company tee", "name": "Company T-shirt" } ] },
"pointsRewards": {}, "points": 0, "pointCategory": "", "thumbUrl":
"C:\\Images\\Store\\CompanyTee.png", "fullUrl":
"C:\\Images\\Store\\CompanyTee.png", "repeatable": 0, "description":
```



```
"Employee's work recognized by a customer", "name": "ReceiveKudos",
"actionPhrase": "{username} was recognized for a job well done!" } ]
}, "points": 0, "tags": "", "value": 1 } }, "responseType": {},
"isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

user.clientLogAction Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

This method is identical to [user.logAction](#) in all ways except it can ONLY log actions for tags with low security enabled. It allows you to bypass medium or high security when the Low security option is enabled on the [action](#) (**Integrate > Actions > New/Edit > Advanced > Low security** checkbox).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|------------------------------|----------|----------------------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| apiKey | No | String | None | 0 - 255 chars | The API key for the site. |
| competitionInstanceId | No | Long | None | N/A | If set, records the value for this user in this competition instance. The competitionInstanceId is returned by user.createCompetition or user.getCompetitionProgress. |
| eventID | No | String | None | 0 - 255 chars | Only used in user.getAchievements. Used for transactional ID for support and troubleshooting. Can also be used to pass in the same ID multiple times to avoid redundant data. |
| metadata | No | Comma-separated list | None | 0 - 10 items | A comma-separated list of metadata tags (key value pairs separated by a colon). Allows you to populate other |

| Name | Required | Type* | Default Value | Range | Comments |
|----------------------|----------|----------------------|---------------|--|---|
| | | | | 0 - 255 chars per item | contextual information around the action taking place. Alphanumeric only; avoid special characters. Metadata is a string comparison and must be an exact match to what's configured in the challenge. Numbers in metadata cannot be treated as numbers, just strings. |
| newsfeed | No | String | None | 0 - 2000 chars | Exact text for newsfeed entries. Supports Nitro tokens . Return URL-encoded HTML for the action in the response of site.getActionFeed . Note: Not recommended for use due to the noise it creates in your program. |
| storeResponse | No | Boolean | false | true/false | When true, stores the response for later lookup via user.getResponses . If using <code>asyncToken</code> , set to true. Can be used to track something specific such as finding what challenge completed for a specific action rather than a list of all completed challenges. |
| tags | YES | Comma-separated list | None | 0 - 10 items 0 - 255 chars per item | Used to pass the action name. A comma-separated list of actions and contextual tags to log. Mainly used to power leaderboards. The parameter is case-sensitive and includes both the action and contextual tag. For example, <code>method=logAction&tags=action_tag,contextual1,contextual2,contextual3</code> . The parameter supports up to a total of 10 items (10 contextual tags, or 10 action tags, or 1 action tag and 9 contextual tags). However, leaderboards and reporting can only support up to 5 contextual tags. |

| Name | Required | Type* | Default Value | Range | Comments |
|------------------|----------|---------|---------------|------------------------|---|
| target | No | String | None | 0 - 255 chars | Associates the action with a target. Often used for API calls that allow you to filter by target. Note: Not recommended for use due to low program value. |
| timestamp | No | Integer | 0 | 0 - 2 ³¹ -1 | The current date/time as received by Bunchball. When blank, timestamp defaults to the current date/time. If specified, allows you to back-date the date/time, which impacts reporting and which challenges/rules are triggered. |
| userId | No | String | None | 0 - 255 chars | User to log the action for. By default, the action is logged for the authenticated user. |
| value | No | Integer | 0 | 0 - 2 ³¹ -1 | The action value to save. Must be a number. The value enables you to create leaderboards, see if the user has met certain criteria, and do running total calculations. Can be used for simple number checks like greater than, less than, and with points multiplier . For example, if this action represents inviting friends, you could pass in a value of 3 to indicate that the user invited 3 friends. If the action represents watching a video, you could pass in the time in seconds that the user watched the video. If the action represents taking a quiz, you could pass in the score. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------------|--|
| isError | <ul style="list-style-type: none"> true - The action(s) failed to log. |

| Response | Description |
|----------|---|
| | <ul style="list-style-type: none"> false - The action(s) logged successfully. |

Sample Request

```
method=user.clientLogAction&asyncToken=&apiKey=&target=&tags=HarassmentTraining&userId=4000&value=2&storeResponse=false&competitionInstanceId=&newsfeed=&metadata=
```


Sample Response

```
{ "response": {}, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

Sample Response - Not a Low Security Action Tag

```
{ "response": { "error": { "code": "241", "message": "One of the specified action tags was not low security" } }, "responseType": {}, "isError": true, "httpError": 0, "httpsStatusCode": 400 }
```

user.creditPoints Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Credits points to a user account and returns the point balance. If the specified user doesn't exist, it is created.

For time based leaderboards, credits are rewarded for the current leaderboard only. You cannot backdate points credits.

Note: Use [user.creditPoints2](#) to credit points and return the points history item.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |

| Name | Required | Type* | Default Value | Range | Comments |
|------------------------------|----------|----------------------|---------------|--|--|
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| points | YES | Integer | 0 | 0 - 2 ³¹ -1 | The total amount to credit. Accepts only positive numbers. |
| pointCategory | No | String | None | 0 - 255 chars | The category to credit to. Defaults to the default point category defined in Nitro Studio (Configuration > Site Settings > Point Categories > new/edit > Set as default category field). |
| tags | No | Comma-separated list | None | 0 - 10 items 0 - 255 chars per item | A comma-separated list of tags to associate with the credit. |
| updateLifetimeBalance | No | Boolean | true | true/false | Whether to update lifetime points balance for the user in addition to the current points balance. |
| userId | No | String | None | 1 - 255 chars | The user's unique ID for the site. Must be alphanumeric and no special characters (can be hashed). It's important to use a commonly known ID. If not specified, defaults to the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------------------|--|
| points | The user's current point balance. |
| pointCategory | The category the points were awarded to. |
| userId | The user who was awarded the points. |


Sample Request

```
method=user.creditPoints&asyncToken=&tags=null&pointCategory=Points&serId=2003&points=50&updateLifetimeBalance=true
```

Sample Response

```
{ "response": { "user": { "userId": "2003", "pointCategory": "Points", "points": 150 } }, "responseType": {}, "isError": false, "httpError": 0, "httpStatusCode": 200 }
```

user.creditPoints2 Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Credits points to a user account and returns the points history item. If the specified user doesn't exist, it is created.

For time based leaderboards, credits are rewarded for the current leaderboard only. You cannot backdate points credits.

Note: Use [user.creditPoints](#) to credit points and return the user's balance.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|----------------------|----------|---------|---------------|------------------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| points | YES | Integer | 0 | 0 - 2 ³¹ -1 | The total amount to credit. Accepts only positive numbers. |
| pointCategory | No | String | None | 0 - 255 chars | The category to credit to. Defaults to the default point category defined in Nitro Studio (Configuration > Site |

| Name | Required | Type* | Default Value | Range | Comments |
|------------------------------|----------|----------------------|---------------|--|---|
| | | | | | Settings > Point Categories > new/edit > Set as default category field). |
| pointCategoryId | No | String | None | 1 - 255 chars | The point category ID to query for. |
| tags | No | Comma-separated list | None | 0 - 10 items 0 - 255 chars per item | A comma-separated list of tags to associate with the credit. |
| timestamp | No | Integer | 0 | 0 - 2 ³¹ -1 | The action date to filter the results by. A UNIX timestamp. |
| updateLifetimeBalance | No | Boolean | true | true/false | Whether to update lifetime points balance for the user in addition to the current points balance. |
| userId | No | String | None | 1 - 255 chars | The user's unique ID for the site. Must be alphanumeric and no special characters (can be hashed). It's important to use a commonly known ID. If not specified, defaults to the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------------|--|
| action | The type of point transaction. |
| amount | The number of points awarded. |
| id | The ID of the points history record. |
| points | The user's current point balance. |
| pointCategory | The point category the points were awarded to. |
| pointCategoryId | The point category ID. |

| Response | Description |
|-------------------------------|--|
| pointCategoryIsDefault | Indicates if the point category for the points history transaction is the default for the site. Returns true or false. |
| tags | The actions or challenges that are associated with the point award. |
| ts | The time when the point transaction occurred (UNIX timestamp). |
| updateLifetimeBalance | Indicates if the lifetime balance was updated with the points history transaction. Returns true or false. |
| userId | The user who was awarded the points. |


Sample Request

```
method=user.creditPoints2&asyncToken=&tags=null&pointCategory=Points&
userId=judi.ingram&points=25&updateLifetimeBalance=true
```

Sample Response

```
{ "response": { "pointsHistoryRecord": { "pointsHistoryItem": [ {
"id": 11263, "ts": 1529079423, "amount": 25, "action": "credit",
"tags": "", "updateLifetimeBalance": true, "pointCategoryId": 1,
"pointCategory": "Points", "pointCategoryIsDefault": true } ] } },
"responseType": {}, "isError": false, "httpError": 0,
"httpStatusCode": 200 }
```

user.debitPoints Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Debits points from a user account and returns the user's balance. If the specified user doesn't exist, it is created.

Points removal applies to the current point in time, and may affect time based leaderboards negatively. You cannot backdate points debits. When debiting points for a specific timeframe, removing more points than a user has earned in that timeframe causes negative numbers to display in the leaderboard.

Note: Use [user.debitPoints2](#) to debit points and return the points history item.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-----------------------------|----------|----------------------|---------------|---|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| debitLifetimeBalance | No | Boolean | false | true/false Truncated | If true, debits points from a credits leaderboard and decreases lifetime points. |
| points | YES | Integer | 0 | 0 - 2 ³¹ -1 | The total amount to debit. Positive numbers only. The balance cannot go below 0. |
| pointCategory | No | String | None | 0 - 255 chars | The category to debit from. Defaults to the default point category defined in Nitro Studio (Configuration > Site Settings > Point Categories > new/edit > Set as default category field). |
| tags | No | Comma-separated list | None | 0 - 10 items 0 - 255 chars per item | A comma-separated list of tags to associate this debit with. |
| userId | No | String | None | 1 - 255 chars | The user's unique ID for the site. Must be alphanumeric and no special characters (can be hashed). It's important to use a commonly known ID. If not specified, defaults to the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------------------|--|
| points | The user's current point balance |
| pointCategory | The category the points were debited from. |
| userId | The user who was debited the points. |


Sample Request

```
method=user.debitPoints&asyncToken=&tags=null&pointCategory=Points&userId=4000&points=2&debitLifetimeBalance=false
```

Sample Response

```
{ "response": { "user": { "userId": "4000", "pointCategory": "Points", "points": 3357 } }, "responseType": {}, "isError": false, "httpError": 0, "httpStatusCode": 200 }
```

user.debitPoints2 Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Debits points from a user account and returns the points history item. If the specified user doesn't exist, it is created.

Points removal applies to the current point in time, and may affect time based leaderboards negatively. You cannot backdate points debits. When debiting points for a specific timeframe, removing more points than a user has earned in that timeframe causes negative numbers to display in the leaderboard.

Note: Use [user.debitPoints](#) to debit points and return the user's balance.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|--|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the |

| Name | Required | Type* | Default Value | Range | Comments |
|-----------------------------|----------|----------------------|---------------|---|---|
| | | | | | response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| debitLifetimeBalance | No | Boolean | false | true/false Truncated | If true, debits points from a credits leaderboard and decreases lifetime points. |
| points | YES | Integer | 0 | 0 - 2 ³¹ -1 | The total amount to debit. Positive numbers only. The balance cannot go below 0. |
| pointCategory | No | String | None | 0 - 255 chars | The category to debit from. Defaults to the default point category defined in Nitro Studio (Configuration > Site Settings > Point Categories > new/edit > Set as default category field). |
| pointCategoryId | No | String | None | 1 255 chars | The point category ID to query for. |
| tags | No | Comma-separated list | None | 0 - 10 items 0 - 255 chars per item | A comma-separated list of tags to associate this debit with. |
| timestamp | No | Integer | 0 | 0 - 2 ³¹ -1 | The action date to filter the results by. A UNIX timestamp. |
| userId | No | String | None | 1 - 255 chars | The user's unique ID for the site. Must be alphanumeric and no special characters (can be hashed). It's important to use a commonly known ID. If not specified, defaults to the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-------------------------------|--|
| action | The type of point transaction. |
| amount | The number of points debited. |
| id | The ID of the points history record. |
| points | The user's current point balance |
| pointCategory | The category the points were debited from. |
| pointCategoryId | The point category ID. |
| pointCategoryIsDefault | Indicates if the point category for the points history transaction is the default for the site. Returns true or false. |
| tags | The actions or challenges that are associated with the point debit. |
| ts | The time when the point transaction occurred (UNIX timestamp). |
| updateLifetimeBalance | Indicates if the lifetime balance was updated with the points history transaction. Returns true or false. |
| userId | The user who was debited the points. |


Sample Request

```
method=user.debitPoints2&asyncToken=&tags=null&pointCategory=Points&userId=judi.ingram&points=10&debitLifetimeBalance=false
```

Sample Response

```
{ "response": { "pointsHistoryRecord": { "pointsHistoryItem": [ {
  "id": 11283, "ts": 1529096164, "amount": -10, "action": "debit",
  "tags": "", "updateLifetimeBalance": false, "pointCategoryId": 1,
  "pointCategory": "Points", "pointCategoryIsDefault": true } ] } },
  "responseType": {}, "isError": false, "httpError": 0,
  "httpStatusCode": 200 }
```

user.exists Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Checks if a user exists. Can be used to minimize API traffic by determining if the user is already known and has data in Nitro.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| userId | YES | String | None | 1 - 255 chars | The ID of the user to check. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|---------------|--|
| exists | Indicates if the user exists. Returns true or false. |


Sample Request

```
method=user.exists&asyncToken=&userId=3000
```

Sample Response

```
{ "response": { "exists": true }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

user.getActionHistory Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Returns the logged action history for the specified user within a given range, most recent activity first. If the specified user doesn't exist, the function fails. Can specify tags to filter on, which can be action tags (the names of actions) or regular tags.

This is the only API call that returns metadata for an action. Used mainly for troubleshooting around actions.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|---------------------|----------|----------------------|---------------|--|--|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| end | No | Integer | Now() | 0 - 2 ³¹ -1 | End date to use. A UNIX timestamp. |
| returnCount | No | Integer | 10 | 1 - 100 | The number of records to return. |
| start | No | Integer | 0 | 0 - 2 ³¹ -1 | Start date to use. A UNIX timestamp. Note: When filtering using start and end dates AND the tags parameter, you must use a minimum of a 7 day duration between the start and end date to return any results. Any smaller duration returns no data, even though action records exist. When filtering by date without the tags parameter, then no minimum duration is necessary. |
| startIndex | No | Integer | 0 | 0 - 2 ³¹ -1 | 0-based index specifying which row from the results to return data from. Can be used to page through the result set. |
| tags | No | Comma-separated list | None | 0 - 5 items Truncated 0 - 255 chars per item | A comma-separated list of tags (contextual or action tags) to filter on. |
| tagsOperator | No | Operator | OR | AND/OR | Actions that have any of the tags are considered. If set to AND, then only actions that have all of the tags are considered. Important for troubleshooting leaderboard accuracy. |
| trackRequest | No | Boolean | false | true/false | When true, stores the contextual tags for later retrieval. |

| Name | Required | Type* | Default Value | Range | Comments |
|--------|----------|--------|---------------|---------------|--|
| userId | No | String | None | 0 - 255 chars | User to get responses for. By default, this is the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------|---|
| tags | The tags associated with the action. |
| ts | The time the action was completed (UNIX timestamp). |
| value | The action's value. |

Sample Request

```
method=user.getActionHistory&asyncToken=&end=&tags=null&returnCount=10&userId=4000&start=&startIndex=0&tagsOperator=OR3
```

Sample Response

```
{ "response": { "actionHistoryRecord": { "actionHistoryItem": [ {
"ts": 1519762875, "value": 2, "tags": "HarassmentTraining" }, { "ts":
1489788550, "value": 0, "tags": "Review_Document" }, { "ts":
1483022894, "value": 0, "tags": "app_login" }, { "ts": 1474317379,
"value": 0, "tags": "HarassmentTest" } ] }, "returnCount": 4,
"startIndex": 0 }, "responseType": {}, "isError": false, "httpError":
0, "httpsStatusCode": 200 }
```

user.getChallengeProgress Method



As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

One of the main API calls that can be used to get a list of a user's completed, in progress, and eligible [challenges](#). Gets the current progress of the user in a specific challenge or all challenges. Only returns active challenges, unless an inactive or deleted challenge was fully completed by the user before it was made inactive or deleted, in which case it is also returned. Challenges are returned in [challenge/folder order](#).

✓ **Best Practice** For performance reasons, you should avoid a large number of active challenges configured in the system. And in cases where this cannot be avoided, use filtering input arguments listed in the following table.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|----------------------|----------|---------|---------------|------------------------|--|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. Mainly used for tracking specific responses associated with a request for large data volumes. Not used for filtering. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| challengeName | No | String | None | 0 - 255 chars | Get progress for this challenge only. Defaults to none, which returns all challenges. |
| folder | No | String | None | 0 - 255 chars | If set, only retrieve challenges in the specified folder. Performance increases dramatically when filtering on folders. |
| isFeatured | No | Boolean | None | true/false | If true, returns only challenges that have the featured flag set to true. If not specified, all challenges are returned. |
| returnCount | No | Integer | 0 | 0 - 2 ³¹ -1 | The number of records to return. Defaults to 0, |

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------------------|----------|---------|---------------|---|--|
| | | | | | which returns all results after startIndex. |
| showActiveOnly | No | Boolean | false | true/false | If true, returns only active challenges. When false, returns inactive and draft challenges. Bunchball highly recommends setting this to true. |
| showCanAchieveChallenge | No | Boolean | false | true/false Truncated | Indicates whether the user can achieve the challenge in the response. Bunchball highly recommends setting this to true. A true value excludes challenges for a user where they do not meet the challenge prerequisites . |
| showOnlyTrophies | No | Boolean | true | true/false Truncated | Returns only challenges with badges . |
| showRulesProgress | No | Boolean | true | true/false Truncated | Show the user's progress toward each of the rules. |
| showServiceTypeInfo | No | Boolean | false | true/false | If false, can boost performance. |
| startIndex | No | Integer | 0 | 0 - 2 ³¹ -1 | A 0-based index specifying which row from the results to return data from. Can be used to page through the result set. Improves performance when used with challenge ordering , because you can return a result on a specific page. |

| Name | Required | Type* | Default Value | Range | Comments |
|---------------|----------|----------------------|---------------|--|--|
| tags | No | Comma-separated list | None | 0 - 5 items Truncated 0 - 255 chars per item | A comma-separated list of action tags to query on. Only returns challenges with rules that use the tags specified. For example, read_article action only returns challenges where this tag is included in the challenge. |
| userId | No | String | None | 0 - 255 chars | User to get responses for. Defaults to the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------------------------|---|
| achieved | Indicates if the challenge was completed: <ul style="list-style-type: none"> ▪ 0 - Not completed. Always returns 0 for repeatable challenges. ▪ 1 - Completed |
| actionUrl | The URL associated with the challenge (the page where the user must go to complete the challenge). |
| applyMultiplier | The points multiplier in use for the challenge. |
| canAchieveChallenge | Returns true or false indicating if the user can complete the challenge. |
| ChallengeRewards | The rewards earned for completing the challenge: <ul style="list-style-type: none"> ▪ type - The reward type. Returns point, notification, or item. ▪ details - Additional reward information. For points, the details may be something like "285424 50" with the number of points awarded after the pipe (). In this example, 50 points are awarded to the user after completing the challenge. ▪ itemName - The name of the point category, catalog item, or notification. |
| completionCount | The number of times the user has completed the challenge. |
| customData | The challenge's custom data. |

| Response | Description |
|------------------------------|--|
| dailyAchievementLimit | The number of times the challenge can be completed in a day. Returns 0 for unlimited. |
| dateCompleted | The date the challenge was completed (UNIX timestamp). For repeatable challenges, returns the last date the challenge was completed. Note: The user must have completed the action specified in the tag parameter to return this response. |
| dateIssued | The date the challenge becomes active (UNIX timestamp). |
| description | The challenge's description. |
| endTime | The date the challenge ends (UNIX timestamp). Returns 0 if no end date is set. Typically used to display the expiration date of the challenge to the user. |
| featured | Returns true or false indicating if the challenge is a featured challenge. |
| folderName | The name of the folder the challenge is located in. |
| fullURL | For challenges with badge rewards, returns the URL of the full size badge image. |
| groupFlag | Indicates if it's a group challenge: <ul style="list-style-type: none"> ▪ 0 - Not a group challenge ▪ 1 - A group challenge |
| groupPointAward | The number of points awarded to the group for completing the challenge. |
| hideUntilEarned | Indicates if the challenge is hidden until complete: <ul style="list-style-type: none"> ▪ 0 - Not hidden ▪ 1 - Hidden Repeatable, hidden challenges show in both complete and incomplete once initially completed. |
| iconUrl | The URL to the icon used for the challenge. |
| id | The challenge ID. |
| metricGoal | Returns metric goal rule details. |
| name | The challenge name. |
| pointAward | Replaced by ChallengeRewards. |
| pointCategory | For challenges with point rewards, returns the name of the point category the points are awarded to. |
| pointCategoryId | The ID of the point category. |
| preferences | Used for metric goals where a segment determines which value to use in the goal. |

| Response | Description |
|------------------------|---|
| prereqMatchType | Indicates if the prerequisite is and or or. |
| repeatable | Indicates if the challenge is repeatable: <ul style="list-style-type: none"> ▪ 0 - Not repeatable ▪ 1 - Repeatable |
| ruleMatchType | Indicates if the rule operator is and or or. |
| rules | <ul style="list-style-type: none"> ▪ metadata - The metadata associated with the rule. Returns name and value pair. ▪ actionTag - The name of the action associated with the rule. ▪ operator - The action goal value: <ul style="list-style-type: none"> ▪ GE - Greater than or equal to ▪ GT - Greater than ▪ LT - Less than ▪ LE - Less than or equal to ▪ EQ - Equal to ▪ description - The rule's description (user-facing). ▪ completed - Returns true or false indicating if the rule has been completed. ▪ goal - The action value goal required by the rule. For Less than scenarios, the action value lists this value as the number configured in Nitro Studio. Must be used with operator for Less than scenarios. ▪ achieved - Indicates the user's progress toward completing the rule. Used with goal ($\text{achieved} / \text{goal} * 100 = \% \text{ complete}$). ▪ type - Indicates the action goal type: <ul style="list-style-type: none"> ▪ sum - Running total ▪ count - Number of calls ▪ sortOrder - An integer value indicating the rules' order within the challenge. ▪ displayOrder - The order the rule is shown in the challenge. ▪ serviceType - Previously used for social actions. ▪ serviceActionType - Previously used for social actions. ▪ prereqRuleIds - The ID of the rule that must be completed before this rule (for multi-rule challenges). ▪ prereqOperator - Indicates if one or all prerequisites must be met. Returns and or or. |

| Response | Description |
|--------------------|--|
| serviceType | Previously used for social actions. |
| startTime | The challenge start date (UNIX timestamp). |
| tags | The challenge's tags. Tags can support additional functionality, such as specifying the order the challenge should display. |
| type | The challenge type. Returns: <ul style="list-style-type: none"> ▪ user - User challenges ▪ groupAggregateUser - Group challenges that aggregate user actions ▪ group - Group challenges based on group actions |
| thumbURL | For challenges with badge rewards, returns the URL of the thumb badge image. |

Sample Request


```
method=user.getChallengeProgress&asyncToken=&returnCount=0&tags=null&
userId=4000&showOnlyTrophies=false&showCanAchieveChallenge=false&star
tIndex=0&showRulesProgress=true&showActiveOnly=false&folder=&challeng
eName=AttendWorkshop
```

Sample Response

```
{ "response": { "challenges": { "challenge": [ { "rules": { "rule": [
{ "metadata": {}, "actionTag": "learn_more", "operator": "GE",
"description": "1 recorded attendance", "completed": "false", "goal":
1, "timeRange": 3600, "achieved": 0, "type": "count", "sortOrder": 0,
"displayOrder": 0, "serviceType": "nitro", "serviceActionType":
"none" } ] }, "challengeRewards": { "challengeReward": [ { "type":
"point", "details": "295682|5", "itemName": "Funny Money" }, {
"type": "notification", "details": "709121", "itemName": "Good Job" }
] }, "id": 4802302 "type": "user" "name": "AttendWorkshop",
"description": "Attend a lunch and learn session", "activeFlag": 1,
"featuredFlag": 0, "serviceType": "1", "pointCategory": "Funny
Money", "iconUrl": "C:\\Users\\purvi.patel\\Documents\\Random
Files\\Lunch.png", "actionUrl":
"C:\\Users\\purvi.patel\\Documents\\Random Files", "thumbUrl":
"https://s3.amazonaws.com/images/724f78eb/userUploadpng1579ab.png",
"fullUrl":
"https://s3.amazonaws.com/images/7283f78eb/userUploadpng157292.png",
"pointAward": 5, "dateIssued": "1475218800", "dateCompleted":
"1474060252", "applyMultiplier": 0, "repeatable": 1,
"hideUntilEarned": 0, "pointCategoryId": 295682, "completionCount":
1, "customData": "WorkshopCustom", "tags": "WorkshopTag",
```

```
"startTime": 1475218800, "endTime": 1504249200, "ruleMatchType": 0,
"prereqMatchType": "or", "preferences": "", "preferenceValues": "",
"preferencesMatchType": "or", "groupFlag": 0, "groupPointAward": 0,
"dailyAchievementLimit": 0, "folderName": "Onboarding" } ] } },
"responseType": {}, "isError": false, "httpError": 0,
"httpsStatusCode": 200 }
```

user.getGroups Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Gets the groups that the specified user is a member of. For all groups for your site, see [site.getGroups](#).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| userId | No | String | None | 0 - 255 chars | User to get the groups for. By default, this is the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-------------|-----------------|
| id | The group ID. |
| name | The group name. |


Sample Request

```
method=user.getGroups&asyncToken=&userId=4000
```

Sample Response

```
{ "response": { "userGroups": { "group": [ { "id": 35, "name": "Country,Department:France,Research" }, { "id": 30, "name": "Country,Department:France,Engineering" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpStatusCode": 200 }
```

user.getLevel Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Gets the user's current [level](#). Works for either points-based or custom leveling. If there are no levels defined or the user doesn't have enough points to get to the first level (in points-based leveling), the user has no level. With custom-leveling, the user defaults to the first level until a level is set using [user.setLevel](#).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------------|---------------|--|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| userIds | No | Generic list | None | 0 - 100 items Truncated | A comma-separated list of user IDs. Defaults to the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------------|---|
| description | The level's description. |
| iconUrl | The URL to the image associated with the level. |
| name | The level name. |
| points | The level's point value. |

| Response | Description |
|-----------|---|
| timestamp | The date the user reached the level (UNIX timestamp). |
| type | The level type. Returns points or custom. |
| userId | The user's ID. |


Sample Request

```
method=user.getLevel&asyncToken=&userIds=4000,2000
```

Sample Response

```
{ "response": { "users": { "user": [ { "siteLevel": { "points": 2000,
"name": "Jedi Knight", "description": "Ready to do battle",
"iconUrl":
"https://s3.amazonaws.com/images/72838eb/userUploadpng1550.png",
"type": "points", "timestamp": 1473801648 }, "userId": "4000" }, {
"siteLevel": { "points": 2000, "name": "Jedi Knight", "description":
"Ready to do battle", "iconUrl":
"https://s3.amazonaws.com/images/7284f78eb/userUploadpng15b50.png",
"type": "points", "timestamp": 0 }, "userId": "2000" } ] } },
"responseType": {}, "isError": false, "httpError": 0,
"httpsStatusCode": 200 }
```

user.getNextChallenge Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Returns a new challenge for a user to work towards. This only applies to non-repeatable, badge challenges. The challenge returned is random. See [user.getChallengeProgress](#) to return challenges in order.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|---------|---------------|---------------|--|
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| showRulesProgress | No | Boolean | false | true/false | Show progress towards the rules for the challenge. |
| userId | No | String | None | 1 - 255 chars | The user to be awarded the challenge. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-----------------------|---|
| activeFlag | Indicates if the challenge is active: <ul style="list-style-type: none"> 0 - Not active 1 - Active |
| applyMultiplier | The points multiplier in use for the challenge. |
| callbackFlag | Indicates if the challenge is enabled to trigger an achievement callback when completed. |
| ChallengeRewards | The rewards associated with the challenge: <ul style="list-style-type: none"> type - Indicates the reward type. Returns point, item, or notification. details - Additional reward information. For points, the details may be something like "285424 50" with the number of points awarded after the pipe (). In this example, 50 points are awarded to the user after completing the challenge. itemName - The name of the point category, catalog item, or notification. |
| dailyAchievementLimit | The number of times the challenge can be completed in a day. Returns 0 for unlimited. |
| dateIssued | The date the challenge was activated. |
| description | The challenge's description. |
| endTime | The date/time the challenge ends (UNIX timestamp). |
| fullUrl | For challenges with badge rewards, returns the URL of the full size badge image. |
| groupFlag | Indicates if the challenge is a group challenge: <ul style="list-style-type: none"> 0 - Not a group challenge. |

| Response | Description |
|-----------------------------|--|
| | <ul style="list-style-type: none"> ▪ 1 - A group challenge. |
| groupPointAward | The number of points awarded for this challenge. |
| hideUntilEarned | Indicates if the challenge is hidden until earned: <ul style="list-style-type: none"> ▪ 0 - Not hidden ▪ 1 - Hidden |
| id | The challenge ID. |
| name | The challenge name. |
| order | Indicates if the challenge rules must be completed in order: <ul style="list-style-type: none"> ▪ 0 - Do not have to be completed in order. ▪ 1 - Must be completed in order. |
| pointAward | The number of points awarded for completing the challenge. |
| pointCategory | For challenges with point rewards, returns the name of the point category the points are awarded to. |
| pointCategoryId | The ID of the point category. |
| preferences | Used for metric goals where a segment determines which value to use in the goal. |
| preferencesMatchType | Indicates if the prerequisite is and or or. |
| repeatable | Indicates if the challenge is repeatable: <ul style="list-style-type: none"> ▪ 0 - Not repeatable ▪ 1 - Repeatable <p>Repeatable, hidden challenges show in both complete and incomplete once initially completed.</p> |
| ruleMatchType | Indicates if the rule operator is and or or. |
| rules | The rules associated with the challenge: <ul style="list-style-type: none"> ▪ actionTag - The name of the action associated with the rule. ▪ description - The rule's description. ▪ displayOrder - The order the rule is shown in the challenge. ▪ goal - The action value. For Less than scenarios, the action value lists this value as the number configured in Nitro Studio. Must be used with operator for Less than scenarios. ▪ metadata - The metadata associated with the rule. Returns name and value pair. ▪ operator - The action goal value: |

| Response | Description |
|--------------------|---|
| | <ul style="list-style-type: none"> ▪ GE - Greater than or equal to ▪ GT - Greater than ▪ LT - Less than ▪ LE - Less than or equal to ▪ EQ - Equal to ▪ prereqOperator - Indicates if one or all prerequisites must be met. Returns and or or. ▪ prereqRuleIds - The ID of the rule that must be completed before this rule (for multi-rule challenges). ▪ serviceActionType - Previously used for social actions. ▪ serviceType - Previously used for social actions. ▪ sortOrder - An integer value indicating the rules' order within the challenge. ▪ type - Indicates the action goal type: <ul style="list-style-type: none"> ▪ sum - Running total ▪ count - Number of calls |
| serviceType | Previously used for social actions. |
| startTime | The date/time the challenge starts (UNIX timestamp). |
| tags | The tags associated with the challenge. |
| thumbUrl | For challenges with badge rewards, returns the URL of the thumb badge image. |

Sample Request


```
method=user.getNextChallenge&asyncToken=&showRulesProgress=false&userId=2001
```

Sample Response

```
{ "response": { "challenges": { "challenge": [ { "rules": { "rule": [ { "metadata": {}, "actionTag": "app_login", "operator": "GE", "description": "App login", "goal": 1, "timeRange": 3600, "type": "count", "sortOrder": 0, "displayOrder": 0, "serviceType": "nitro", "serviceActionType": "none" } ] }, "challengeRewards": { "challengeReward": [ { "type": "point", "details": "295169|44", "itemName": "Points" } ] }, "id": 4820744, "name": "Action phrase test", "description": "Action phrase test", "activeFlag": 1, "featuredFlag": 0, "serviceType": "1", "pointCategory": "Points",
```

```
"thumbUrl":
"https://s3.amazonaws.com/images/72830ac4f78eb/5f49de40dd.png",
"fullUrl":
"https://s3.amazonaws.com/images/7283d2ac4f78eb/49de3260.png",
"pointAward": 44, "dateIssued": "1508772780", "applyMultiplier": 0,
"repeatable": 0, "hideUntilEarned": 0, "pointCategoryId": 295169,
"completionCount": 0, "tags": "", "startTime": 1508772780, "endTime":
0, "ruleMatchType": 0, "prereqMatchType": "or", "preferences": "",
"preferenceValues": "", "preferencesMatchType": "or", "groupFlag": 0,
"groupPointAward": 0, "dailyAchievementLimit": 0, "folderName":
"Test" } ] } }, "responseType": {}, "isError": false, "httpError": 0,
"httpsStatusCode": 200 }
```

user.getNextLevel Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Gets the level after the user's current level in the site's level list. Works for either points-based or custom level type.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|--|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| userId | No | String | None | 0 - 255 chars | The user's unique ID for the site. Must be alphanumeric and no special characters (can be hashed). It's important to use a commonly known ID. If not specified, defaults to the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------------|--|
| description | The level's description. |
| iconUrl | The URL to the image associated with the level. |
| name | The level name. |
| points | The level's point value. To determine how many points the user needs to reach this level, you need to calculate the difference between the level's point value and the user's current points |
| timestamp | Returns 0. |
| type | The level type. Returns points or custom. |
| userId | The user's ID. |


Sample Request

```
method=user.getNextLevel&asyncToken=&userId=4000
```

Sample Response

```
{ "response": { "users": { "user": [ { "siteLevel": { "points": 10000, "name": "Jedi Master", "description": "Master of the galaxy", "iconUrl": "https://s3.amazonaws.com/images/72878eb/userUploadpng1570c74.png", "type": "points", "timestamp": 0 }, "userId": "4000" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

user.getOwnedItems Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Gets owned items for the given user.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|--|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the |

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|--|
| | | | | | response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| userId | No | String | None | 0 - 255 chars | If not specified, returns items for the user associated with the sessionKey. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-----------------------------|--|
| background | Indicates if the item is a background item: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| canUnselect | Returns true or false. |
| catalogItemId | The item's catalog ID. |
| catalogName | The name of the catalog the item is contained in. |
| category | The category the item is contained in. |
| categoryIds | The ID of the category the item is contained in. |
| customData | The custom data associated with the item. |
| description | The item's description. |
| doCallback | Indicates if a callback is triggered when a user purchases the item: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| fullUrl | URL to the full size image associated with the item. |
| id | The item's ID number. |
| lastPurchased | The date and time the item was last purchased. |
| maxOwnedCountPerUser | The maximum number of the item a user can purchase. |
| name | The item's name. |
| orders | The order the item appears in the catalog. |
| pointCategoryId | The ID of the point category that points are deducted from when the user purchases this item. |

| Response | Description |
|------------------------|--|
| pointCategoryId | The name of the point category that points are deducted from when the user purchase this item. |
| price | The cost of the item (in points). |
| realItem | Indicates if the item shows in the user's Owned Items list in widgets: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| recordId | The ID of the record and user. |
| tags | The tags associated with the item. |
| thumbUrl | URL to the thumbnail image associated with the item. |
| zOrder | The item's Z-Order number. |


Sample Request

```
method=user.getOwnedItems&asyncToken=&userId=2000
```

Sample Response

```
{ "response": { "ownedItemsRecord": { "ownedItems": { "catalogItem":
[ { "id": 111459082, "description": "Company tee", "customData": "",
"background": 0, "thumbUrl":
"https://www.bunchball.com/badge_logo_web.png", "designer": "",
"zOrder": 0, "name": "Company T-shirt", "catalogName": "Store",
"fullUrl": "https://www.bunchball.com/badge_logo_web.png", "price":
0, "categoryIds": "2895873", "category": "Shopping", "tags": "",
"canUnselect": "true", "realItem": 1, "doCallback": 0,
"passExtraData": 0, "orders": "0", "pointCategoryId": 295682,
"pointCategoryName": "Funny Money", "lastPurchased": "2016-09-14
08:31:24", "catalogItemId": 31208834, "maxOwnedCountPerUser": 0 }, {
"id": 111459078, "description": "Company tee", "customData": "",
"background": 0, "thumbUrl":
"https://www.bunchball.com/logo_web.png", "designer": "", "zOrder":
0, "name": "Company T-shirt", "catalogName": "Store", "fullUrl":
"https://www.bunchball.com/logo_web.png", "price": 0, "categoryIds":
"2895873", "category": "Shopping", "tags": "", "canUnselect": "true",
"realItem": 1, "doCallback": 0, "passExtraData": 0, "orders": "0",
"pointCategoryId": 295682, "pointCategoryName": "Funny Money",
"lastPurchased": "2016-09-14 08:11:01", "catalogItemId": 31208834,
"maxOwnedCountPerUser": 0 } ] }, "recordId": "OwnedItemsRecord.2000"
} }, "responseType": {}, "isError": false, "httpError": 0,
"httpsStatusCode": 200 }
```

user.getPendingNotifications Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Used by the Nitro [JavaScript library](#) to retrieve notifications to display for a specified user. In most cases, you will not call this method directly. Instead, use the nitro.js Nitro object's [showPendingNotifications](#) method because it does all the string replacement and animation for you.

The response includes all metadata for the [notifications](#) and any styles used by those notifications, unreplaced. The response also includes values for all dynamic settings, as well as any relevant trigger metadata. All string replacement must be done by the caller, as well as implementation of the default settings (duration, delay, repeat, corner, direction).

Note: Using this method removes the notification(s) from the user's queue.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|----------------------|---------------|--|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| returnCount | No | Integer | 10 | 1 - 100 | The number of notifications to return. |
| tags | No | Comma-separated list | None | 0 - 5 items Truncated 0 - 255 chars per item | A comma-separated list of tags to filter on. Tags are associated with notifications in Reward & Motivate > Notifications > create/edit > Advanced Details . |
| userId | No | String | None | 0 - 255 chars | User to get notifications for. By default, this is the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------------------------|---|
| action | The action the user completed to trigger the notification. |
| balance | The user's point balance. |
| challenge | The challenge the action was completed for. |
| challenges | The challenge(s) the notification is used in: <ul style="list-style-type: none"> ▪ name - The challenge name. ▪ id - The challenge ID. |
| challengeTrophyFullUrl | For challenges with badge rewards, returns the URL of the full size badge image. |
| challengeTrophyThumbUrl | For challenges with badge rewards, returns the URL of the thumb badge image. |
| corner | The location the notification displays. Returns Bottom Right, Bottom Left, Top Right, Top Left, or Center of Screen. |
| delay | The length of time to wait before the notification appears (toast only). |
| description | The notification's description. |
| direction | For a popup notification, the way the notification moves onto the screen. Returns Vertical or Horizontal. |
| duration | The length of time to display the notification. |
| firstName | The user's first name. |
| frequency | The notification's frequency. Returns Unlimited, Once Per Day, or Only Once. |
| lastName | The user's last name. |
| lifetimeBalance | The user's lifetime point balance. |
| name | The notification's name. |
| newsfeed | The newsfeed entry text specified for the action. |
| notificationSettings | The notification's appearance. Each setting has a name/value pair: <ul style="list-style-type: none"> ▪ 90x90 Image - The image that displays in the notification. ▪ Title - The notification's title. ▪ Subtitle - The notification's subtitle. ▪ Message - The notification's message text. |
| notification styles | <ul style="list-style-type: none"> ▪ name - The name of the notification style the notification uses. ▪ type - The notification style type, returns either popup or feed. |

| Response | Description |
|---------------------------|---|
| | <ul style="list-style-type: none"> html - The notification style's html. css - The notification style's css. order - The notification's sort order. |
| order | The order the notification should display if multiple notifications are pending. |
| pointCategory | For challenges with point rewards, returns the name of the point category the points are awarded to. |
| pointCategoryLimit | The lifetime maximum balance a user can have in this point category. |
| points | The number of points associated with the action. |
| styleName | The notification style . |
| timestamp | The time the method was called (UNIX timestamp). |
| type | The notification type, returns either popup or feed. |
| updated | The time the notification settings were updated (UNIX timestamp). |
| userId | The ID of the user receiving the notification. |
| value | The action value. |

Sample Request

```
method=user.getPendingNotifications&asyncToken=&userId=4000&returnCount=1&tags=null
```

Sample Response

```
{ "response": { "notifications": { "notification": [ { "challenges": { "minis": [ { "name": "Add a New Beneficiary", "id": 4805220 }, { "name": "AttendWorkshop", "id": 4802302 }, { "name": "Collaborate with others", "id": 4802234 }, { "name": "Log in to the mobile app", "id": 4805135 }, { "name": "Post a Document Contest", "id": 4753793 }, { "name": "TrainingVideos", "id": 4799489 } ] }, "notificationSettings": { "notificationSetting": [ { "name": "90x90 Image", "value": "http://dynamic.bunchball.com/assets/avatar/{nitro_apiKey}/{nitro_userId}/thumb.png?size=90" }, { "name": "Title", "value": "WINNER WINNER" }, { "name": "Subtitle", "value": "Good Job!" }, { "name": "Message", "value": "Way to go completing your challenge" } ] }, "timestamp": 1473453478, "name": "Good Job", "type": "popup", "description": "Good job completing your challenge", "updated": 1472681666, "doShare": 0, "version": 0, "styleName": "Badge of Honor", "userId": "2000", "firstName": "David", "lastName": "Brown",
```


```

"balance": "6489", "lifetimeBalance": "6489", "action":
"RELEASE_DEPLOYED", "value": "0", "pointCategory": "Points",
"pointCategoryIconUrl":
"https://s3.amazonaws.com/images/7283d4f78eb/userUploadpng151.png",
"pointCategoryLimit": "null", "order": 0, "duration": "5", "delay":
"0", "frequency": "Unlimited", "corner": "Bottom Right", "direction":
"Vertical" } ] }, "notificationStyles": { "notificationStyle": [ {
"name": "Badge of Honor", "type": "popup", "html": "<div
class=\"main01\"> <div class=\"row\"> <img
src=\"http://assets.bunchball.com/scripts/media/notifications/popup01
/images/top.png\" width=\"329\" height=\"39\" alt=\"\" /> </div> <div
class=\"bg01\"> <div class=\"image\"> <img
src=\"{{notification_setting_90x90
Image|http://dynamic.bunchball.com/assets/avatar/{nitro_apiKey}/{nitro
_userId}/thumb.png?size=90|HTML}}\" width=\"90\" height=\"90\"
alt=\"\" /> </div> <div class=\"imagetxt\"> <h1>
{{notification_setting_Title|AENEAN TRISTIQUE ARCU EGET
QUAMON|string}} </h1> <p> {{notification_setting_Subtitle|Lorem ipsum
dolor sit amet, consectetur adipiscing elit sagittis.|string}} </p>
</div> </div> <div class=\"bg02\"> <p>
{{notification_setting_Message|Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Ut facilisis, est ac vulputate sagittis,
leo ipsum congue velit, at sagittis nisl ante eget odio. Nunc ornare
malesuada neque. |string}} </p> </div> <div class=\"row\"> <img
src=\"http://assets.bunchball.com/media/notifications/popup01/images/
down.png\" width=\"329\" height=\"30\" alt=\"\" /> </div> </div>\",
"css": "div.main01 { clear: both; margin: 0px; width: 329px; text-
align: left; line-height: 16px; } div.row { clear: both; line-height:
0; } div.cell { display: table-cell; float: left; } div.bg01 { clear:
both; background-image:
url(http://assets.bunchball.com/scripts/media/notifications/popup01/i
images/bg01.png) !important; background-repeat: repeat-y; background-
image: none; filter: none !important; filter:
progid:DXImageTransform.Microsoft.AlphaImageLoader(src='http://assets
.bunchball.com/scripts/media/notifications/popup01/images/bg01.png',
sizingMethod='scale'); display: table; width: 329px; } div.image {
float: left; margin-left: 29px; position: relative; background-image:
url(http://assets.bunchball.com/scripts/media/notifications/popup01/i
images/image-bg.jpg); background-repeat: no-repeat; width: 122px;
height: 102px; } div.image img { position: absolute; top: 2px; left:
13px; } div.imagetxt { float: left; width: 153px; background-image:
url(http://assets.bunchball.com/media/notifications/popup01/images/im
agetxt-bg.jpg); background-repeat: no-repeat; background-position:
top right; } div.imagetxt h1 { color: white; font-size: 13px; font-
weight: bold; padding: 0 0 5px 0; margin: 0; width: 135px; line-
height: 19px; font-family: \"Trebuchet MS\"; } div.imagetxt p {
color: white; font-size: 11px; font-weight: bold; width: 135px; line-

```

```
height: 16px; padding: 0; margin: 0; font-family: \"Trebuchet MS\"; }
div.bg02 { clear: both; background-image:
url(http://assets.bunchball.com/media/notifications/popup01/images/bg
02.png) !important; background-repeat: repeat-y; background-image:
none; filter: none !important; filter:
progid:DXImageTransform.Microsoft.AlphaImageLoader(src='http://assets
.bunchball.com/media/notifications/popup01/images/bg02.png',
sizingMethod='scale'); display: table; width: 329px; } div.bg02 p {
width: 252px; margin-left: 44px; padding: 10px 0; color: #c0c0c0;
font-family: \"Trebuchet MS\"; font-size: 11px; }\", \"order\": 0 } ] }
}, \"responseType\": {}, \"isError\": false, \"httpError\": 0,
\"httpsStatusCode\": 200 }
```

user.getPointsBalance Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Returns the point balance for the specified user. If the specified user doesn't exist, the function fails.

The criteria can be BALANCE (the user's point balance over the period, which is credits minus debits), CREDITS (sum of all credits for the user in the specified time period), or DEBITS (sum of debits for the user in the specified time period). Using the start and end dates allow you to see a user's points accrual during a specific time range.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| criteria | No | String | BALANCE | 0 - 255 chars | How to compute the balance: <ul style="list-style-type: none"> ▪ BALANCE - Credits minus debits during the time range, if using start and end |

| Name | Required | Type* | Default Value | Range | Comments |
|-----------------------------|----------|----------------------|---------------|---|--|
| | | | | | <p>date. Current points balance if not using time range.</p> <ul style="list-style-type: none"> ▪ CREDITS - Sum during the time range, if using start and end date. Lifetime points if not using time range. ▪ DEBITS - Sum during the time period. |
| end | No | Integer | Now() | 0 - 2 ³¹ -1 | End date to use. A UNIX timestamp. |
| includeYearlyCredits | No | Boolean | false | true/false | <p>If specified, then point categories with yearly max credits also include the credits earned so far this year.</p> <p>Yearly max credits are set in Nitro Studio > Configuration > Site Settings > Point Categories > new/edit > Yearly Max Credits field).</p> |
| pointCategory | No | String | None | 0 - 255 chars | <p>The point category to retrieve the balance for. Can be "all" to get all the categories.</p> <p>Defaults to the default point category defined in Nitro Studio (Configuration > Site Settings > Point Categories > new/edit > Set as default category field).</p> |
| pointsId | No | String | None | 0 - 255 chars | The user's point category points ID to query for. |
| start | No | Integer | 0 | 0 - 2 ³¹ -1 | Start date to use. A UNIX timestamp. |
| tags | No | Comma-separated list | None | 0 - 10 items Truncated | A comma-separated list of values. Only considers credits and debits with these tags. |

| Name | Required | Type* | Default Value | Range | Comments |
|---------------|----------|--------|---------------|------------------------|---|
| | | | | 0 - 255 chars per item | |
| userId | No | String | None | 0 - 255 chars | The user's unique ID for the site. Must be alphanumeric and no special characters (can be hashed). It's important to use a commonly known ID. If not specified, gets userId from sessionKey. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------------|--|
| lifetimeBalance | The number of lifetime points earned by the user. |
| pointCategories | <ul style="list-style-type: none"> ▪ pointsId - The user's point category points ID. The ID is only returned if the user has points in the point category. ▪ id - The point category ID. ▪ name - The point category name. ▪ points - The number of points the user earned in the point category. ▪ lifetimeBalance - The user's lifetime balance for the point category. ▪ iconUrl - For point categories with an associated image, the URL to the image. ▪ shortName - The point category's short name. ▪ isDefault - Returns true or false indicating if this is the default point category. ▪ yearlyMaxCredits - The maximum number of points the user can be awarded each year for the point category. ▪ yearlyCredits - The number of points earned so far this year for the point category. |
| points | The user's current points balance based on the selected criteria parameter. |
| userId | The user's ID. |
| yearlyCredits | The number of points earned by the user so far this year. |


Sample Request

```
method=user.getPointsBalance&asyncToken=&end=&tags=null&includeYearlyCredits=false&pointCategory=all&userId=judi.ingram&start=&criteria=BALANCE
```

Sample Response

```
{ "response": { "balance": { "pointCategories": { "pointCategory": [ { "pointsId": 6320, "id": 1, "name": "Points", "premium": 0, "points": 6200, "lifetimeBalance": 6200, "iconUrl": "https://s3-us-west.amazonaws.com/bunchball/point_category_icon-1?15235271", "shortName": "Pts", "isDefault": true }, { "pointsId": 6354, "id": 589, "name": "Recognition", "premium": 0, "points": 125, "lifetimeBalance": 125, "iconUrl": "https://s3-us-west.amazonaws.com/bunchball/point_category_icon--559?151384", "isDefault": false } ] }, "userId": "judi.ingram", "points": 6200, "lifetimeBalance": 6220 } }, "responseType": {}, "isError": false, "httpError": 0, "httpStatusCode": 200 }
```

user.getPointsHistory Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Returns the point accumulation history for the specified user within a given range, most recent activity first. If the specified user doesn't exist, the function fails.

Used mainly for reporting and troubleshooting.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|---------|---------------|------------------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| end | No | Integer | Now() | 0 - 2 ³¹ -1 | End date to use. A UNIX timestamp. |

| Name | Required | Type* | Default Value | Range | Comments |
|------------------------|----------|----------------------|---------------|---|---|
| pointCategory | No | String | None | 0 - 255 chars | The point category to query for. Can be "all" to get all the categories. Defaults to the default point category defined in Nitro Studio (Configuration > Site Settings > Point Categories > new/edit > Set as default category field). |
| pointCategoryId | No | String | None | 0 - 255 chars | The point category ID to query for. |
| returnCount | No | Integer | 10 | 1 - 100 | The number of records to return. |
| start | No | Integer | 0 | 0 - 2 ³¹ -1 | Start date to use. A UNIX timestamp. |
| startIndex | No | Integer | 0 | 0 - 2 ³¹ -1 | 0-based index of which row from the results to return data from. Can be used to page through the result set. |
| tags | No | Comma-separated list | None | 0 - 10 items Truncated 0 - 255 chars per item | A comma-separated list of tags to filter on. |
| tagsOperator | No | Operator | OR | AND/OR | Actions that have any of the tags are considered. If set to AND, then only actions that have all of the tags are considered. |
| userId | No | String | None | 0 - 255 chars | The user's unique ID for the site. Must be alphanumeric and no special characters (can be hashed). It's important to use a commonly known ID. If not specified, uses the user associated with the sessionKey. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-------------------------------|--|
| action | The type of point transaction: <ul style="list-style-type: none"> ▪ credit - Points given to the user. ▪ debit - Points removed from the user. |
| amount | The number of points awarded or debited. |
| id | The ID of the points history record. |
| pointCategory | The point category the points are associated with. |
| pointCategoryId | The point category ID. |
| pointCategoryIsDefault | Indicates if the point category for the points history transaction is the default for the site. Returns true or false. |
| tags | The actions or challenges that are associated with the point award/debit. |
| ts | The time when the point transaction occurred (UNIX timestamp). |
| updateLifetimeBalance | Indicates if the lifetime balance was updated with the points history transaction. Returns true or false. |


Sample Request

```
method=user.getPointsHistory&asyncToken=&end=&tags=COMPLETE_COURSE&returnCount=10&pointCategory=all&userId=judi.ingram&start=&startIndex=0&tagsOperator=OR
```

Sample Response

```
{ "response": { "pointsHistoryRecord": { "pointsHistoryItem": [ {
  "id": 11249, "ts": 1529070067, "amount": 50, "action": "credit",
  "tags": "QUIZ_COMPLETED,challenge:Become Keepersite
  Certified,COMPLETE_COURSE,DOWNLOAD_DOCUMENT",
  "updateLifetimeBalance": true, "pointCategoryId": 1, "pointCategory":
  "Points", "pointCategoryIsDefault": true } ] }, "returnCount": 10,
  "startIndex": 0 }, "responseType": {}, "isError": false, "httpError":
  0, "httpStatusCode": 200 }
```

user.getPreference Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Gets a [user preference](#).

Each user has a set of preferences that can be defined for them. You can use preferences to add additional information about the user, such as first name, last name, or other custom data. There are five special reserved preferences that Nitro uses:

- **firstName** - Displays the first name of the user in the profile, leaderboards, events feed, etc.
- **lastName** - Displays the last name of the user in the profile, leaderboards, events feed, etc.
- **userPhotoUrl** - Displays the photo/avatar image of the user in the profile and leaderboards.
- **role** - Displays the user's job title/role in the Profile widget.
- **anonymous** - Used with ANY value, including no value, to hide the user from leaderboards and events feeds.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| name | YES | String | None | 0 - 255 chars | The preference name. |
| userId | No | String | None | 1 - 255 chars | User to get the preference for. By default, this is the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------|-------------------------|
| name | The preference name. |
| value | The preference's value. |

Sample Request

```
method=user.getPreference&asyncToken=&userId=2000&name=firstName
```

Sample Response

```
{ "response": { "userPreferences": { "userPreference": [ { "name": "firstName", "value": "Kasey" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

user.getPreferences Method



As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Gets the specified [preferences](#) for a given user. If Personally Identifiable Information (PII) is a concern, Bunchball does not recommend setting these preferences in Nitro. An internal service should be used to look up this information instead.

Each user has a set of preferences that can be defined for them. You can use preferences to add additional information about the user, such as first name, last name, or other custom data. There are five special reserved preferences that Nitro uses:

- **firstName** - Displays the first name of the user in the profile, leaderboards, events feed, etc.
- **lastName** - Displays the last name of the user in the profile, leaderboards, events feed, etc.
- **userPhotoUrl** - Displays the photo/avatar image of the user in the profile and leaderboards.
- **role** - Displays the user's job title/role in the Profile widget.
- **anonymous** - Used with ANY value, including no value, to hide the user from leaderboards and events feeds.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|--|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is |

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|---------------------|---------------|---------------|--|
| | | | | | the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| names | YES | Pipe-separated list | None | 1 - 100 items | Pipe-separated list of preference names. If not specified, all preferences are returned. |
| userId | No | String | None | 1 - 255 chars | User to get preferences for. By default, this is the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------|-------------------------|
| name | The preference name. |
| value | The preference's value. |


Sample Request

```
method=user.getPreferences&asyncToken=&userId=2000&names=
```

Sample Response

```
{ "response": { "userPreferences": { "userPreference": [ { "name": "Scale", "value": "6" }, { "name": "SiteID", "value": "Sacramento" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

user.getResponses Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Allows you to get a set of challenges for a specific action. You can instruct Nitro to store the response of a [logAction](#) call using the [storeResponse](#) parameter. Then you can look up the response later using `user.getResponses`. This is an effective way to avoid having to wait for the response server-side. You can look it up client-side after the page is done loading.

After retrieving the responses, this method deletes them from the server. This means you can only look up a given response once, and this method only returns the responses that were stored since the last time `getResponses` was called for the user. When there is more than one stored response to retrieve, all responses are returned.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The <code>sessionKey</code> response value returned from user.login . |
| userId | No | String | None | 0 - 255 chars | User to get responses for. By default, this is the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|---------------------------|---|
| Achievements | <ul style="list-style-type: none"> ▪ points - The number of points received from the completed challenge. ▪ tags - The action names. ▪ value - The action's value. ▪ target - If specified in the action, the target name. |
| catalogItems | For challenges with catalog item rewards, returns the item name. |
| challengesAchieved | <ul style="list-style-type: none"> ▪ points - The number of points received from the completed challenge. ▪ pointCategory - The category the points were awarded from. ▪ thumbUrl - For challenges with badge rewards, returns the URL of the thumbnail badge image. ▪ fullUrl - For challenges with badge rewards, returns the URL of the full size badge image. ▪ repeatable - Indicates if the challenge is repeatable. <ul style="list-style-type: none"> ▪ 0 - Not repeatable ▪ 1 - Repeatable ▪ description - The challenge's description. |

| Response | Description |
|----------------------|--|
| | <ul style="list-style-type: none"> name - The challenge name. |
| pointsRewards | <ul style="list-style-type: none"> points - The number of points received from the completed challenge. pointCategory - The point category the points were awarded from. |


Sample Request

```
method=user.getResponses&asyncToken=&userId=1000
```

Sample Response

```
{ "response": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<Nitro>\n<responses><Nitro method=\"\" server=\"nitro-sandbox-i-05b/nitro\" res=\"ok\" asyncToken=\"\"> <Achievements points=\"0\" tags=\"KUDO_RECEIVED\" value=\"0\" target=\"null\">\n<challengesAchieved/> </Achievements> </Nitro>\n</responses>\n</Nitro>", "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

user.joinGroup Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Adds a single user to a [group](#) (to add multiple users, see [site.addUsersToGroup](#)).

The "Restrict group membership to one group" option limits a user from joining multiple groups (Nitro Studio > **Configuration** > **Site Settings** > **General** > **Groups**). If users are only allowed to join one group, `user.joinGroup` adds the user to the new group and removes them from the old group. If users can join multiple groups, you may want to use [user.leaveGroup](#) to remove the user from the old group.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |

| Name | Required | Type* | Default Value | Range | Comments |
|------------|----------|--------|---------------|---------------|---|
| groupName | YES | String | None | 1 - 255 chars | The name of the group. If the group doesn't exist, it may be created automatically based on your Site Settings (Configuration > Site Settings > General > Groups > Automatically create groups when a user joins). |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| userId | No | String | None | 0 - 255 chars | ID of the user to add to the group. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------|------------------------------------|
| name | Name of the group the user joined. |


Sample Request

```
method=user.joinGroup&asyncToken=&userId=2002&groupName=Arizona
```

Sample Response

```
{ "response": { "userGroups": { "group": [ { "name": "Arizona" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

user.leaveAllGroups Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Removes a user from all the manual [groups](#) they are a member of. To leave one group, see [user.leaveGroup](#). If you are using hierarchical groups with the Group Leaderboard widget, remove users from groups with the [Org Structure](#) API.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| userId | No | String | None | 0 - 255 chars | User to remove from all groups. Defaults to the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------------|---|
| isError | <ul style="list-style-type: none"> ▪ true - The user was not removed. ▪ false - The user was successfully removed from all groups |

Sample Request

```
method=user.leaveAllGroups&asyncToken=&userId=2001
```

Sample Response

```
{ "response": { "userGroups": {} }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

user.leaveGroup Method



As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Removes the specified user from a manual [group](#) and returns the list of groups the user is still in. To remove multiple users from a group, see [site.removeUsersFromGroup](#). If you are using hierarchical groups with the Group Leaderboard widget, remove users from groups with the [Org Structure](#) API.

By default, when removing a user from a group, the user's points remain with that group. To transfer a user's points when switching groups, enable the "Retain points when switching groups" option (Nitro Studio > **Configuration** > **Site Settings** > **General** > **Groups**).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| groupName | YES | String | None | 1 - 255 chars | Name of the group to remove the user from. |
| userId | No | String | None | 0 - 255 chars | The user you want to remove from the group. Defaults to the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------------|--|
| isError | <ul style="list-style-type: none"> false - The user was successfully removed from all groups. true - The user was not removed. |
| name | The name of the group the user is still a member of. |


Sample Request

```
method=user.leaveGroup&asyncToken=&userId=2001&groupName=Arizona
```

Sample Response

```
{ "response": { "userGroups": { "group": [ { "name": "NewYork" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

user.logAction Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. The REST version of user.logAction is [/users/{userId}/actions](#).

Logs an action or page-visit to the Nitro server. This is the primary API method used in the Nitro rewards system. Actions that are logged via this method can trigger rule completion and challenge rewards, and can be used to create leaderboards. The majority of log action processing is performed asynchronously so completed challenge information is not included in the response and instead can be obtained either through [user.getResponses](#) or [Callback URLs](#).

See [action logging](#) for recommendations and best practices.

You can:

- Instruct Nitro to store the response of the logAction call using the storeResponse parameter. Then, you can look up the response later using [user.getResponses](#). This allows you to determine the challenges completed by the action. You can look it up client-side after the page is done loading.
- Receive [URL callbacks](#) for newly completed challenges.
- Pass multiple [actions](#) to logAction at once, as well as up to 10 metadata pairs.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. Can also be used as a filter for additional API calls. For example, if user.getResponse passes this token, it only gets a list of challenges for this type of action. This is not stored in Nitro and cannot be used in our reporting. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |

| Name | Required | Type* | Default Value | Range | Comments |
|----------------------|----------|----------------------|---------------|--|---|
| eventID | No | String | None | 0 - 255 chars | Only used in <code>user.getAchievements</code> . Used for transactional ID for support and troubleshooting. Can also be used to pass in the same ID multiple times to avoid redundant data. |
| metadata | No | Comma-separated list | None | 0 - 10 items 0 - 255 chars per item | <p>A comma-separated list of metadata tags (key value pairs separated by a colon). Allows you to populate other contextual information around the action taking place.</p> <p> Alphanumeric. Avoid special characters other than # @ & ? and !. Commas are NOT supported within the key value pair.</p> <p>Metadata is a string comparison and must be an exact match to what is configured in the challenge. Numbers in metadata cannot be treated as numbers, just strings.</p> <p>Note: The Nitro platform currently supports up to a total of 10 metadata pairs to be associated with actions logged. However, analytics can only support up to 5 metadata pairs.</p> |
| newsfeed | No | String | None | 0 - 2000 chars | <p>Exact text for newsfeed entries. Supports Nitro tokens. Returns URL-encoded HTML for the action in the response of site.getActionFeed.</p> <p>Note: Not recommended for use due to the noise it creates in your program.</p> |
| storeResponse | No | Boolean | false | true/false | <p>When true, stores the response for later lookup via user.getResponses. If using <code>asyncToken</code>, set to true.</p> <p>Can be used to track something specific such as finding what challenge completed for a specific action rather than a list of all completed challenges.</p> |

| Name | Required | Type* | Default Value | Range | Comments |
|------------------|----------|----------------------|---------------|--|---|
| tags | YES | Comma-separated list | None | 0 - 10 items 0 - 255 chars per item | <p>Used to pass the action name. A comma-separated list of actions and contextual tags to log. Mainly used to power leaderboards.</p> <p>The parameter is case-sensitive and includes both the action and contextual tag. For example, <code>method=logAction&tags=action_tag,contextual1,contextual2,contextual3</code>.</p> <p>The parameter supports up to a total of 10 items (10 contextual tags, or 10 action tags, or 1 action tag and 9 contextual tags). However, leaderboards and reporting can only support up to 5 contextual tags.</p> <p>Note: Bunchball discourages logging actions with multiple comma-delimited tags and encourages using metadata and preferences.</p> |
| target | No | String | None | 0 - 255 chars | <p>Associates the action with a target. Often used for API calls that allow you to filter by target.</p> <p>Note: Not recommended for use due to low program value.</p> |
| timestamp | No | Integer | 0 | 0 - 2 ³¹ -1 | <p>The date/time as received by Bunchball (UNIX timestamp)</p> <p>When blank, defaults to the current date/time. If specified, allows you to backdate the date/time, which impacts reporting and which challenges/rules are triggered. To log a backdated action, the Make this challenge active option must be enabled in the challenge.</p> |
| userId | No | String | None | 0 - 255 chars | <p>User to log the action for. By default, the action is logged for the authenticated user. Optional if sessionKey is used.</p> |

| Name | Required | Type* | Default Value | Range | Comments |
|-------|----------|---------|---------------|------------------------|---|
| value | No | Integer | 0 | 0 - 2 ³¹ -1 | The action value to save. Must be a number. The value enables you to create leaderboards, see if the user has met certain criteria, and do running total calculations. Can be used for simple number checks like greater than, less than, and with points multiplier . For example, if this action represents taking a quiz, you could pass in the score. Or if you must meet a certain sales goal, you can pass in the dollar amount. |

*[Type Descriptions](#)

Response Parameters

When a tag fails due to rate limiting, the response of logAction always returns it in a list. The request only returns an error response if no tags are successfully processed. If at least one tag is valid, the response is a normal successful logAction response.

| Response | Description |
|----------|------------------------------------|
| ok | The action(s) logged successfully. |
| err | The action(s) failed to log. |


Sample Request

```
method=user.logAction&asyncToken=&target=&tags=attend_2016_conference
&userId=3000&value=0&storeResponse=false&competitionInstanceId=&newsfeed=&metadata=
```

Sample Response

```
{ "response": {}, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

user.login Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Logs a user into Nitro. The response provides the sessionKey parameter used in traditional API calls.

Security

If medium or high security is enabled on your site, login requires a signature as one of the arguments. The final signature looks like MD5(apiKey + secretKey + ts + userId + length). The signature is constructed by:

1. Concatenating the arguments together in this order: concat = apiKey + secretKey + ts + userId.
2. Taking the length (number of characters in the concatenated string) and appending it to the end.
3. Taking the MD5 hash of the result.

The Nitro Server takes the provided arguments, along with its own copy of the Secret key, and constructs the same signature. If there's a match, the transaction proceeds. If there is no match, the transaction is rejected.

The following are basic examples of generating an MD5 hashed signature for a user.login call:

- **C#**

```
//Classes used for this example using System; using System.Text;
//Generate signature. Formula: (apiKey + secretKey + ts + userId) +
lengthOfString public static String makeSignature(){ String
signatureToHash= someApiKey + someSecretKey + someTimestamp +
someUserId; signatureToHash += signatureToHash.Length.ToString();
return md5(signatureToHash); } //String to md5 hash public static
string md5(string input){ //Use input string to calculate MD5 hash
using (System.Security.Cryptography.MD5 md5 =
System.Security.Cryptography.MD5.Create()){ byte[] inputBytes =
System.Text.Encoding.ASCII.GetBytes(input); byte[] hashBytes =
md5.ComputeHash(inputBytes); // Convert the byte array to hexadecimal
string StringBuilder sb = new StringBuilder(); for (int i = 0; i <
hashBytes.Length; i++){ sb.Append(hashBytes[i].ToString("X2")); }
return sb.ToString(); } }
```

- **Java**

```
//Classes used for this example import java.io.*; import
java.security.*; import java.math.BigInteger; //Generate signature.
Formula: (apiKey + secretKey + ts + userId) + lengthOfString public
String buildSignature(){ String signature = someApiKey +
```

```

someSecretKey + someTimestamp + someUser; signature +=
Integer.toString(signature.length()); String signatureEncoded = "";
try{ signatureEncoded = md5(signature); }
catch(NoSuchAlgorithmException e){ System.out.println(e); }
catch(UnsupportedEncodingException f){ System.out.println(f); }
return signatureEncoded; } /** * Return an MD5 hashed string * *
@param input = String * @return md5 hashed String * @throws
NoSuchAlgorithmException * @throws UnsupportedEncodingException */
private static String md5(String input) throws
NoSuchAlgorithmException, UnsupportedEncodingException{ String result
= input; if (input != null) { MessageDigest md =
MessageDigest.getInstance("MD5"); //or "SHA-1"
md.update(input.toString().getBytes("UTF-8")); BigInteger hash = new
BigInteger(1, md.digest()); result = hash.toString(16); while
(result.length() < 32) { //40 for SHA-1 result = "0" + result; } }
return result; }

```

- **PHP**

```

<?php //Generate signature. Formula: (apiKey + secretKey + ts +
userId) + lengthOfString function makeSignature() { $sig =
$someApiKey . $someSecretKey . $someTimestamp . $someUserId; $sig .=
strlen($sig); return $sig = md5($sig); }

```

- **Node.js**

```

var crypto = require('crypto'); //Use for MD5 hashing //Generate
signature. Formula: (apiKey + secretKey + ts + userId) +
lengthOfString var makeSignature = function(opt){ let signature =
someApiKey + someSecretKey + someTimestamp + someUserId; signature =
signature.concat(signature.length); return
crypto.createHash('md5').update(signature).digest('hex'); }

```

If you disable authentication on your site, you no longer need a timestamp (ts) or signature (sig) when you log in. This is true wherever you normally supply a signature, such as when you embed [widgets](#) or use the [JavaScript library](#). Authentication is disabled in [Site Settings](#).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |

| Name | Required | Type* | Default Value | Range | Comments |
|------------------|----------|--------|---------------|---------------|---|
| apiKey | YES | String | None | 0 - 255 chars | The site's API key. |
| firstName | No | String | None | 0 - 255 chars | User's first name to save on server. Overrides an old value. Use to avoid redundant API calls where this information is loaded separately. You can pass in optional firstName information for a user which is stored as a preference (named 'firstName') and can be looked up later using user.getPreference . Also returned as part of the response to most site.* methods, such as site.getPointsLeaders . You can put anything in this field, such as a username or email address. Can be used for a user's custom data that you don't want to have to look up later when rendering leaderboards and similar things. |
| lastName | No | String | None | 0 - 255 chars | User's last name to save on server. Overrides an old value. Use to avoid redundant API calls where this information is loaded separately. You can pass in optional lastName information for a user which is stored as a preference (named 'lastName') and can be looked up later using user.getPreference . Also returned as part of the response to most site.* methods, such as site.getPointsLeaders . You can put anything in this field, such as a username or email address. Can be used for a user's custom data that you don't want to have to look up later when rendering leaderboards and similar things. |
| sig | No | String | None | 0 - 255 chars | Signature = MD5 (apiKey, secretKey, ts, userId, length). Only required if medium or high security is enabled. |
| ts | No | String | None | 0 - 255 chars | UNIX timestamp. Timestamp is 10 digits, in seconds (for example, 1473284311). Needs to be within the last 72 hours to be valid. Only required if authentication is enabled. |
| userId | YES | String | None | 1 - 255 chars | The user's unique ID for the site. Must be alphanumeric and no special characters (can be |

| Name | Required | Type* | Default Value | Range | Comments |
|------|----------|-------|---------------|-------|---|
| | | | | | hashed). It's important to use a commonly known ID. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-------------------------|--|
| <code>expiresIn</code> | UNIX timestamp of the login expiration date. |
| <code>sessionKey</code> | The <code>sessionKey</code> parameter for traditional API calls. |

Sample Request

```
method=user.login&apiKey=100&userId=2000
```

Sample Response

```
{ "response": { "login": { "sessionKey": "2MjA5WN1OGQx8MA==",
"expiresIn": 2592000 } }, "responseType": {}, "isError": false,
"httpError": 0, "httpsStatusCode": 200 }
```

user.modifyUserId Method



As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Modifies the `userId` of a specific user to a new value. Typically this is used when an anonymous user is changed to a registered user or to change from an uncommon user ID to something more commonly known.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------------|----------|--------|---------------|---------------|---|
| <code>asyncToken</code> | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|--|
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| newUserId | YES | String | None | 1 - 255 chars | New ID to assign to the user. This ID must be unique and cannot already exist. |
| oldUserId | YES | String | None | 1 - 255 chars | User's current ID. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------------|--|
| isError | <ul style="list-style-type: none"> ▪ true - The user ID was not modified. ▪ false - The user ID was modified successfully. |

Sample Request

```
method=user.modifyUserId&asyncToken=&oldUserId=2000&newUserId=6000
```

Sample Response

```
{ "response": {}, "responseType": {}, "isError": false, "httpError": 0, "httpStatusCode": 200 }
```

Sample Request - Existing Username

```
method=user.modifyUserId&asyncToken=&oldUserId=2001&newUserId=4000
```

Sample Response

```
{ "response": { "error": { "code": 150, "message": "New UserID already exists: : newUserName='4000'" } }, "responseType": {}, "isError": true, "httpError": 0, "httpStatusCode": 400 }
```


Sample Request - Old Username Doesn't Exist

```
method=user.modifyUserId&asyncToken=&oldUserId=6000&newUserId=6006
```

Sample Response

```
{ "response": { "error": { "code": 140, "message": "Invalid userId: : oldUserName='6000' does not exist" } }, "responseType": {}, "isError": true, "httpError": 0, "httpStatusCode": 400 }
```

user.purchaseItem Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Purchases an item for the user.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|------------------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| itemId | YES | Long | None | 0 - 2 ³¹ -1 | The catalog item ID of the item to purchase. |
| quantity | No | String | 1 | 0 - 255 chars | The number of items to purchase. |
| userId | No | String | None | 0 - 255 chars | The ID of the user to purchase the item(s) for. If not specified, uses the user associated with the sessionKey. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------------|---|
| catalogItem | <p>The catalog items owned by the user:</p> <ul style="list-style-type: none"> ▪ background - Indicates if the item is a background item: <ul style="list-style-type: none"> ▪ 0 - Not a background item ▪ 1 - A background item |

| Response | Description |
|------------------------|--|
| | <ul style="list-style-type: none"> ▪ catalogItemId - The item's ID number. ▪ catalogName - The name of the catalog the item is in. ▪ category - The names of the categories the item is in. ▪ categoryIds - The ID's of the category the item is in. ▪ customData - The custom data associated with the item. ▪ description - The item's description. ▪ doCallback - Indicates if a callback is triggered when a user purchases the item: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes ▪ id - The item's ID number. ▪ lastPurchased - The date/time the item was last purchased (UNIX timestamp). ▪ maxOwnedCountPerUser - The maximum number of the item a user can purchase. ▪ name - The item's name. ▪ orders - The order the item appears in the catalog. ▪ pointCategoryId - The ID of the point category that points are deducted from when the user purchases this item. ▪ pointCategoryName - The name of the point category that points are deducted from when the user purchase this item. ▪ price - The cost of the item (in points). ▪ realItem - Indicates if the item shows in the user's Owned Items list in widgets: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes ▪ tags - The tags associated with the item. ▪ zOrder - The item's Z-Order number. |
| lifetimeBalance | The user's lifetime points balance in the default point category. |
| pointCategory | Point categories available to the user: <ul style="list-style-type: none"> ▪ id - The point category ID. ▪ name - The point category name. ▪ premium - Indicates if this is a premium points category: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes ▪ points - The number of points the user earned in the point category. |

| Response | Description |
|---------------------|--|
| | <ul style="list-style-type: none"> ▪ lifetimeBalance - The user's lifetime balance for the point category. ▪ iconUrl - For point categories with an associated image, the URL to the image. ▪ shortName - The point category's short name. ▪ isDefault - Returns true or false indicating if this is the default point category. |
| points | The user's point balance in the default point category. |
| pointsRecord | The purchase information as it displays in widgets: <ul style="list-style-type: none"> ▪ currentBalance - The user's current balance for the default point category. ▪ lifetimeBalance - The user's lifetime balance for the default point category. ▪ labelAbbrSingular - The abbreviated label that describes the user's current point balance (singular). ▪ labelAbbrPlural - The abbreviated label that describes the user's current point balance (plural). ▪ labelPlural - The label that describes the user's current point balance (plural). ▪ labelSingular - The label that describes the user's current point balance (singular). |
| userId | The user's ID that purchased the item. |

Sample Request


method=user.purchaseItem&itemId=31211596&quantity=1&userId=6000

Sample Response

```
{ "response": { "balance": { "pointCategories": { "pointCategory": [
{ "id": 295686, "name": "KudosPoints", "premium": 0, "points": 10,
"lifetimeBalance": 10, "iconUrl":
"https://images/7283d2f78eb/userUploadpng1575280ebaa.png",
"shortName": "Kudos", "isDefault": false }, { "id": 295169, "name":
"Points", "premium": 0, "points": 947, "lifetimeBalance": 947,
"isDefault": true } ] }, "userId": "6000", "points": 947,
"lifetimeBalance": 947 }, "pointsRecord": { "recordId":
"PointsRecord.6000", "currentBalance": 947, "lifetimeBalance": 947,
"labelSingular": "point", "labelPlural": "points",
"labelAbbrSingular": "pt", "labelAbbrPlural": "pts" },
"ownedItemsRecord": { "ownedItems": { "catalogItem": [ { "id":
111471123, "description": "", "customData": "", "background": 0,
"designer": "", "zOrder": 0, "name": "Shoes", "catalogName":
"Sports", "price": 0, "categoryIds": "2896057", "category": "Cheer",
"tags": "", "canUnselect": "true", "realItem": 0, "doCallback": 0,
"passExtraData": 0, "orders": "0", "pointCategoryId": 295169,
```

```
"pointCategoryName": "Points", "lastPurchased": "2017-10-03
10:38:41", "catalogItemId": 31211596, "maxOwnedCountPerUser": 0 }, {
"id": 111471122, "description": "", "customData": "", "background":
0, "designer": "", "zOrder": 0, "name": "Shoes", "catalogName":
"Sports", "price": 0, "categoryIds": "2896057", "category": "Cheer",
"tags": "", "canUnselect": "true", "realItem": 0, "doCallback": 0,
"passExtraData": 0, "orders": "0", "pointCategoryId": 295169,
"pointCategoryName": "Points", "lastPurchased": "2017-10-03
10:38:33", "catalogItemId": 31211596, "maxOwnedCountPerUser": 0 } ]
}, "recordId": "OwnedItemsRecord.6000" } }, "responseType": {},
"isError": false, "httpError": 0, "httpStatusCode": 200 }
```

user.removePreference Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Disables a [user preference](#) so that it never shows up in the list of user preferences for this user again. If the specified user doesn't exist, it's created.

Each user has a set of preferences that can be defined for them. You can use preferences to add additional information about the user, such as first name, last name, or other custom data. There are five special reserved preferences that Nitro uses:

- **firstName** - Displays the first name of the user in the profile, leaderboards, events feed, etc.
- **lastName** - Displays the last name of the user in the profile, leaderboards, events feed, etc.
- **userPhotoUrl** - Displays the photo/avatar image of the user in the profile and leaderboards.
- **role** - Displays the user's job title/role in the Profile widget.
- **anonymous** - Used with ANY value, including no value, to hide the user from leaderboards and events feeds.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|--|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the |

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| | | | | | response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| name | YES | String | None | 0 - 255 chars | Name of the preference to remove. |
| userId | YES | String | None | 1 - 255 chars | The user's unique ID for the site. Must be alphanumeric and no special characters (can be hashed). It's important to use a commonly known ID. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------------|--|
| res | <ul style="list-style-type: none"> ok - The preference was successfully removed. err - The preference was not removed. |
| userPreferences | Returns the name value pair of the user's remaining preferences. |


Sample Request

method=user.removePreference&asyncToken=&userId=4000&name=hireDate

Sample Response

```
{ "response": { "userPreferences": { "userPreference": [ { "name": "firstName", "value": "Vanna" }, { "name": "lastName", "value": "Darville" }, { "name": "userPhotoUrl", "value": "https://s3.amazonaws.com/images/dc749ad/userUploadpng15ef2fb.png" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpStatusCode": 200 }
```

user.resetLevel Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Resets the user's level to the lowest level. Only applies to custom [levels](#), not point-based levels.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| userId | No | String | None | 0 - 255 chars | The user's unique ID for the site. Must be alphanumeric and no special characters (can be hashed). It's important to use a commonly known ID. Defaults to the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------------|--|
| description | The level's description. |
| name | The name of the level the user is reset to. |
| timestamp | The time the level was reset (UNIX timestamp). |
| type | The level type. Returns custom. |
| userId | The ID of the user the level was reset for. |


Sample Request

```
method=user.resetLevel&asyncToken=&userId=4000
```

Sample Response

```
{ "response": { "users": { "user": [ { "siteLevel": { "name": "Basic", "description": "Level 1 - No training", "type": "custom", "timestamp": 0 }, "userId": "4000" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpStatusCode": 200 }
```


user.sellbackItem Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Sells an item. **Note:** Bunchball support is needed to enable sellback functionality. [Contact support](#) if you receive the "This site has not enabled sellback" message when using this API.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| itemId | No | Long | None | N/A | The catalog item ID of the item to sell back. If more than one of this item is owned, it selects a random one. |
| ownedItemId | No | Long | None | N/A | The owned item ID of the item to sell back. Must be specified if itemId is not. |
| quantity | No | String | 1 | 0 - 255 chars | The number of items to sell back if specifying itemId. |
| userId | No | String | None | 0 - 255 chars | The ID of the user selling back the item. If not specified, gets userId from sessionKey. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|---------------|---|
| background | Indicates if the item is a background item: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| catalogItemId | The item's ID number. |
| catalogName | The name of the catalog the item is in. |
| category | The name of the category the item is in. |

| Response | Description |
|-----------------------------|--|
| categoryIds | The ID's of the category the item is in. |
| customData | The metadata associated with the catalog item. |
| description | The item's description. |
| doCallback | Indicates if a callback is triggered when a user purchases the item: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| fullUrl | URL to the full size image associated with the item. |
| id | The catalog item ID. |
| lastPurchased | The date/time the item was last purchased. |
| maxOwnedCountPerUser | The maximum number of the item a user can purchase. |
| name | The item's name. |
| orders | The order the item appears in the catalog. |
| pointCategoryId | The ID of the point category that points are returned to when the user returns the item. |
| pointCategoryName | The name of the point category that points are returned to when the user returns the item. |
| price | The cost of the item (in points). |
| realItem | Indicates if the item shows in the user's Owned Items list in widgets: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| tags | The tags associated with the item. |
| thumbUrl | URL to the thumbnail image associated with the item. |
| zOrder | The item's Z-Order number. |

Sample Request


```
method=user.sellbackItem&itemId=30151594&asyncToken=&userId=10638&quantity=1
```

Sample Response

```
{ "response": { "ownedItemsRecord": { "ownedItems": { "catalogItem": [ { "id": 112388831, "description": "", "customData": "", "background": 0, "thumbUrl": "test.com", "designer": "", "zOrder": 0, "name": "test", "catalogName": "test", "fullUrl": "test.com",
```

```
"price": 0, "categoryIds": "2816586", "category": "Clothing", "tags":
"", "canUnselect": "true", "realItem": 0, "doCallback": 0,
"passExtraData": 0, "orders": "0", "pointCategoryId": 34963,
"pointCategoryName": "Points", "lastPurchased": "2018-03-06
08:43:37", "catalogItemId": 30151594, "maxOwnedCountPerUser": 0 }, {
"id": 112388830, "description": "", "customData": "", "background":
0, "thumbUrl": "test.com", "designer": "", "zOrder": 0, "name":
"test", "catalogName": "test", "fullUrl": "test.com", "price": 0,
"categoryIds": "2816586", "category": "Clothing", "tags": "",
"canUnselect": "true", "realItem": 0, "doCallback": 0,
"passExtraData": 0, "orders": "0", "pointCategoryId": 34963,
"pointCategoryName": "Points", "lastPurchased": "2018-03-06
08:43:37", "catalogItemId": 30151594, "maxOwnedCountPerUser": 0 } ]
}, "recordId": "OwnedItemsRecord.10638" } }, "responseType": {},
"isError": false, "httpError": 0, "httpStatusCode": 200 }
```

user.setLevel Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Sets the user's [level](#). Only valid when the site is using custom levels, not point-based levels.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|--|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| levelName | YES | String | None | 0 - 255 chars | The name of the level to assign to the user. |
| userId | No | String | None | 0 - 255 chars | The user's unique ID for the site. Must be alphanumeric and no special characters (can be hashed). It's important to use a commonly known ID. If not specified, defaults to the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------------|--|
| description | The level's description. |
| name | The name of the level the user is set to. |
| timestamp | The time the level was set (UNIX timestamp). |
| type | The level type. Returns custom. |
| userId | The ID of the user the level was set for. |

Sample Request

```
method=user.setLevel&asyncToken=&userId=4000&levelName=Intermediate
```

Sample Response

```
{ "response": { "users": { "user": [ { "siteLevel": { "name":
"Intermediate", "description": "Level 2 - 10 or fewer training
sessions", "type": "custom", "timestamp": 0 }, "userId": "4000" } ] }
}, "responseType": {}, "isError": false, "httpError": 0,
"httpStatusCode": 200 }
```

user.setPreference Method



As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Sets a **user preference** (also known as a segment in Nitro Studio). If the specified user doesn't exist, it is created.

Each user has a set of preferences that can be defined for them. You can use preferences to add additional information about the user, such as first name, last name, or other custom data. There are five special reserved preferences that Nitro uses:

- **firstName** - Displays the first name of the user in the profile, leaderboards, events feed, etc.
- **lastName** - Displays the last name of the user in the profile, leaderboards, events feed, etc.

- **userPhotoUrl** - Displays the photo/avatar image of the user in the profile and leaderboards.
- **role** - Displays the user's job title/role in the Profile widget.
- **anonymous** - Used with ANY value, including no value, to hide the user from leaderboards and events feeds.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|----------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| name | YES | String | None | 0 - 255 chars | Name of the preference. Alphanumeric only. No special characters. Limited to 128 characters. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| userId | No | String | None | 1 - 255 chars | The user's unique ID for the site. Must be alphanumeric and no special characters (can be hashed). It's important to use a commonly known ID. |
| value | No | String | None | 0 - 8096 chars | Value of the preference. Alphanumeric only. No special characters. Limited to 128 characters. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------|-------------------------|
| name | The preference name. |
| value | The preference's value. |


Sample Request

```
method=user.setPreference&asyncToken=&userId=4000&name=hireDate&value=March 9
```

Sample Response

```
{ "response": { "userPreferences": { "userPreference": [ { "name": "hireDate", "value": "March 9" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpStatusCode": 200 }
```

user.setPreferences Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Sets multiple [user preferences](#) (also known as a segment in Nitro Studio). If the specified user doesn't exist, it is created.

Each user has a set of preferences that can be defined for them. You can use preferences to add additional information about the user, such as first name, last name, or other custom data. There are five special reserved preferences that Nitro uses:

- **firstName** - Displays the first name of the user in the profile, leaderboards, events feed, etc.
- **lastName** - Displays the last name of the user in the profile, leaderboards, events feed, etc.
- **userPhotoUrl** - Displays the photo/avatar image of the user in the profile and leaderboards.
- **role** - Displays the user's job title/role in the Profile widget.
- **anonymous** - Used with ANY value, including no value, to hide the user from leaderboards and events feeds.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |

| Name | Required | Type* | Default Value | Range | Comments |
|---------------|----------|---------------------|---------------|---------------|--|
| names | YES | Pipe-separated list | None | 0 - 100 items | Pipe-separated list of preference names. Limited to 128 characters. |
| userId | No | String | None | 1 - 255 chars | The user's unique ID for the site. Must be alphanumeric and no special characters (can be hashed). It's important to use a commonly known ID. |
| values | No | Pipe-separated list | None | 0 - 100 items | Pipe-separated list of preference values. The number of entries here should match the number of entries in the names parameter. Limited to 128 characters. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------|-------------------------|
| name | The preference name. |
| value | The preference's value. |


Sample Request

```
method=user.setPreferences&values=February
11|Telecom|Phoenix&asyncToken=&userId=6000&names=hireDate|Division|Lo
cation
```

Sample Response

```
{ "response": { "userPreferences": { "userPreference": [ { "name":
"Division", "value": "Telecom" }, { "name": "hireDate", "value":
"February 11" }, { "name": "Location", "value": "Phoenix" }, {
"name": "SiteID", "value": "SanDiego" } ] } }, "responseType": {},
"isError": false, "httpError": 0, "httpStatusCode": 200 }
```

user.storeNotifications Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Stores one or more [notifications](#) for one or more users. The stored notifications are shown to users without requiring the user to log any actions or complete any challenges. No triggers are necessary. You can store a notification for any user at any time.

Note: Runtime tokens that are trigger-based do not work when you store notifications using this method because there are no actions or challenges involved.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------------|----------|----------------------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| notificationNames | YES | Comma-separated list | None | 0 - 255 items | A comma-separated list of notification names to store. Notifications must be created in Nitro Studio and referenced here. |
| userIds | No | Generic list | None | 0 - 255 items | A comma-separated list of users to store notifications for. Defaults to the authenticated user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------|---|
| ok | The notification was successfully stored. |
| err | The notification was not stored. |

Sample Request

```
method=user.storeNotifications&asyncToken=&userIds=6000&notificationNames=Good Job
```


Sample Response

```
{ "response": {}, "responseType": {}, "isError": false, "httpError": 0, "httpStatusCode": 200 }
```


Sample Request - Invalid Notification Name

```
method=user.storeNotifications&asyncToken=&userIds=4000&notificationNames=Super
```

Sample Response

```
{ "response": { "error": { "code": 219, "message": "No valid notification names specified" } }, "responseType": {}, "isError": true, "httpError": 0, "httpStatusCode": 400 }
```

user.transferPoints Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Transfers current and lifetime point balances in all [point categories](#) from the source user to a destination user, and resets the points of the source user. The "New User Points" of the source user are not transferred. Bypasses the daily maximum credit limit.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| destUserId | YES | String | None | 1 - 255 chars | Unique ID of user to transfer points to. This user may or may not exist in system. |
| srcUserId | YES | String | None | 1 - 255 chars | Unique ID of user to transfer points from. This user should already exist in system. Parameter resets after this operation. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------|--|
| firstName | The user's first name |
| lastName | The user's last name. |
| userId | The ID of the user that received the points. |

Sample Request

```
method=user.transferPoints&asyncToken=&destUserId=2000&srcUserId=2002
```

Sample Response

```
{ "response": { "user": { "userId": "2000", "firstName": "",
"lastName": "" } }, "responseType": {}, "isError": false,
"httpError": 0, "httpStatusCode": 200 }
```


Sample Request - Source Username Doesn't Exist

```
method=user.transferPoints&asyncToken=&destUserId=2000&srcUserId=8000
```

Sample Response

```
{ "response": { "error": { "code": 203, "message": "Source user does
not exist, userId: 8000" } }, "responseType": {}, "isError": true,
"httpError": 0, "httpStatusCode": 400 }
```

group.getChallengeProgress Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Gets the progress a group has made towards completing a challenge. (For individual user progress, see [user.getChallengeProgress](#).) The return value lists the rules associated with the challenge, the goal, the value achieved by the group, and if the challenge has been completed.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|----------------------------|----------|---------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| challengeName | No | String | None | 0 - 255 chars | The challenge name. |
| folder | No | String | None | 0 - 255 chars | If set, retrieves only the challenges in that folder. |
| groupName | YES | String | None | 1 - 255 chars | The group name. |
| isFeatured | No | Boolean | None | true/false | If true, returns only challenges that have the featured flag set to true. If not specified, all challenges are returned. |
| showServiceTypeInfo | No | Boolean | None | 1 - 255 chars | Set to false to improve performance. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------------------------|---|
| applyMultiplier | The points multiplier in use for the challenge. |
| canAchieveChallenge | Returns true or false indicating if the group is eligible to participate in the challenge. |
| catalogItemName | For challenges with item rewards, returns the catalog item name. |
| catalogItemThumbUrl | For challenges with item rewards, returns the URL of the catalog item. |
| ChallengeRewards | <p>The rewards associated with the challenge:</p> <ul style="list-style-type: none"> ▪ type - The reward type. Returns point, item, or notification. ▪ details - Additional reward information. For points, the details may be something like "285424 50" with the number of points awarded after |

| Response | Description |
|------------------------------|--|
| | <p>the pipe (). In this example, 50 points are awarded to the user after completing the challenge.</p> <ul style="list-style-type: none"> ▪ itemName - The name of the point category, catalog item, or notification. |
| completionCount | The number of times the challenge has been completed. |
| dailyAchievementLimit | The number of times the challenge can be completed in a day. Returns 0 for unlimited. |
| dateIssued | The date the challenge was activated. |
| description | The challenge description. |
| endTime | The date/time the challenge ends (UNIX timestamp). |
| featured | Returns true or false indicating if the challenge is a featured challenge. |
| folderName | The name of the folder the challenge is located in. |
| fullUrl | For challenges with badge rewards, returns the URL of the full size badge image. |
| groupFlag | <p>Indicates if the challenge is a group challenge:</p> <ul style="list-style-type: none"> ▪ 0 - Not a group challenge ▪ 1 - A group challenge |
| groupPointAward | The number of points awarded for this challenge. |
| hideUntilEarned | <p>Indicates if the challenge is hidden until earned:</p> <ul style="list-style-type: none"> ▪ 0 - Not hidden ▪ 1 - Hidden |
| id | The challenge ID. |
| name | The challenge name. |
| pointAward | The number of points awarded for completing the challenge. |
| pointCategory | For challenges with point rewards, returns the name of the point category the points are awarded to. |
| pointCategoryId | The point category ID. |
| preferences | Used for metric goals where a segment determines which value to use in the goal. |
| repeatable | <p>Indicates if the challenge is repeatable:</p> <ul style="list-style-type: none"> ▪ 0 - Not repeatable ▪ 1 - Repeatable |

| Response | Description |
|--------------------|---|
| | Repeatable, hidden challenges show in both complete and incomplete once initially completed. |
| rules | <ul style="list-style-type: none"> ▪ Metadata - The metadata associated with the rule. Returns name and value pair. ▪ actionTag - The name of the action associated with the rule. ▪ operator - The action goal value: <ul style="list-style-type: none"> ▪ GE - Greater than or equal to ▪ GT - Greater than ▪ LT - Less than ▪ LE - Less than or equal to ▪ EQ - Equal to ▪ completed - Returns true or false indicating if the action has been completed. ▪ goal - The action value. For Less than scenarios, the action value lists this value as the number configured in Nitro Studio. Must be used with operator for Less than scenarios. ▪ achieved - Indicates if the action was completed: <ul style="list-style-type: none"> ▪ 0 - Not completed ▪ 1 - Completed ▪ type - Indicates the action goal type: <ul style="list-style-type: none"> ▪ sum - Running total ▪ count - Number of calls ▪ sortOrder - An integer value indicating the rules' order within the challenge. ▪ displayOrder - The order the rule is shown in the challenge. ▪ serviceType - Previously used for social actions. ▪ serviceActionType - Previously used for social actions. ▪ prereqRuleIds - The ID of the rule that must be completed before this rule (for multi-rule challenges). ▪ prereqOperator - Indicates if one or all prerequisites must be met. Returns and or or. |
| serviceType | Previously used for social actions. |
| startTime | The date/time the challenge starts (UNIX timestamp). |
| tags | The tags associated with the challenge. |

| Response | Description |
|-----------------|--|
| thumbUrl | For challenges with badge rewards, returns the URL of the thumb badge image. |


Sample Request

```
method=group.getChallengeProgress&asyncToken=&folder=&challengeName=Award%20Kudos&groupName=NewYork
```

Sample Response

```
{ "response": { "challenges": { "challenge": [ { "rules": { "rule": [
{ "metadata": {}, "actionTag": "RESUME_SUBMITTED" "operator": "GT",
"completed": "false", "goal": 15, "timeRange": 3600, "achieved": 4,
"type": "count", "sortOrder": 0, "displayOrder": 0, "serviceType":
"nitro", "serviceActionType": "none" } ] }, "challengeRewards": {
"challengeReward": [ { "type": "point", "details": "295682|25",
"itemName": "Funny Money" }, { "type": "item", "details": "31208705",
"itemName": "Logo2" }, { "type": "notification", "details": "710529",
"itemName": "Impressive" } ] }, "id": 4802284, "name": "Award Kudos",
"description": "Employees acknowledging the work of other
employees.", "activeFlag": 1, "featuredFlag": 0, "serviceType": "1",
"catalogItemName": "Logo2", "catalogItemThumbUrl":
"C:\\Users\\Documents\\ImageFiles\\logo2.png", "pointCategory":
"Funny Money", "thumbUrl":
"https://s3.amazonaws.com/images/7284f78eb/userUploadpng157a11.png",
"fullUrl":
"https://s3.amazonaws.com/images/7284f78eb/userUploadpng1571ae3.png",
"pointAward": 25, "dateIssued": "1473442620", "dateCompleted":
"1474386618", "applyMultiplier": 0, "repeatable": 0,
"hideUntilEarned": 0, "pointCategoryId": 295682, "completionCount":
1, "tags": "achievement", "startTime": 1473442620, "endTime": 0,
"ruleMatchType": 0, "prereqMatchType": "or", "preferences": "",
"preferenceValues": "", "preferencesMatchType": "or", "groupFlag": 1,
"groupPointAward": 0, "dailyAchievementLimit": 0,
"canAchieveChallenge": true, "folderName": "Collaboration" } ] } },
"responseType": {}, "isError": false, "httpError": 0,
"httpStatusCode": 200 }
```

group.getPreference Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Gets an individual group preference. To get all preferences for a group, use [group.getPreferences](#).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| groupName | YES | String | None | 1 - 255 chars | The group to get the preference for. |
| name | YES | String | None | 0 - 255 chars | The preference name. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------|-----------------------|
| name | The preference name. |
| value | The preference value. |


Sample Request

method=group.getPreference&asyncToken=&name=NY&groupName=NewYork

Sample Response

```
{ "response": { "groupPreferences": { "groupPreference": [ { "name": "NY", "value": "NY" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpStatusCode": 200 }
```

group.getPreferences Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Gets multiple preferences for a given group. To get an individual preference, use [group.getPreference](#).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|---------------------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| groupName | YES | String | None | 1 - 255 chars | The group to get the preference for. |
| names | No | Pipe-separated list | None | 0 - 100 items | Pipe-separated list of preference names. If not specified, all preferences are returned. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------|-----------------------|
| name | The preference name. |
| value | The preference value. |


Sample Request

```
method=group.getPreferences&asyncToken=&names=&groupName=NewYork
```

Sample Response

```
{ "response": { "groupPreferences": { "groupPreference": [ { "name": "groupName", "value": "NewYork" }, { "name": "NY", "value": "NY" }, { "name": "Pref1", "value": "Value1" }, { "name": "Pref2", "value": "Value2" }, { "name": "Pref3", "value": "Value3" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpStatusCode": 200 }
```


group.getUsers Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Returns users belonging to a certain group.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|---------------------|---------------|---------------|--|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| groupName | YES | String | None | 0 - 255 chars | Group to retrieve users from. |
| preferences | No | Pipe-separated list | None | 0 - 3 items | Pipe-separated list of the keys in your site preferences key:value pairs to include in the response for each user returned (Nitro Studio > Configuration > Site Settings > Preferences). |
| returnCount | No | Integer | 10 | 1 - 100 | The number of records to return. |
| startIndex | No | Integer | 0 | 0 - 5000 | A 0-based index specifying which row from the results to return data from. Can be used to page through the result set. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------------|-------------------------|
| userId | The user ID. |
| userPreferences | The user's preferences. |


Sample Request

```
method=group.getUsers&asyncToken=&startIndex=0&returnCount=3&preferences=&groupName=NewYork
```

Sample Response

```
{ "response": { "users": { "user": [ { "userPreferences": {},
"userId": "1000" }, { "userPreferences": {}, "userId": "1001" }, {
"userPreferences": {}, "userId": "2000" } ] } }, "responseType": {},
"isError": false, "httpError": 0, "httpStatusCode": 200 }
```

group.removePreference Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Deletes a group preference.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| groupName | YES | String | None | 1 - 255 chars | The user's unique ID on your site. |
| name | YES | String | None | 0 - 255 chars | Name of the preference to remove. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------------------------|--|
| DeleteSuccess or err | Indicates the preference was successfully removed or errors when the preference was not found. |
| id | The deleted preference's ID. |
| name | The deleted preference's name. |
| object | The object name. |


Sample Request

```
method=group.removePreference&asyncToken=&name=Pref1&groupName=Arizona
```

Sample Response

```
{ "response": { "deleteSuccess": { "object": "GroupPreference",
"name": "Pref1", "id": 34364593 } }, "responseType": {}, "isError":
false, "httpError": 0, "httpStatusCode": 200 }
```

group.setPreference Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Sets a group preference. To set multiple preferences, use [group.setPreferences](#).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|----------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| groupName | YES | String | None | 1 - 255 chars | The group to set the preference for. |
| name | YES | String | None | 0 - 255 chars | Name of the preference. |
| value | No | String | None | 0 - 2048 chars | Value of the preference. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-------------|----------------------|
| name | The preference name. |

| Response | Description |
|----------|-----------------------|
| value | The preference value. |


Sample Request

```
method=group.setPreference&asyncToken=&name=PrefName&value=PrefValue&
groupName=NewYork
```

Sample Response

```
{ "response": { "groupPreferences": { "groupPreference": [ { "name":
"PrefName", "value": "PrefValue" } ] } }, "responseType": {},
"isError": false, "httpError": 0, "httpStatusCode": 200 }
```

group.setPreferences Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Sets multiple group preferences. To set a single preference, see [group.setPreference](#).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|---------------------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| groupName | YES | String | None | 1 - 255 chars | The user's unique ID on your site. |
| names | YES | Pipe-separated list | None | 0 - 100 items | Pipe-separated list of preference names. |
| values | No | Pipe-separated list | None | 0 - 100 items | Pipe-separated list of preference values. The number of entries here should match the number of entries in the names parameter. |

*[Type Descriptions](#)

Response Parameters

Returns a list of all the group's preferences.

| Response | Description |
|--------------|-----------------------|
| name | The preference name. |
| value | The preference value. |


Sample Request

```
method=group.setPreferences&asyncToken=&values=Value1|Value2|Value3&names=Pref1|Pref2|Pref3&groupName=Arizona
```

Sample Response

```
{ "response": { "userPreferences": { "userPreference": [ { "name": "CaveCreek", "value": "Cave Creek" }, { "name": "groupName", "value": "Arizona" }, { "name": "Phx", "value": "Phoenix" }, { "name": "Pref1", "value": "Value1" }, { "name": "Pref2", "value": "Value2" }, { "name": "Pref3", "value": "Value3" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpStatusCode": 200 }
```

site.addUsersToGroup Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Adds multiple users to a group. If the group or users do not exist, they are created for you.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| groupName | YES | String | None | 1 - 255 chars | The group name. If the group doesn't exist, it may be created automatically based on your |

| Name | Required | Type* | Default Value | Range | Comments |
|----------------|----------|--------------|---------------|--------------|--|
| | | | | | Site Settings (Configuration > Site Settings > General > Groups > Automatically create groups when a user joins). |
| userIds | YES | Generic list | None | 0 -100 items | A comma-separated list of unique user IDs. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------|--|
| groupName | The name of the group users were added to. |
| userName | The ID of the user added to the group. |

Sample Request

```
method=site.addUsersToGroup&asyncToken=&userIds=1000,1001&groupName=NewYork
```

Sample Response

```
{ "response": { "groupUsers": { "groupUsers": [ { "userName": "1001" }, { "userName": "1000" } ], "groupName": "NewYork" } }, "responseType": {}, "isError": false, "httpError": 0, "httpStatusCode": 200 }
```


Sample Request - Invalid Group Name

```
method=site.addUsersToGroup&asyncToken=&userIds=1006,1007,1008&groupName=Arizona
```

Sample Response

```
{ "response": { "error": { "code": 120, "message": "Invalid Parameter: Group does not exist, set autoCreateGroup to true for site preferences to auto create the group up on request" } }, "responseType": {}, "isError": true, "httpError": 0, "httpStatusCode": 400 }
```

site.getActionFeed Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Returns a newsfeed (XML or JSON) with the achievements earned by users in the program. Typical achievements displayed are challenges completed and level changes. Site-wide feeds are updated every five minutes. However, newsfeeds are cached so there is often a slight delay on updates.

Note: Bunchball only guarantees that the feed contains challenges completed within the last three months.

Setup

| | |
|--------------------------|--|
| | |
| <p>Challenges</p> | <p>Challenges are completed as the result of actions, and therefore have to have their feed entries set up in advance in Nitro Studio. To have a specific challenge returned, an action phrase needs to be set in the challenge's Action Phrase field.</p> <p>Action phrase is a phrase that describes the action being done. It supports variables that are replaced at runtime with actual values. Variable substitution works as follows:</p> <ul style="list-style-type: none"> ▪ "{username}" is substituted with "first_name + space + last_name", and hyperlinked (with an HTML "a" anchor tag) using the Userlink URL specified for the site (Configuration > Site Settings > General > Users > Userlink URL field). ▪ "{userId}" is substituted with the user's unique ID. ▪ If a URL is specified for the challenge, it's used as the hyperlink for the part of the action phrase enclosed between [square brackets]. <p>Show me an example</p> <p>If the have a challenge with the following:</p> <ul style="list-style-type: none"> ▪ Content = "{username} made a [post to the forums]." ▪ URL="http://www.site.com/forums/post" ▪ Userlink URL=http://www.site.com/users/{userId} <p>When the challenge is achieved, this is returned as HTML: John Doe made a post to the forums.</p> |
| <p>Levels</p> | <p>To return a newsfeed entry when a user changes levels, an action phrase must be set in the level's Action phrase field. The action phrase is returned in the feed when a new level is achieved.</p> |

| | |
|----------------|--|
| Items | To return a newsfeed entry when a user purchases or gifts an item, you must configure and use callbacks. |
| Actions | <p>In user.logAction, a newsfeed parameter can be supplied that can be any arbitrary HTML. If a value is specified, then this newsfeed entry is added to the feed. If no value is specified, then nothing shows up for that action in the feed. No variable replacement occurs in this newsfeed entry, it is passed through "as is". Because you can pass arbitrary HTML, you can put very detailed information and hyperlinks into action feed entries.</p> <p>Note: Bunchball does not recommend displaying actions in the newsfeed as the volume of actions decreases the value of newsfeed items.</p> |

Arguments

| Name | Required | Type | Default Value | Range | Comments |
|--------------------|----------|---------------------|---------------|---------------|--|
| apiKey | YES | String | None | 0 - 255 chars | Site to get the feed for. |
| groupName | No | String | None | 0 - 255 chars | <p>Only show newsfeed entries for members of the specified group. If showGroupChallenges is set to true, returns challenges completed by this group.</p> <p>Limited to one group.</p> <p>Note: You cannot specify both userIds and a groupName.</p> |
| preferences | No | Pipe-separated list | None | 0 - 3 items | <p>Allows you to specify a key for data to include in the response for each user returned. Items are case-sensitive. Reserved preferences include: firstName, lastName, and userPhotoUrl.</p> <p>If this parameter is specified, content is</p> |

| Name | Required | Type | Default Value | Range | Comments |
|--------------------------------|----------|---------|---------------|------------|--|
| | | | | | returned as an xml attribute. |
| returnCount | No | Integer | 20 | 1 - 50 | Maximum number of values to return. |
| showActions | No | Boolean | true | true/false | If set to true, returns events a user has logged. Also returns challenges completed as a result of the logged action if that challenge has an action phrase . |
| showChallengesCompleted | No | Boolean | true | true/false | If set to true, returns completed challenges that have an action phrase set in Nitro Studio. |
| showComments | No | Boolean | false | true/false | If set to true, returns comments a user receives. |
| showGifts | No | Boolean | false | true/false | If set to true, use callbacks to have the newsfeed return the gifts a user has given to another user. |
| showGroupChallenges | No | Boolean | false | true/false | If set to true, and a groupName is specified, returns the action phrases configured for group challenges completed by users within that group. Note: Only challenges with an action phrase defined show in the newsfeed. |
| showLevelChanges | No | Boolean | false | true/false | If set to true, returns the level changes a user has made. The text in the feed is determined by the action phrase set for the specific level (Reward > Levels > level name > Action phrase field). If not specified, the |

| Name | Required | Type | Default Value | Range | Comments |
|-----------------------|----------|--------------|---------------|---------------|---|
| | | | | | text defaults to "{userName} just leveled {direction} to {levelName}," where the text in braces is replaced at runtime by tokens . |
| showOwnedItems | No | Boolean | false | true/false | If set to true, use callbacks to have the newsfeed include users who have recently bought items. |
| userIds | No | Generic list | None | 0 - 100 items | A comma-separated list of users to get the feed for. If blank, all users are returned. Note: You cannot specify both userIds and a groupName. |

Response Parameters

| Response | Description |
|------------------------|--|
| content | The action phrase. |
| firstName | The first name of the user completing the action. |
| iconUrl | The name and location of the image associated with the action, level, or challenge. |
| itemName | The action, level, or challenge name. |
| lastName | The last name of the user completing the action. |
| note | For level changes, indicates if the user moved UP or DOWN a level. |
| ts | The time the action was completed (UNIX timestamp). |
| type | The action type: <ul style="list-style-type: none"> ▪ Challenge - The user completed a challenge. ▪ Level - The user moved up or down a level. |
| UserId | The ID of the user completing the action. |
| UserPreferences | Returns the value of the key specified in the preferences parameter. |

Sample Request

```
method=site.getActionFeed&showLevelChanges=true&apiKey=xxxx&showComments=true&groupName=&userIds=&returnCount=20&showGifts=false&showActions=true&showOwnedItems=true&showGroupChallenges=false&showChallengesCompleted=true&preferences=
```

Sample Response

```
{ "response": { "items": { "entry": [ { "userPreferences": {}, "content": "Great job achieving Padawan", "type": "Level", "ts": "1513633289", "iconUrl": "filelocation", "userId": "2000", "firstName": "David", "lastName": "Brown", "itemName": "Padawan" }, { "UserPreferences": {}, "content": "Rob Smith helped all of us in a small but significant way.", "type": "Challenge", "ts": "1473706777", "iconUrl": "filelocation\\MrClean.png", "userId": "7000", "firstName": "Rob", "lastName": "Smith", "itemName": "Beautification" }, { "userPreferences": {}, "content": "Passed the Something about nothing Quiz!", "type": "Challenge", "ts": "1513631651", "userId": "5214", "firstName": "Sanford", "lastName": "Ross", "itemName": "Pass the Something about nothing Quiz" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```


Sample Request - Group Name Not Specified

```
method=site.getActionFeed&showLevelChanges=false&apiKey=xxxx&showComments=false&groupName=&userIds=&returnCount=20&showGifts=false&showActions=true&showOwnedItems=false&showGroupChallenges=true&showChallengesCompleted=true&preferences=
```

Sample Response

```
{ "response": { "error": { "code": 120, "message": "Invalid Parameter: You must specify a group when showGroupChallenges is set" } }, "responseType": {}, "isError": true, "httpError": 0, "httpsStatusCode": 400 }
```

site.getActionLeaders Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Enables you to query the action history to generate site-wide leaderboards for any action data. This returns a list of users that are determined by the criteria that you supply.

Use cases for this method may be:

- Users who watched the most videos (criteria = SUM).
- Users who invited the most friends in a single session (criteria = MAX).
- Users with the most forum posts (criteria = COUNT or SUM).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| criteria | YES | String | None | 0 - 255 chars | The criteria to use to generate the table. Possible values are: <ul style="list-style-type: none"> ▪ MAX - Returns the users with the highest recorded action values. ▪ MIN - Returns the users with the lowest recorded action values. ▪ SUM - Returns the users with the highest cumulative values. ▪ COUNT - Returns the users with the most number of actions logged. Does not use action value. ▪ AVERAGE - Returns users with the highest average values. |
| duration | No | String | DAY | 0 - 255 chars | Works in connection with the start parameter to determine duration. |

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|---------------------|---------------|---------------|--|
| | | | | | <ul style="list-style-type: none"> ▪ DAY - Shows actions completed from 12:00 am EST on the start date. ▪ WEEK - Shows actions completed from 12:00 am EST on the Sunday from the week of the start date. ▪ MONTH - Shows actions completed from 12:00 am EST on the first day of the month of the start date. ▪ QUARTER - Shows actions completed from 12:00 am EST of the first day of the month of the start date. The leaderboard runs for 90 days. ▪ YEAR - Shows actions completed from 12:00 am EST on the first day of the start date year. ▪ ALLTIME - The start date is ignored. |
| groupName | No | String | None | 0 - 255 chars | <p>Limited to 1 group. Only returns users in the specified group.</p> <p>Note: You cannot specify both userIds and a groupName unless withRank=true is also passed. In addition, if withSurroundingUsers=true, you get a contextual leaderboard for just the specified group.</p> |
| preferences | No | Pipe-separated list | None | 0 - 3 items | Returns up to 3 group preferences for each group in the leaderboard. |
| returnCount | No | Integer | 10 | 1 - 100 | The number of values to return. |

| Name | Required | Type* | Default Value | Range | Comments |
|-----------------------------|----------|----------------------|---------------|---|---|
| start | No | Integer | None | 0 - 2 ³¹ -1 | UNIX timestamp. Sets the start date that works in connection with the duration parameter. All times are Eastern Standard Time. |
| tags | YES | Comma-separated list | None | 0 - 5 items <u>Truncated</u> 0 - 255 chars per item | A comma-separated list of tags (either action tags or regular tags) to filter on. Filters the list of actions from which to base the leaderboard on. |
| tagsOperator | No | Operator | OR | AND/OR | Actions that have any of the tags are considered. If set to AND, only actions that have all of the tags are considered. |
| userIds | No | Generic list | None | 0 - 100 items <u>Truncated</u> | A comma-separated list of users to generate the leaderboard for. All other users are ignored. When withRank is true, only one userId is allowed. Note: You cannot specify both userIds and a groupName unless withRank=true is also passed. In addition, if withSurroundingUsers=true, you get a contextual leaderboard for just the specified group. |
| withRank | No | Boolean | false | true/false | If true, returns the specified user's rank. Must pass a value in the userIds parameter. Limited to 1 userId. Works with any point category or duration specified. |
| withSurroundingUsers | No | Boolean | false | true/false | When used with userIds and withRank parameters, returns a contextual leaderboard for the specified user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------------|--|
| firstName | The user's first name. |
| lastName | The user's last name. |
| userId | The user's Nitro ID. |
| UserPreferences | Returns the value of the key specified in the preferences parameter. |
| value | The action value. |


Sample Request

```
method=site.getActionLeaders&asyncToken=&preferences=&tags=Clean,comment_document&userIds=&tagsOperator=OR&returnCount=10&start=&groupName=NewYork&criteria=MAX&duration=ALLTIME
```

Sample Response

```
{ "response": { "leaderboardId": 82140385, "actions": { "action": [ {
"userPreferences": {}, "userId": "2000", "firstName": "David",
"lastName": "Brown", "value": "2" }, { "userPreferences": {},
"userId": "1000", "firstName": "Vanna", "lastName": "Darville",
"value": "0" }, { "userPreferences": {}, "userId": "1001",
"firstName": "Julie", "lastName": "Wildes", "value": "0" }, {
"userPreferences": {}, "userId": "6000", "value": "0" } ] } },
"responseType": {}, "isError": false, "httpError": 0,
"httpsStatusCode": 200 }
```

site.getActionTargetLeaders Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Generates a leaderboard for the targets that have been rated the highest or most often. This can only be called with action tags.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|---------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| criteria | YES | String | None | 0 - 32 chars | The criteria to rank the leaderboard by, must be AVERAGE or COUNT. |
| returnCount | No | Integer | 10 | 1 - 100 | Number of leaders to return. |
| tag | YES | String | None | 0 - 255 chars | The name of the tag to get leaders on. |
| target | No | String | None | 0 - 255 chars | Associates the action with a target. Often used for API calls that allow you to filter by target. ✓ Best Practice Not recommended for use due to low program value. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------|-------------------|
| name | The target name. |
| value | The target value. |


Sample Request

```
method=site.getActionTargetLeaders&asyncToken=&criteria=COUNT&tag=comment_document
```

Sample Response

```
{ "response": { "targetLeaders": {} }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```


site.getCatalog Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Gets all the items in the specified catalog. If no catalog is specified, will return all items for all catalogs.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| catalogName | No | String | None | 0 - 255 chars | The catalog to return. If none specified, returns all catalogs. |
| tag | No | String | None | 0 - 255 chars | The tag to filter on. If none specified, all items are returned. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------------|--|
| category | The category details within the catalog: <ul style="list-style-type: none"> ▪ canUnselect - Returns true or false. ▪ catalogName - The name of the catalog. ▪ childCategoryIds - The child categories' IDs (when nested categories exist). ▪ createTime - The date/time the category was created (UNIX timestamp). ▪ id - The category ID. ▪ name - The category name. |
| catalogItem | The item details within the catalog: <ul style="list-style-type: none"> ▪ background - Indicates if the item is a background item: <ul style="list-style-type: none"> ▪ 0 - Not a background item ▪ 1 - A background item |

| Response | Description |
|----------|--|
| | <ul style="list-style-type: none"> ▪ catalogItemId - The ID of the item. ▪ catalogName - The name of the catalog the item is in. ▪ category - The names of the categories the item is in. ▪ categoryIds - The ID's of the category the item is in. ▪ countRemaining - The number of items remaining for purchase. ▪ createTime - The date/time the item was created in the catalog (UNIX timestamp). ▪ customData - The custom data associated with the item. ▪ description - The item's description. ▪ doCallback - Indicates if a callback is triggered when a user purchases the item: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes ▪ id - The item's ID number. ▪ lastUpdated - The date/time the item was last updated (UNIX timestamp). ▪ maxOwnedCountPerUser - The maximum number of the item a user can purchase. ▪ name - The item's name. ▪ orders - The order the item appears in the catalog. ▪ ownedCount - The number of items of this type that users own. ▪ pointCategoryId - The ID of the point category that points are deducted from when the user purchases this item. ▪ pointCategoryName - The name of the point category that points are deducted from when the user purchase this item. ▪ price - The cost of the item (in points). ▪ realItem - Indicates if the item shows in the user's Owned Items list in widgets: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes ▪ tags - The tags associated with the item. ▪ zOrder - The item's Z-Order number. |


Sample Request

```
method=site.getCatalog&asyncToken=&catalogName=sports
```

Sample Response

```
{ "response": { "catalogRecord": [ { "categories": { "category": [ {
"id": 2896056, "name": "Football", "catalogName": "Sports", "order":
1, "canUnselect": "true", "childCategoryIds": "", "createTime":
1482429824 }, { "id": 2896057, "name": "Cheer", "catalogName":
"Sports", "order": 1, "canUnselect": "true", "childCategoryIds": "",
"createTime": 1482429952 } ] }, "catalogItems": { "catalogItem": [ {
"id": 31211595, "description": "", "customData": "", "background": 0,
"designer": "", "zOrder": 0, "countRemaining": -1, "name": "Shoes",
"catalogName": "Sports", "price": 0, "categoryIds": "2896057",
"category": "Cheer", "tags": "", "canUnselect": "null", "realItem":
0, "doCallback": 0, "passExtraData": 0, "ownedCount": 0, "orders":
"0", "createTime": 1482437760, "pointCategoryId": 295169,
"pointCategoryName": "Points", "lastUpdated": "1482437760",
"catalogItemId": 31211595, "maxOwnedCountPerUser": 0 } ] }, "name":
"Sports" } ] }, "responseType": {}, "isError": false, "httpError": 0,
"httpsStatusCode": 200 }
```

site.getCatalogItem Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Gets information about a catalog item.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|------------------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| itemId | YES | Long | None | 0 - 2 ³¹ -1 | The ID of the catalog item to get. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-------------|---|
| catalogItem | <p>The item details within the catalog:</p> <ul style="list-style-type: none"> ▪ background - Indicates if the item is a background item: <ul style="list-style-type: none"> ▪ 0 - Not a background item ▪ 1 - A background item ▪ catalogItemId - The item's ID number. ▪ catalogName - The name of the catalog the item is in. ▪ category - The names of the categories the item is in. ▪ categoryIds - The ID's of the category the item is in. ▪ countRemaining - The number of items remaining for purchase. ▪ createTime - The date/time the item was created in the catalog (UNIX timestamp). ▪ customData - The custom data associated with the item. ▪ description - The item's description. ▪ doCallback - Indicates if a callback is triggered when a user purchases the item: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes ▪ fullUrl - The URL to the full size image of the item. ▪ id - The item's ID number. ▪ lastUpdated - The date/time the item was last updated (UNIX timestamp). ▪ maxOwnedCountPerUser - The maximum number of the item a user can purchase. ▪ name - The item's name. ▪ orders - The order the item appears in the catalog. ▪ ownedCount - The number of items of this type that users own. ▪ pointCategoryId - The ID of the point category that points are deducted from when the user purchases this item. ▪ pointCategoryName - The name of the point category that points are deducted from when the user purchase this item. ▪ price - The cost of the item (in points). ▪ realItem - Indicates if the item shows in the user's Owned Items list in widgets: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |

| Response | Description |
|-------------|--|
| | <ul style="list-style-type: none"> ▪ tags - The tags associated with the item. ▪ thumbUrl - The URL to the thumbnail size image of the item. ▪ zOrder - The item's Z-Order number. |
| name | The name of the catalog the item is in. |


Sample Request

method=site.getCatalogItem&itemId=31211594

Sample Response

```
{ "response": { "catalogRecord": [ { "catalogItems": { "catalogItem": [ { "id": 31211594, "description": "", "customData": "", "background": 0, "thumbUrl": "C:\\images\\bow.jpg", "designer": "", "zOrder": 0, "countRemaining": -1, "name": "Bow", "catalogName": "Sports", "fullUrl": "C:\\Documents\\Files\\bow.jpg", "price": 60, "categoryIds": "2896057", "category": "Cheer", "tags": "", "canUnselect": "null", "realItem": 0, "doCallback": 0, "passExtraData": 0, "ownedCount": 6, "orders": "0", "createTime": 1482436224, "pointCategoryId": 295169, "pointCategoryName": "Points", "lastUpdated": "1482436224", "catalogItemId": 31211594, "maxOwnedCountPerUser": 0 } ] }, "name": "Sports" } ] }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

site.getChallengeLeaders Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Generates a leaderboard for the users that have completed the most challenges.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|--|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is |

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| | | | | | the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| duration | No | String | DAY | 0 - 255 chars | Works in connection with the start parameter to determine duration. <ul style="list-style-type: none"> ▪ DAY - Shows challenges completed from 12:00 am EST on the start date. ▪ WEEK - Shows challenges completed from 12:00 am EST on the Sunday from the week of the start date. ▪ MONTH - Shows challenges completed from 12:00 am EST on the first day of the month of the start date. ▪ QUARTER - Shows challenges completed from 12:00 am EST of the first day of the month of the start date. The leaderboard runs for 90 days. ▪ YEAR - Shows challenges completed from 12:00 am EST on the first day of the start date year. ▪ ALLTIME - The start date is ignored. |
| groupName | No | String | None | 0 - 255 chars | Limited to 1 group. Only returns users in the specified group. Note: You cannot specify both userIds and a groupName unless withRank=true is also passed. In addition, if |

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------------|----------|---------------------|---------------|--|---|
| | | | | | withSurroundingUsers=true, you get a contextual leaderboard for just the specified group. |
| includeRepeatable | No | Boolean | true | true/false | Include repeatable challenges. |
| preferences | No | Pipe-separated list | None | 0 - 3 items | Returns up to 3 group preferences for each group in the leaderboard. |
| prefix | No | String | None | 0 - 255 chars | Only returns groups that start with the specific prefix. Underscore is used as the delimiter. |
| returnCount | No | Integer | 10 | 1 - 100 | Number of leaders to return. |
| start | No | Integer | None | 0 - 2 ³¹ -1 | UNIX timestamp. Sets the start date that works in connection with the duration parameter. All times are Eastern Standard Time. |
| trophiesOnly | No | Boolean | false | true/false | Includes only those challenges that have trophies associated with them. |
| userIds | No | Generic list | None | 0 - 100 items Truncated | A comma-separated list of users to generate the leaderboard for. All other users are ignored. When withRank is true, only one userId is allowed. Note: You cannot specify both userIds and a groupName unless withRank=true is also passed. In addition, if withSurroundingUsers=true, you get a contextual leaderboard for just the specified group. |
| withLeaderTagOnly | No | Boolean | true | true/false | Only includes challenges with the "challengeLeaders" tag. |

| Name | Required | Type* | Default Value | Range | Comments |
|-----------------------------|----------|---------|---------------|------------|---|
| withRank | No | Boolean | false | true/false | If true, returns the specified user's rank. Must pass a value in the userIds parameter. Limited to 1 userId. Works with any point category or duration specified. |
| withSurroundingUsers | No | Boolean | false | true/false | When used with userIds and withRank parameters, returns a contextual leaderboard for the specified user. |

[*Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------------|--|
| firstName | The user's first name. |
| lastName | The user's last name. |
| userId | The user's Nitro ID. |
| UserPreferences | Returns the value of the key specified in the preferences parameter. |
| value | The number of challenges completed. |


Sample Request

```
method=site.getChallengeLeaders&trophiesOnly=false&asyncToken=&preferences=&userIds=&returnCount=3&start=&duration=ALLTIME&withLeaderTagOnly=false&includeRepeatable=true
```

Sample Response

```
{ "response": { "challenges": { "challenge": [ { "userPreferences": {}, "userId": "1000", "firstName": "Vanna", "lastName": "Darville", "value": "27" }, { "userPreferences": {}, "userId": "1003", "firstName": "Captain", "lastName": "America", "value": "27" }, { "userPreferences": {}, "userId": "6000", "value": "27" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```


site.getGroupActionLeaders Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Gets a leaderboard with the groups that have the highest action values or most number of actions logged.

Note: Group leaderboards are calculated asynchronously and updated every 30 minutes. For groups, data does not display until after you log actions. For a site, the leaderboard takes the user's lifetime balance into account.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| criteria | No | String | COUNT | 0 - 255 chars | The type of leaders to return: <ul style="list-style-type: none"> ▪ SUM - Action value ▪ COUNT - Number of actions |
| duration | No | String | DAY | 0 - 255 chars | Works in connection with the start parameter to determine duration. <ul style="list-style-type: none"> ▪ DAY - Shows actions completed from 12:00 am EST on the start date. ▪ WEEK - Shows actions completed from 12:00 am EST on the Sunday from the week of the start date. ▪ MONTH - Shows actions completed from 12:00 am EST on the first day of the month of the start date. ▪ QUARTER - Shows actions completed from 12:00 am EST of the first day of the month of the start date. The leaderboard runs for 90 days. |

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|----------------------|---------------|---|---|
| | | | | | <ul style="list-style-type: none"> YEAR - Shows actions completed from 12:00 am EST on the first day of the start date year. ALLTIME - The start date is ignored. |
| preferences | No | Pipe-separated list | None | 0 - 3 items | Returns up to 3 group preferences for each group in the leaderboard. |
| prefix | No | String | None | 0 - 255 chars | Only returns groups that start with the specific prefix. Underscore is used as the delimiter. |
| returnCount | No | Integer | 10 | 1 - 100 | The number of values to return. |
| start | No | Integer | None | 0 - 2 ³¹ -1 | UNIX timestamp. Sets the start date that works in connection with the duration parameter. All times are Eastern Standard Time. |
| tags | YES | Comma-separated list | None | 1 - 5 items <u>Truncated</u> 0 - 255 chars per item | A comma-separated list of values. Only considers actions logged with these tags. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-------------------------|---|
| groupName | The group name. |
| GroupPreferences | The preferences set using group.setPreference . |
| value | The action value or number of actions depending on the selected criteria parameter. |


Sample Request

```
method=site.getGroupActionLeaders&asyncToken=&tags=comment_document&returnCount=3&start=&criteria=COUNT&duration=ALLTIME
```

Sample Response

```
{ "response": { "leaderboardId": 82140392, "groupLeaders": {
"groupLeader": [ { "groupPreferences": {}, "groupName": "NewYork",
"value": 14 }, { "groupPreferences": {}, "groupName":
"Country,Department:US,Sales", "value": 12 }, { "groupPreferences":
{}, "groupName": "Country:US", "value": 12 } ] } }, "responseType":
{}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

site.getGroupPointsLeaders Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Gets a leaderboard with the groups that have the highest sum of their users' balances.

Note: Group leaderboards are calculated asynchronously and updated every 30 minutes. For groups, data does not display until after you log actions. For a site, the leaderboard takes the user's lifetime balance into account.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|---------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| average | No | Boolean | false | true/false | Returns groups based on the average number of points they have per user. |
| criteria | No | String | BALANCE | 0 - 255 chars | The type of leaders to return: <ul style="list-style-type: none"> ▪ BALANCE - Users with the most points (credits minus debits). Uses current points balance. ▪ CREDITS - Users who have gained the most. Uses lifetime points balance. ▪ DEBITS - Users who have spent the most. |

| Name | Required | Type* | Default Value | Range | Comments |
|----------------------|----------|---------------------|---------------|------------------------|--|
| duration | No | String | DAY | 0 - 255 chars | Works in connection with the start parameter to determine duration. <ul style="list-style-type: none"> ▪ DAY - Shows points earned from 12:00 am EST on the start date. ▪ WEEK - Shows points earned from 12:00 am EST on the Sunday from the week of the start date. ▪ MONTH - Shows points earned from 12:00 am EST on the first day of the month of the start date. ▪ QUARTER - Shows points earned from 12:00 am EST of the first day of the month of the start date. The leaderboard runs for 90 days. ▪ YEAR - Shows points earned from 12:00 am EST on the first day of the start date year. ▪ ALLTIME - The start date is ignored. |
| pointCategory | No | String | None | 0 - 255 chars | The point category to get leaders for. Defaults to the default point category defined in Nitro Studio (Configuration > Site Settings > Point Categories > new/edit > Set as default category field). |
| preferences | No | Pipe-separated list | None | 0 - 3 items | Returns up to 3 group preferences for each group in the leaderboard. |
| prefix | No | String | None | 0 - 255 chars | Only returns groups that start with the specific prefix. Underscore is used as the delimiter. |
| returnCount | No | Integer | 10 | 1 - 100 | The number of values to return. |
| start | No | Integer | None | 0 - 2 ³¹ -1 | UNIX timestamp. Sets the start date that works in connection with the duration parameter. All times are Eastern Standard Time. |

| Name | Required | Type* | Default Value | Range | Comments |
|-------------|----------|----------------------|---------------|--|--|
| tags | YES | Comma-separated list | None | 0 - 5 items Truncated 0 - 255 chars per item | A comma-separated list of values. Only considers credits and debits with these tags. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-------------------------|---|
| groupName | The group name. |
| GroupPreferences | The preferences set using group.setPreference . |
| value | The group's current points balance, lifetime points balance, or debit balance depending on the selected criteria parameter. |

Sample Request

```
method=site.getGroupPointsLeaders&asyncToken=&tags=comment_document&pointCategory=&returnCount=2&start=&prefix=&criteria=BALANCE&duration=ALLTIME
```

Sample Response

```
{ "response": { "leaderboardId": 82140401, "groupLeaders": {
"groupLeader": [ { "groupPreferences": {}, "groupName": "NewYork",
"value": 904 }, { "groupPreferences": {}, "groupName":
"Country,Department:US,Sales", "value": 520 } ] } }, "responseType":
{}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

site.getGroups Method



As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Returns all groups for the site or only groups within a specified folder. For groups that a specific user is a member of, see [user.getGroups](#).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|---------|---------------|------------------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| folderName | No | String | None | 1 - 255 chars | The folder that contains the groups you want to return. |
| returnAll | No | String | None | 1 - 255 chars | Specifies whether to return everything as a flat result. When set to true, returns all groups regardless of the values in the returnCount and startIndex parameters. For example, if you have 200 groups and want to see all groups, set returnAll to true. If you only want to get groups 100-200, set returnCount to 100 and startIndex to 100. If you only want to get groups 150-200, set returnCount to 50 and startIndex to 150. |
| returnCount | No | Integer | 10 | 1 - 100 | The number of records to return. |
| sortBy | No | String | None | 1 - 255 chars | The order to sort the items in the response. Accepts date or name. |
| startIndex | No | Integer | 0 | 0 - 2 ³¹ -1 | A 0-based index specifying which row from the results to return data from. Can be used to page through the result set. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------------|---|
| count | The number of users in the group. |
| name | The group name. |
| preferences | Lists the group's preferences: <ul style="list-style-type: none"> ▪ name - The preference name. ▪ value - The preference value. |

| Response | Description |
|----------|--|
| | <ul style="list-style-type: none"> groupId - The group ID. |


Sample Request

```
method=site.getGroups&asyncToken=&pageSize=20&returnCount=3&returnAll=&folderName=&startIndex=0&sortBy=name
```

Sample Response

```
{ "response": { "groups": { "group": [ { "preferences": {
"groupPreference": [ { "name": "CaveCreek", "value": "Cave Creek",
"groupId": 2162945 }, { "name": "Phx", "value": "Phoenix", "groupId":
2162945 }, { "name": "NY", "value": "NewYork", "groupId": 2162945 } ]
}, "name": "Arizona", "count": 4 }, { "preferences": {
"groupPreference": [ { "name": "groupName", "value": "California",
"groupId": 2163073 } ] }, "name": "California", "count": 3 }, {
"preferences": {}, "name": "Colorado", "count": 0 } ] } },
"responseType": {}, "isError": false, "httpError": 0,
"httpsStatusCode": 200 }
```

site.getLevels Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Gets all the levels defined for the site. Each site level tag contains a type attribute which can be either custom or points. A level of type "custom" has an additional order attribute.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|-------------------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| type | No | String | Level type specified in | 0 - 255 chars | The type of level to get (Points or Custom). Defaults to the site level type selected in Nitro |

| Name | Required | Type* | Default Value | Range | Comments |
|------|----------|-------|---------------|-------|--|
| | | | Nitro Studio | | Studio (Configuration > Site Settings > Levels > Leveling Type). |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-----------------------|--|
| actionPhrase | The action phrase that displays in the newsfeed when a user reaches the level. |
| description | The level description. |
| iconUrl | The URL to the image associated with the level. |
| name | The level name. |
| notificationId | The ID of the notification associated with the level. |
| points | The number of points needed to reach the level. |
| type | The level type (points or custom). |


Sample Request

```
method=site.getLevels&asyncToken=&type=points
```

Sample Response

```
{ "response": { "siteLevels": { "siteLevel": [ { "points": 500,
"name": "Padawan", "description": "Learning the force", "iconUrl":
"https://s3.amazonaws.com/images/72878eb/userUploadpng1573dc1.png",
"type": "points", "notificationId": 709121, "actionPhrase": "Great
job achieving Padawan" }, { "points": 2000, "name": "Jedi Knight",
"description": "Ready to do battle", "iconUrl":
"https://s3.amazonaws.com/images/7284f78eb/userUploadpng157b50.png",
"type": "points", "notificationId": 709121, "actionPhrase": "A Jedi
is born" }, { "points": 10000, "name": "Jedi Master", "description":
"Master of the galaxy", "iconUrl":
"https://s3.amazonaws.com/images/7283ddb30ac4f78eb/userUploadpngacc74
.png", "type": "points", "notificationId": 709121, "actionPhrase":
"You have made it" } ] } }, "responseType": {}, "isError": false,
"httpError": 0, "httpsStatusCode": 200 }
```


site.getPointsLeaders Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Retrieves the point leaders in a specific set of tags. Extra arguments determine how many users are returned and the time frame to consider. Thereafter, any relevant write immediately updates the leaderboard with user data.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| criteria | No | String | BALANCE | 0 - 255 chars | The type of leaders to return: <ul style="list-style-type: none"> ▪ BALANCE - Users with the most points (credits minus debits). Uses current points balance. ▪ CREDITS - Users who have gained the most. Uses lifetime points balance. ▪ DEBITS - Users who have spent the most. |
| duration | No | String | DAY | 0 - 255 chars | Works in connection with the start parameter to determine duration. <ul style="list-style-type: none"> ▪ DAY - Shows points earned from 12:00 am EST on the start date. ▪ WEEK - Shows points earned from 12:00 am EST on the Sunday from the week of the start date. |

| Name | Required | Type* | Default Value | Range | Comments |
|----------------------|----------|---------------------|---------------|---------------|---|
| | | | | | <ul style="list-style-type: none"> ▪ MONTH - Shows points earned from 12:00 am EST on the first day of the month of the start date. ▪ QUARTER - Shows points earned from 12:00 am EST of the first day of the month of the start date. The leaderboard runs for 90 days. ▪ YEAR - Shows points earned from 12:00 am EST on the first day of the start date year. ▪ ALLTIME - The start date is ignored. |
| groupName | No | String | None | 0 - 255 chars | <p>Limited to 1 group. Only returns users in the specified group.</p> <p>Note: You cannot specify both userIds and a groupName unless withRank=true is also passed. In addition, if withSurroundingUsers=true, you get a contextual leaderboard for just the specified group.</p> |
| pointCategory | No | String | None | 0 - 255 chars | <p>The category to get leaders for (only 1 at a time). Defaults to the default point category defined in Nitro Studio (Configuration > Site Settings > Point Categories > new/edit > Set as default category field).</p> |
| preferences | No | Pipe-separated list | None | 0 - 3 items | <p>Allows you to specify a key for data to include in the response for each user returned. Items are case-sensitive. Reserved</p> |

| Name | Required | Type* | Default Value | Range | Comments |
|---------------------|----------|----------------------|---------------|--|---|
| | | | | | preferences include: firstName, lastName, and userPhotoUrl. |
| returnCount | No | Integer | 10 | 1 - 100 Truncated | The number of users to return. Returns the top 100 unless specified. |
| start | No | Integer | None | 0 - 2 ³¹ -1 | UNIX timestamp. Sets the start date that works in connection with the duration parameter. All times are Eastern Standard Time. |
| tags | No | Comma-separated list | None | 0 - 5 items Truncated 0 - 255 chars per item | A comma-separated list of tags. For the criteria you select (credits, debits, balance), only records with these tags are included. For tag based leaderboards, a maximum of 5 tags are allowed. |
| tagsOperator | No | Operator | OR | AND/OR | Actions that have any of the tags are considered. If set to AND, then only actions that have all of the tags are considered. |
| userIds | No | Generic list | None | 0 - 100 items Truncated | A comma-separated list of users to generate the leaderboard for. All other users are ignored. When withRank is true, only one userId is allowed. Note: You cannot specify both userIds and a groupName unless withRank=true is also passed. In addition, if withSurroundingUsers=true, you get a contextual leaderboard for just the specified group. |

| Name | Required | Type* | Default Value | Range | Comments |
|-----------------------------|----------|---------|---------------|------------|---|
| withRank | No | Boolean | false | true/false | If true, returns the specified user's rank. Must pass a value in the userIds parameter. Limited to 1 userId. Works with any point category or duration specified. |
| withSurroundingUsers | No | Boolean | false | true/false | When used with userIds and withRank parameters, returns a contextual leaderboard for the specified user. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------------|--|
| firstName | The user's first name. |
| lastName | The user's last name. |
| points | The user's current points balance, lifetime points balance, or debit balance depending on the selected criteria parameter. |
| user | The user's Nitro ID. |
| UserPreferences | Returns the value of the key specified in the preferences parameter. |

Sample Request

```
method=site.getPointsLeaders&asyncToken=&preferences=&pointCategory=P
oints&userIds=&tags=null&returnCount=&withSurroundingUsers=true&start
=&groupName=&tagsOperator=OR&criteria=BALANCE&duration=QUARTER&withRa
nk=false
```

Sample Response

```
{ "response": { "leaderboardId": 82144980, "leaders": { "leader": [ {
"userPreferences": {}, "userId": "6000", "points": 1364 }, {
"userPreferences": {}, "userId": "1000", "firstName": "Vanna",
"lastName": "Darville", "points": 900 }, { "userPreferences": {},
"userId": "9000", "firstName": "David", "lastName": "Brown",
"points": 250 }, { "userPreferences": {}, "userId": "1001",
"firstName": "Julie", "lastName": "Wildes", "points": 250 } ] } },
```

```
"responseType": {}, "isError": false, "httpError": 0,
"httpsStatusCode": 200 }
```

Sample Request - Invalid Criteria

```
method=site.getPointsLeaders&asyncToken=&preferences=&pointCategory=Points&userIds=&tags=null&returnCount=10&withSurroundingUsers=false&start=&groupName=NewYork&tagsOperator=OR&criteria=CredDeb&duration=DAY&withRank=false
```

Sample Response

```
{ "response": { "error": { "code": 126, "message": "Invalid Criteria" } }, "responseType": {}, "isError": true, "httpError": 0,
"httpsStatusCode": 400 }
```

site.getRecentActions Method



As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Allows you to see recent activity across your site by querying the action history. Only returns recent actions, not a full history, that were triggered on your site. For user level action history, see [user.getActionHistory](#).

Used mainly for spot validation or to display action activity to other users on your site.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|---------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| returnCount | No | Integer | 10 | 1 - 255 | The number of values to return. |

| Name | Required | Type* | Default Value | Range | Comments |
|---------------------|----------|----------------------|---------------|--|--|
| tags | No | Comma-separated list | None | 0 - 5 items Truncated 0 - 255 chars per item | A comma-separated list of tags (action tags or regular tags) to filter on. |
| tagsOperator | No | Operator | OR | AND/OR | If set to OR, actions that have any of the tags are considered. If set to AND, then only actions that have all of the tags are considered. |
| userIds | No | Generic list | None | 0 - 100 items Truncated | A comma-separated list of user IDs to exclusively consider. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------|---|
| firstName | First name of the user that completed the action. |
| lastName | Last name of the user that completed the action. |
| tags | The name of the completed action. |
| ts | The time the action was completed (UNIX timestamp). |
| userId | The user's Nitro ID. |
| value | The point value of the completed action. |


Sample Request

```
method=site.getRecentActions&asyncToken=&userIds=&tagsOperator=OR&returnCount=2&tags=null
```

Sample Response

```
{ "response": { "actions": { "action": [ { "userId": "4000", "firstName": "Kasey", "lastName": "Mang", "value": "0", "ts": 1520283222, "tags": "BLOG_CREATED," }, { "userId": "2000", "value": "0", "ts": 1520283201, "tags": "COMMENTED_DOCUMENT," } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

site.getRecentChallenges Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Gets the most recently completed challenges (only guaranteed to contain challenges achieved in the last three months). The list is ordered in reverse chronological order. Used mainly for troubleshooting.

Allows you to see all challenge completions across your site. For user level challenge completion, see [user.getChallengeProgress](#).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------------|----------|---------------------|---------------|--------------------------------------|--|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| challengeName | No | String | None | 1 - 255 chars | If specified, then the response is filtered to only include the specified challenge. Note: You cannot specify both userIds and a challengeName. |
| includeRepeatable | No | Boolean | true | true/false | Include repeatable challenges. |
| preferences | No | Pipe-separated list | None | 0 - 3 items | Allows you to specify a key for data to include in the response for each user returned. Items are case-sensitive. Reserved preferences include: firstName, lastName, and userPhotoUrl. |
| returnCount | No | Integer | 10 | 1 - 250 Truncated | The number of leaders to return. |

| Name | Required | Type* | Default Value | Range | Comments |
|---------------------|----------|--------------|---------------|--|--|
| trophiesOnly | No | Boolean | false | true/false | Includes only those challenges that have trophies associated with them. |
| userIds | No | Generic list | None | 0 - 100 items Truncated | A comma-separated list of user IDs. If specified, only these users are returned. Note: You cannot specify both <code>userIds</code> and a <code>challengeName</code> . |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------------|--|
| actionPhrase | The challenge's action phrase. |
| firstName | First name of the user that completed the challenge. |
| fullUrl | URL to the full size image associated with the challenge. |
| lastName | Last name of the user that completed the challenge. |
| name | The challenge name. |
| thumbUrl | URL to the thumbnail image associated with the challenge. |
| ts | The time the challenge was completed (UNIX timestamp). |
| userId | The user's Nitro ID. |
| UserPreferences | Returns the value of the key specified in the preferences parameter. |

Sample Request

```
method=site.getRecentChallenges&trophiesOnly=false&asyncToken=&preferences=&userIds=&returnCount=3&challengeName=&includeRepeatable=true
```


Sample Response

```
{ "response": { "challenges": { "challenge": [ { "userPreferences": {}, "userId": "2000", "name": "Save the Whales", "thumbUrl": "https://s3.amazonaws.com/images/72878eb/userUploadpng119c57.png", "fullUrl": "https://s3.amazonaws.com/images/7ac4f78eb/userUploadpng1f64c.png", "ts": 1520283201 }, { "userPreferences": {}, "userId": "3000", "name": "Log in to the mobile app", "thumbUrl":
```



```
"https://s3.amazonaws.com/images/7278eb/userUploadpng150f6.png",
"fullUrl":
"https://s3.amazonaws.com/images/c4f78eb/userUploadpng158b9.png",
"ts": 1519228120 }, { "userPreferences": {}, "userId": "2500",
"name": "March Madness", "actionPhrase": "{username} has completed
the March Madness challenge!", "thumbUrl":
"https://s3.amazonaws.com/images/7f78eb/userUploadpng15f49.png",
"fullUrl":
"https://s3.amazonaws.com/images/0ac4f78eb/userUploadpng1260.png",
"ts": 1519228120 } ] } }, "responseType": {}, "isError": false,
"httpError": 0, "httpsStatusCode": 200 }
```

site.getRecentUpdates Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Returns users who recently updated their avatar.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|--------------|---------------|--|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |
| criteria | YES | String | None | 0 - 255 chars | AVATAR |
| returnCount | No | Integer | 10 | 1 - 100 | The number of values to return. |
| userIds | No | Generic list | None | 0 - 100 items Truncated | A comma-separated list of user IDs to exclusively consider. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------------|---|
| ts | The date/time the item was updated. |
| userID | The Nitro ID of the user that updated their avatar. |
| Update type | The update type. Returns avatar. |


Sample Request

```
method=site.getRecentUpdates&asyncToken=&userIds=2001&returnCount=1&criteria=avatar
```

Sample Response

```
{ "response": { "updates": { "Update type": "avatar", "userId": "2001", "ts": "1473806705" } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

site.removeUsersFromGroup Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable.

Removes the specified users from a group. If the group doesn't exist, an error is returned.

By default, when removing a user from a group, the user's points remain with that group. To transfer a user's points when switching groups, enable the "Retain points when switching groups" option (Nitro Studio > **Configuration** > **Site Settings** > **General** > **Groups**).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |

| Name | Required | Type* | Default Value | Range | Comments |
|------------------|----------|--------------|---------------|--|--|
| groupName | YES | String | None | 1 - 255 chars | The name of the group. |
| userIds | YES | Generic list | None | 0 - 100 items Truncated | A comma-separated list of unique user IDs. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------|------------------------------------|
| groupName | The group users were removed from. |
| userName | The user's Nitro ID. |


Sample Request

```
method=site.removeUsersFromGroup&asyncToken=&userIds=5000,4000,&groupName=NewYork
```

Sample Response

```
{ "response": { "groupUsers": { "groupUsers": [ { "userName": "5000" }, { "userName": "4000" } ], "groupName": "NewYork" } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.backupSiteConfig Method

 Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Creates a backup of your Nitro site. Can be called from a repeatable/cron-like process to set up automated backups. This method is used in the Nitro Studio [Backup and Restore](#) process.

admin.backupSiteConfig supports OAuth security. To run this method you must:

1. [Create an OAuth Client](#) with an **fl backup** scope and record the Client Id and Secret Key values.
2. Use the Client Id and Secret Key to [generate an OAuth token](#).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|---------|---------------|------------------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| category | No | Integer | None | 0 - 255 chars | The action category. |
| description | No | String | None | 0 - 255 chars | The action description. |
| folderId | No | Long | None | 0 - 2 ³¹ -1 | The folder you want to contain the action. |
| iconUrl | No | String | None | 0 - 255 chars | The full path URL to the image used to represent this action in the actions list. |
| lowSecurity | No | Integer | None | 0 - 2 ³¹ -1 | Indicates if users can log this action regardless of whether your program has high, medium, or low security. If yes, you can use the user.clientLogAction method to log actions. <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| name | YES | String | None | 1 - 255 chars | The action name. |
| prefixMatch | No | Integer | None | 0 - 2 ³¹ -1 | Indicates if this action should log when a user completes an action with a name that exactly matches this action's name or has this action's name as the prefix. For example, if your action is named "UPLOAD" and you enable prefix matching, then completing an action named "UPLOAD_VIDEO" also causes the "UPLOAD" action to be logged. <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes <p>Note: Bunchball recommends using metadata instead of prefix matching.</p> |

| Name | Required | Type* | Default Value | Range | Comments |
|-----------|----------|---------|---------------|------------------------|--|
| rateLimit | No | Integer | None | 0 - 2 ³¹ -1 | If non-zero, multiple log actions performed for this action in the given time window will return an action rate exceeded warning |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-------------------|--|
| category | The category the action is in. |
| description | The action description. |
| id | The ID of the action. |
| isActionTag | Indicates if the tag is set on the action: <ul style="list-style-type: none"> 0 - No 1 - Yes |
| isEditable | Indicates if the action is editable: <ul style="list-style-type: none"> 0 - No 1 - Yes |
| lowSecurity | Indicates if the action has low security: <ul style="list-style-type: none"> 0 - No 1 - Yes |
| name | The action name. |
| prefixMatch | Indicates if prefix matching is enabled: <ul style="list-style-type: none"> 0 - No 1 - Yes |
| rateLimit | The maximum number of times this action can be logged in seconds, minutes, hours, or days. |
| serviceActionType | Previously used for social actions. |
| serviceType | Previously used for social actions. |
| status | Indicates if the action is active. |


Sample Request

```
method=admin.createAction&name=UPDATE_DOCUMENT&description=Update%20existing%20document
```

Sample Response

```
{ "response": { "tags": { "tag": [ { "id": 2734344780, "clientId":
"150401|-463|523603149867973", "name": "UPDATE_DOCUMENT",
"description": "Update existing document", "rateLimit": 0,
"isActionTag": 1, "lowSecurity": 0, "prefixMatch": 0, "category": 0,
"serviceType": 1, "serviceActionType": 1, "isEditable": 1, "order":
0, "status": "active" } ] } }, "responseType": {}, "isError": false,
"httpError": 0, "httpsStatusCode": 200 }
```

admin.createCatalog Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Creates a new avatar catalog.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|------------|----------|---------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| name | YES | String | None | 0 - 255 chars | The catalog name. |
| type | No | Integer | 1 | 1 | The catalog type: 1 - avatar catalog |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------|--------------------|
| catalogItems | The catalog items. |
| categories | The category name. |
| id | The catalog ID. |
| name | The catalog name. |

| Response | Description |
|----------|--|
| type | Returns the catalog type: 1 - avatar catalog |


Sample Request

```
method=admin.createCatalog&name=Sports&type=1
```

Sample Response

```
{ "response": { "catalog": { "categories": {}, "catalogItems": {},
"id": 433206, "clientId": "150401|111|1380526281039776", "name":
"Sports", "type": 1 } }, "responseType": {}, "isError": false,
"httpError": 0, "httpsStatusCode": 200 }
```

admin.createCatalogCategory Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Creates a new catalog category.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------|----------|---------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| canUnselect | No | Boolean | true | true/false | |
| catalogId | YES | Long | None | N/A | The Nitro-generated unique ID of the catalog to add the category to. Use admin.getAllCatalogContents to determine the catalogId. |
| name | YES | String | None | 1 - 255 chars | The category name. |

| Name | Required | Type* | Default Value | Range | Comments |
|---------------|----------|---------|---------------|------------------------|---|
| order | No | Integer | None | 0 - 2 ³¹ -1 | The category's order within the category hierarchy. |
| zOrder | No | Integer | None | 0 -32767 | The number that represents the Z-Order of the category. Categories with higher numbers are layered on top of categories with lower numbers. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-------------------------|--|
| canUnselect | |
| catalogId | The ID of the catalog containing the category. |
| catalogItems | |
| categories | |
| childCategoryIds | The IDs of subcategories within the category. |
| createTime | The date/time the category was created (UNIX timestamp). |
| id | The category ID. |
| name | The category name. |
| order | The category's order within the category hierarchy. |
| parents | |


Sample Request

```
method=admin.createCatalogCategory&name=Football&catalogId=433205&canUnselect=true&order=1
```

Sample Response

```
{ "response": { "category": { "categories": {}, "catalogItems": {}, "id": 2896061, "clientId": "150401|754|1380908821526849", "name": "Football", "catalogId": 433205, "order": 1, "canUnselect": "true", "childCategoryIds": "", "createTime": 1483633152, "parents": "" } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.createCatalogItem Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Creates a new catalog item.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|----------------------|---------------|--|--|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| active | No | Boolean | true | true/false | Indicates if the item is active within the catalog. |
| background | No | Boolean | false | true/false | Indicates if the item is a background item. |
| categoryIds | YES | Comma-separated list | None | 0 - 10 items 0 - 255 chars per item | A comma-separated list of categoryIds that the item belongs to. Use admin.getAllCatalogContents to determine the categoryIds. |
| cost | No | Integer | 0 | 0 - 2 ³¹ -1 | The cost to purchase the item (in points). |
| count | No | Integer | -1 | -1 - 2 ³¹ -1 | The number of items available for purchase. Can be unlimited or a fixed quantity. The count does not reflect items awarded manually from Nitro Studio. |
| customData | No | String | None | 0 - 10000 chars | The metadata to associate with the item. |

| Name | Required | Type* | Default Value | Range | Comments |
|-----------------------------|----------|----------------------|---------------|-------------------------|---|
| description | No | String | None | 0 - 255 chars | The item's description. |
| detailsUrl | No | String | None | 0 - 255 chars | The URL to the page that contains information about the item. |
| doCallback | No | Boolean | false | true/false | Indicates if Nitro should trigger a callback to the Purchase Notify URL (set in Configuration > Site Settings > General > Callbacks) when a user purchases the item. |
| fullUrl | No | String | None | 0 - 255 chars | The URL for the image of the item. |
| levelId | No | Long | None | N/A | If specified, only users who have reached the selected level can purchase the item. |
| maxOwnedCountPerUser | No | Integer | None | -1 - 2 ³¹ -1 | The maximum number of the item that any one user can own. Can be unlimited or a fixed quantity. |
| name | YES | String | None | 1 - 255 chars | The category item name. |
| orders | No | Comma-separated list | None | 0 - 255 items | The order the item appears in the catalog. |
| realItem | No | Boolean | false | true/false | Indicates if the item shows in the user's Owned Items list in widgets. |
| thumbUrl | No | String | None | 0 - 255 chars | The URL for the thumbnail of the item. |
| zOrder | No | Integer | None | 0 - 32767 | The number that represents the Z-Order of the item. Items with higher numbers get layered on top of items with lower numbers. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-----------------------------|--|
| active | Indicates if the item is active within the catalog: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| background | Indicates if the item is a background item: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| catalogItemId | The item's catalog ID. |
| catalogName | The name of the catalog the item is contained in. |
| category | The name of the category points are deducted from to purchase the item. |
| countRemaining | The number of items of this type remaining in the store. |
| createTime | The date/time the item was created in the catalog (UNIX timestamp). |
| customData | The custom data associated with the item. |
| description | The item's description. |
| doCallback | Indicates if a callback is triggered when a user purchases the item: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| endTime | The date/time the item is no longer available (UNIX timestamp). |
| maxOwnedCountPerUser | The maximum number of the item a user can purchase. |
| name | The item's name. |
| id | The item's ID number. |
| orders | The order the item appears in the catalog. |
| ownedCount | The number of items of this type that users own. |
| parentCategories | The ID numbers and names of this item's parent categories. |
| pointCategoryId | The ID of the point category that points are deducted from when the user purchases this item. |
| pointCategoryName | The name of the point category that points are deducted from when the user purchase this item. |
| price | The cost of the item (in points). |
| realItem | Indicates if the item shows in the user's Owned Items list in widgets: <ul style="list-style-type: none"> ▪ 0 - No |

| Response | Description |
|-----------|--|
| | <ul style="list-style-type: none"> 1 - Yes |
| startTime | The date/time the item becomes available (UNIX timestamp). |
| tags | The tags associated with the item. |
| zOrder | The item's Z-Order number. |


Sample Request

```
method=admin.createCatalogItem&categoryIds=2896057&cost=100&name=Shoes
```

Sample Response

```
{ "response": { "catalogItem": { "id": 31211596, "clientId": "150401|-579|1380877875326593", "description": "", "customData": "", "background": 0, "designer": "", "zOrder": 0, "countRemaining": -1, "name": "Shoes", "catalogName": "Sports", "price": 0, "categoryIds": "2896057", "category": "Cheer", "tags": "", "canUnselect": "null", "active": 1, "realItem": 0, "doCallback": 0, "passExtraData": 0, "ownedCount": 0, "orders": "0", "parentCategories": "2896057:Cheer", "createTime": 1483633408, "pointCategoryId": 295169, "pointCategoryName": "Points", "catalogItemId": 31211596, "startTime": 0, "endTime": 0, "maxOwnedCountPerUser": 0 } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.createChallenge Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Creates a new challenge. Challenge rules (and prerequisites) are created separately using the [admin.createRule](#) method.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|------------|----------|--------|---------------|---------------|--|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which |

| Name | Required | Type* | Default Value | Range | Comments |
|------------------------------|----------|---------|---------------|------------------------|---|
| | | | | | request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| actionPhrase | No | String | None | 0 - 255 chars | The phrase to use in the newsfeed for stories related to this challenge. |
| actionUrl | No | String | None | 0 - 255 chars | The URL that displays in the newsfeed when the challenge is completed. |
| activeFlag | No | Integer | None | 0 - 2 ³¹ -1 | Indicates whether the challenge is active and visible to the user: <ul style="list-style-type: none"> ▪ 0 - inactive (saves in draft mode) ▪ 1 - active |
| applyMultiplier | No | Integer | None | 0 - 2 ³¹ -1 | If set, multiply the pointAward value by the action value logged by the user (only applies when using a single rule with a goal of 'value'). |
| callbackFlag | No | Integer | 0 | 0 - 2 ³¹ -1 | Indicates if a callback should be sent every time the challenge is completed: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| catalogItemId | No | Long | None | N/A | If set, rewards the user with the specified catalog item for completing the challenge. |
| customData | No | String | None | 0 - 10000 chars | The metadata to associate with the challenge. |
| dailyAchievementLimit | No | Integer | None | 0 - 2 ³¹ -1 | If repeatable, the maximum number of times the challenge can be completed in one day. |
| description | No | String | None | 0 - 2048 chars | The challenge description. |
| emailNotificationId | No | Long | None | N/A | For internal use only. |

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------------|----------|---------|---------------|--|---|
| end | No | Integer | 0 | 0 - 2 ³¹ -1 | UNIX timestamp. Sets the date the challenge ends. All times are Eastern Standard Time. |
| folderId | No | Long | None | N/A | The ID of the folder the challenge is located in. Use admin.getFolders to get a list of folders and their folder IDs. |
| fullUrl | No | String | None | 0 - 255 chars | The URL for the trophy or badge to award to the user upon challenge completion. |
| groupChallenge | No | Boolean | None | true/false | Not recommended for use (available for backward compatibility). Use "type" parameter instead. |
| hideBeforeEarned | No | Integer | None | 0 - 2 ³¹ -1 | Indicates if the challenge should be hidden until the user completes it: <ul style="list-style-type: none"> ▪ 0 - Not hidden ▪ 1 - Hidden |
| iconUrl | No | String | None | 0 - 255 chars | The URL for the icon of the trophy or badge associated with the challenge. |
| levelId | No | Long | None | N/A | For internal use only. |
| limit | No | Integer | None | -1 - 2 ³¹ -1 | The maximum number of points to be awarded, or the number of times the challenge can be completed. |
| limitType | No | String | None | 0 - 255 chars | The limit type. Choices are 'points' and 'challenges' depending on which type of limit was specified. |
| name | YES | String | None | 0 - 255 chars | The challenge name. Must be unique. |
| newApi | No | String | None | 0 - 255 chars Truncated | Indicates if the request is using the rule API. |
| notificationId | No | Long | None | N/A | The ID of the notification a user should see after completing the challenge. |
| order | No | Integer | None | 0 - 2 ³¹ -1 | Determines the order the challenge should display in the trophy case. |

| Name | Required | Type* | Default Value | Range | Comments |
|-----------------------------|----------|----------------------|---------------|--|--|
| pointAward | No | Integer | None | -1 - 2 ³¹ -1 | The number of points to credit (or debit) a user after completing the challenge. |
| pointCategoryId | No | Long | None | N/A | The ID of the point category that points are awarded from. |
| preferences | No | Comma-separated list | None | 0 - 3 items Truncated | A comma-separated list of preferences required to participate in the challenge. Required when using preferenceValues. |
| preferencesMatchType | No | String | or | 0 - Unlimited chars | Indicates how preferences associated with the challenge must be matched to complete it. Options are: <ul style="list-style-type: none"> ▪ and - All ▪ or - Any |
| preferenceValues | No | Comma-separated list | None | 0 - 3 items Truncated | A comma-separated list of the preferences' values. The number of values should match the number of preferences, otherwise values are truncated. Required when using preferences. |
| prereqCatalogItemId | No | Long | None | N/A Truncated | The ID of the catalog item required to complete the challenge. |
| prereqMatchType | No | String | or | 0 - Unlimited chars | Indicates how prerequisites associated with the challenge must be matched to complete it. Accepts and, all, or any. |
| repeatable | No | Integer | None | 0 - 2 ³¹ -1 | Indicates if the challenge can be completed multiple times: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| rewards | No | Comma-separated list | None | 0 -10 items Truncated | A list of points, notifications, and catalog items that are awarded to a user when they complete the challenge. |

| Name | Required | Type* | Default Value | Range | Comments |
|----------------------|----------|----------------------|---------------|------------------------|---|
| ruleMatchType | No | Integer | None | 0 - 2 ³¹ -1 | Indicates how rules associated with the challenge must be matched to complete it. Options are: <ul style="list-style-type: none"> ▪ 0 - All ▪ 1 - Any ▪ 2 - Order listed |
| start | No | Integer | 0 | 0 - 2 ³¹ -1 | UNIX timestamp. Sets the start date after which the challenge can be completed. All times are Eastern Standard Time. |
| tags | No | Comma-separated list | None | 0 - 10 items | Used to pass the action name. A comma-separated list of actions and contextual tags to log. Mainly used to power leaderboards. The parameter is case-sensitive and includes both the action and contextual tag. For example, <code>method=logAction&tags=action_tag,contextual1,contextual2,contextual3</code> . The parameter supports up to a total of 10 items (10 contextual tags, or 10 action tags, or 1 action tag and 9 contextual tags). However, leaderboards and reporting can only support up to 5 contextual tags. Note: Bunchball discourages logging actions with multiple comma-delimited tags and encourages using metadata and preferences. |
| thumbUrl | No | String | None | 0 - 255 chars | The URL for the thumbnail of the trophy or badge to award to the user upon challenge completion. |
| type | No | String | user | 0 - Unlimited chars | The type of challenge to be created. Options are: <ul style="list-style-type: none"> ▪ user - User challenges |

| Name | Required | Type* | Default Value | Range | Comments |
|------|----------|-------|---------------|-------|--|
| | | | | | <ul style="list-style-type: none"> ▪ groupAggregateUser - Group challenges that aggregate user actions ▪ group - Group challenges based on group actions |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------------------|---|
| activeFlag | Indicates if the challenge is active: <ul style="list-style-type: none"> ▪ 0 - Not active ▪ 1 - Active |
| applyMultiplier | The points multiplier in use for the challenge. |
| callbackFlag | Indicates if the challenge is enabled to trigger an achievement callback when completed: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| catalogItemId | The ID of the catalog item awarded to a user when they complete the challenge. |
| ChallengeRewards | The rewards associated with the challenge: <ul style="list-style-type: none"> ▪ type - Indicates the reward type. Returns point, item, or notification. ▪ details - Additional reward information. For points, the details may be something like "285424 50" with the number of points awarded after the pipe (). In this example, 50 points are awarded to the user after completing the challenge. ▪ itemName - The name of the point category, catalog item, or notification. |
| dailyAchievementLimit | The number of times the challenge can be completed in a day. Returns 0 for unlimited. |
| dateIssued | The date the challenge was activated (UNIX timestamp). |
| description | The challenge's description. |
| endTime | The date/time the challenge ends (UNIX timestamp). |
| featuredFlag | Indicates if the challenge is featured within the Missions widget: |

| Response | Description |
|-----------------------------|--|
| | <ul style="list-style-type: none"> ▪ 0 - Not featured ▪ 1 - Featured |
| groupFlag | Indicates if the challenge is a group challenge: <ul style="list-style-type: none"> ▪ 0 - Not a group challenge ▪ 1 - A group challenge |
| groupPointAward | The number of points awarded to the group for this challenge. |
| hideUntilEarned | Indicates if the challenge is hidden until earned: <ul style="list-style-type: none"> ▪ 0 - Not hidden ▪ 1 - Hidden |
| id | The challenge ID. |
| name | The challenge name. |
| order | Indicates if the challenge rules must be completed in order: <ul style="list-style-type: none"> ▪ 0 - Do not have to be completed in order ▪ 1 - Must be completed in order |
| pointAward | The number of points awarded for completing the challenge. |
| pointCategory | For challenges with point rewards, returns the name of the point category the points are awarded to. |
| preferences | Used for metric goals where a segment determines which value to use in the goal. |
| preferencesMatchType | Indicates if the prerequisite is 'and' or 'or'. |
| preferenceValues | The values for the preferences list. |
| prereqMatchType | Indicates if the prerequisite is 'and' or 'or'. |
| repeatable | Indicates if the challenge is repeatable: <ul style="list-style-type: none"> ▪ 0 - Not repeatable ▪ 1 - Repeatable Repeatable, hidden challenges show in both complete and incomplete once initially completed. |
| ruleMatchType | Indicates if the rule operator is 'and' or 'or'. |
| serviceType | Previously used for social actions. |
| startTime | The date/time the challenge starts (UNIX timestamp). |
| tags | The tags associated with the challenge. |
| version | For internal use and backward compatibility. |


Sample Request

```
method=admin.createChallenge&catalogItemId=31211594&dailyAchievementLimit=1&description=The Road to Nationals&name=CANationals&repeatable=1
```

Sample Response

```
{ "response": { "challenges": { "challenge": [ { "rules": {}, "challengeRewards": { "challengeReward": [ { "type": "item", "details": "31211594", "itemName": "Bow" } ] }, "id": 4806389, "name": "CANationals", "description": "The Road to Nationals", "activeFlag": 1, "featuredFlag": 0, "callbackFlag": 0, "serviceType": "1", "order": 0, "pointCategory": "Points", "pointAward": 0, "dateIssued": "1483650720", "applyMultiplier": 0, "repeatable": 1, "hideUntilEarned": 0, "catalogItemId": 31211594, "tags": "", "startTime": 1483650720, "endTime": 0, "ruleMatchType": 0, "prereqMatchType": "or", "preferences": "", "preferenceValues": "", "preferencesMatchType": "or", "groupFlag": 0, "groupPointAward": 0, "dailyAchievementLimit": 1, "version": "5.2" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.createFolder Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Creates a new folder in Nitro Studio and returns the folderId of the newly created folder.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| name | YES | String | None | 1 - 255 chars | The folder name. |

| Name | Required | Type* | Default Value | Range | Comments |
|----------|----------|--------|---------------|------------------------|--|
| parentId | No | Long | None | 0 - 2 ³¹ -1 | The ID of the parent folder to create the folder within. Use admin.getFolders to get a list of folders and their folder IDs. |
| type | YES | String | None | 1 - 255 chars | The folder type. Accepts "challenge" or "action". |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------|---|
| created | The date/time the folder was created. |
| id | The folder ID. |
| name | The challenge name. |
| type | The folder type. Returns action or challenge. |


Sample Request

```
method=admin.createFolder&name=ChallengeFolder&type=challenge
```

Sample Response

```
{ "response": { "folders": { "folder": [ { "id": 574477, "name": "ChallengeFolder", "type": "challenge", "order": 16, "created": 1487366936, "clientId": "150401|-813|173076943771477" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.createMetadatum Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Adds metadata to an existing rule. You can add metadata to further qualify the action logged by the user. When you add [metadata](#) name:value pairs to a rule, the metadata logged in the [/users/{userId}/actions](#) API must match the rule metadata exactly for the rule to be completed by a user. Keep in mind that metadata is case sensitive, and as such, the best practice is to use

simple strings without special characters and/or numeric IDs as metadata names and values. This minimizes the risk of typos and incorrect formatting from completing rules.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|--|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| name | YES | String | None | 0 - 255 chars | The name half of the metadata name:value pair required for the rule. |
| ruleId | YES | Long | None | N/A | The ID of the rule the metadata is associated with. Use admin.getChallenges with <code>includeRules= true</code> to return rule IDs. |
| value | YES | String | None | 0 - 255 chars | The value half of the metadata name:value pair required for the rule. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------------|--|
| activeFlag | Indicates if the challenge is active: <ul style="list-style-type: none"> ▪ 0 - Not active ▪ 1 - Active |
| applyMultiplier | The points multiplier in use for the challenge. |
| callbackFlag | Indicates if the challenge is enabled to trigger an achievement callback when completed: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| catalogItemId | The ID of the catalog item awarded for completing the challenge. |
| ChallengeReward | The rewards associated with the challenge: <ul style="list-style-type: none"> ▪ type - Indicates the reward type. Returns point, item, or notification. ▪ details - Additional reward information. For points, the details may be something like "285424 50" with the number of points awarded after |

| Response | Description |
|------------------------------|---|
| | <p>the pipe (). In this example, 50 points are awarded to the user after completing the challenge.</p> <ul style="list-style-type: none"> ▪ itemName - The name of the point category, catalog item, or notification. |
| custom | The challenge's custom data. |
| dailyAchievementLimit | The number of times the challenge can be completed in a day. Returns 0 for unlimited. |
| dateIssued | The date the challenge becomes active (UNIX timestamp). |
| description | The challenge's description. |
| endTime | The date the challenge ends (UNIX timestamp). Returns 0 if no end date is set. Typically used to display the expiration date of the challenge to the user. |
| groupFlag | <p>Indicates if it's a group challenge:</p> <ul style="list-style-type: none"> ▪ 0 - Not a group challenge ▪ 1 - A group challenge |
| groupPointAward | The number of points awarded to the group for completing the challenge. |
| hideUntilEarned | <p>Indicates if the challenge is hidden until complete:</p> <ul style="list-style-type: none"> ▪ 0 - Not hidden ▪ 1 - Hidden <p>Repeatable, hidden challenges show in both complete and incomplete once initially completed.</p> |
| id | The challenge ID. |
| metadata | <p>The metadata associated with the rule:</p> <ul style="list-style-type: none"> ▪ id - The ID of the rule. ▪ name - The name half of the metadata name:value pair. ▪ value - The value half of the metadata name:value pair. |
| name | The challenge name. |
| order | <p>Indicates if the challenge rules must be completed in order:</p> <ul style="list-style-type: none"> ▪ 0 - Do not have to be completed in order. ▪ 1 - Must be completed in order. |
| pointAward | The number of points awarded for completing the challenge. |
| pointCategory | For challenges with point rewards, returns the name of the point category the points are awarded to. |
| pointCategoryId | The ID of the point category. |

| Response | Description |
|-----------------------------|--|
| preferences | Used for metric goals where a segment determines which value to use in the goal. |
| preferencesMatchType | Indicates if the preference operator is "and" or "or". |
| preferenceValues | The values for the preferences list. |
| prereqMatchType | Indicates if the prerequisite operator is "and" or "or". |
| repeatable | <p>Indicates if the challenge is repeatable:</p> <ul style="list-style-type: none"> ▪ 0 - Not repeatable ▪ 1 - Repeatable <p>Repeatable, hidden challenges show in both complete and incomplete once initially completed.</p> |
| rules | <p>The rules associated with the challenge:</p> <ul style="list-style-type: none"> ▪ actionTag - The name of the action associated with the rule. ▪ displayOrder - The rule's order within the column. ▪ description - The rule's action description. ▪ goal - The action value goal required by the rule. For Less than scenarios, the action value lists this value as the number configured in Nitro Studio. Must be used with operator for Less than scenarios. ▪ id - The rule ID. ▪ itemName - The action name. ▪ operator - The action goal value: <ul style="list-style-type: none"> ▪ GE - Greater than or equal to ▪ GT - Greater than ▪ LT - Less than ▪ LE - Less than or equal to ▪ EQ - Equal to ▪ prereqOperator - Indicates if one or all prerequisites must be met. Returns "and" or "or". ▪ prereqRuleIds - The ID of the rule that must be completed before this rule (for multi-rule challenges). ▪ serviceActionType - Previously used for social actions. ▪ serviceType - Previously used for social actions. ▪ sortOrder - An integer value indicating the rules' order within the challenge. ▪ type - Indicates the action goal type: |

| Response | Description |
|----------------------|--|
| | <ul style="list-style-type: none"> ▪ sum - Running total ▪ count - Number of calls |
| ruleMatchType | Indicates if the rule operator is "and" or "or". |
| serviceType | Previously used for social actions. |
| startTime | The challenge start date (UNIX timestamp). |
| tags | The challenge's tags. |
| version | For internal use and backward compatibility. |


Sample Request

method=admin.createMetadatum&ruleId=6984368&name=content&value=video

Sample Response

```
{ "response": { "challenges": { "challenge": [ { "rules": { "rule": [
{ "metadata": { "metadatum": [ { "id": 518772, "clientId":
"150401|342|510916945787980", "name": "content", "value": "video" } ]
}, "id": 6999480, "clientId": "150401|-126|1458682661823470",
"tagId": 2734215237, "actionTag": "learn_more", "operator": "GE",
"description": "Research motorcoach", "goal": 2, "timeRange": 3600,
"type": "count", "sortOrder": 0, "displayOrder": 0, "serviceType":
"nitro", "serviceActionType": "none", "prereqRuleIds": "",
"prereqOperator": "or", "custom": "2734215237", "itemName":
"learn_more" }, { "metadata": {}, "id": 6999481, "clientId":
"150401|-20|1458412117198676", "operator": "", "goal": 0, "type":
"operator", "sortOrder": 0, "displayOrder": 0, "prereqRuleIds":
"6999480", "prereqOperator": "or" }, { "metadata": {}, "id": 7000681,
"clientId": "150401|-265|73679544724968", "operator": "", "goal": 0,
"type": "operator", "sortOrder": 0, "displayOrder": 0,
"prereqRuleIds": "6999481", "prereqOperator": "and" } ] } },
"challengeRewards": { "challengeReward": [ { "type": "item",
"details": "31211594", "itemName": "Bow" } ] }, "id": 4806388,
"name": "CANationals", "description": "The Road to Nationals",
"activeFlag": 1, "featuredFlag": 0, "callbackFlag": 0, "serviceType":
"1", "order": 0, "pointCategory": "Points", "pointAward": 0,
"dateIssued": "1483650660", "applyMultiplier": 0, "repeatable": 1,
"hideUntilEarned": 1, "catalogItemId": 31211594, "tags": "expenses",
"startTime": 1483650660, "endTime": 1488265200, "ruleMatchType": 0,
"prereqMatchType": "or", "preferences": "", "preferenceValues": "",
"preferencesMatchType": "or", "groupFlag": 0, "groupPointAward": 0,
"dailyAchievementLimit": 1, "version": "5.2" } ] } }, "responseType":
{}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.createRule Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Creates and adds a rule for the specified challenge. Rules are “completed” by users when the correct action, value, and metadata is logged via [/users/{userId}/actions](#).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|---------------------|----------|---------|---------------|------------------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| challengeId | No | Long | None | N/A | The ID of the challenge the rule should be associated with. The ID is returned by admin.getChallenges . |
| countGoals | No | Integer | None | 0 - 2 ³¹ -1 | A comma separated list of the number of times the action value rule is satisfied. |
| description | No | String | None | 0 - 255 chars | The user-facing description of this rule. |
| displayOrder | No | Long | 0 | N/A | The order the rule is shown in the challenge. |
| folderId | No | Long | None | N/A | The ID of the folder the rule should be associated with. Applies to withinFolder rules. |
| goal | No | Integer | None | 0 - 2 ³¹ -1 | The numerical value for how much or how often the action must be logged. |
| isNegated | No | Boolean | false | true/false | Set to false to preserve backward compatibility. Only group prerequisite |

| Name | Required | Type* | Default Value | Range | Comments |
|-----------------------|----------|----------------------|---------------|--|---|
| | | | | | rules support a true value to allow a group prerequisite exclusion. |
| names | No | String | None | 0 - 255 chars | The user preference names. |
| newApi | No | String | None | 0 - 255 chars Truncated | Indicates if the request is using the Rule API. Always set to true. |
| operator | No | String | None | 0 - 255 chars | The comparison operator to use to determine if the rule goal has been met: <ul style="list-style-type: none"> ▪ GT - greater than ▪ GE - greater than or equal to ▪ EQ - equal to ▪ LT - less than ▪ LE - less than or equal to |
| prereqOperator | No | String | or | 0 - 5 chars | Must be either 'and' or 'or' |
| prereqRuleIds | No | Comma-separated list | None | | The prerequisite rules' IDs. |
| segment | No | String | None | 0 - 255 chars | The segment name. |
| tagId | No | Long | None | N/A | The ID of the tag the rule should be associated with. |
| timeRange | No | Integer | None | 0 - 2 ³¹ -1 | The time range of the rule in seconds. Applies to timeRange type rules. |
| type | YES | String | None | 0 - 255 chars | The action goal for this rule (what to measure): <ul style="list-style-type: none"> ▪ challengePrereq - Use to define which challenge(s) the user must complete before they are eligible for this challenge. |

| Name | Required | Type* | Default Value | Range | Comments |
|------|----------|-------|---------------|-------|---|
| | | | | | <ul style="list-style-type: none"> ▪ count - The number of times a user must perform the action. ▪ groupClassPrereq - The group class the user must be in to be eligible for the challenge. ▪ metricGoal - Use for challenges that apply to different segments of your program where each segment has different action criteria. ▪ operator - Use to create multiple rules for challenges. To use this type, you must specify a <code>prereqOperator</code> for the and/or operator, <code>prereqRuleIds</code> to indicate which rules this operator is for, and the <code>challengeId</code> that the rules are part of. ▪ preferencePrereq - Use to create a prerequisite for the challenge. The preference is defined in the custom field with a pipe separator. For example, <code>custom=test1_18/12\$@_@\$level1 true</code>. This also requires a "root" node which is an operator node. The prerequisite rule and the action rule should have the root node as a prerequisite. ▪ sum - The total value a user must accumulate. ▪ timeRange - The number of times a user must perform the action within the specified time frame. ▪ value - The value for the action that a user must achieve. ▪ withinFolder - Users complete a number of actions within a folder. |

| Name | Required | Type* | Default Value | Range | Comments |
|----------------------------------|----------|----------------------|---------------|---------------|--|
| url | No | String | None | 0 - 255 chars | The URL where users begin to complete this rule. |
| values | No | Comma-separated list | None | | The user preference values. |
| valueLowerBounds | No | Comma-separated list | None | | Used with valueUpperBounds to support Between logic. Sets the greater than value. MIN can be used when there is an upper bound defined, but no lower bound. |
| valueUpperBounds | No | Comma-separated list | None | | Used with valueLowerBounds to support Between logic. Sets the less than value. MAX can be used when there is a lower bound defined, but no upper bound. |
| valueLowerBoundInclusives | No | Boolean | false | true/false | Indicates whether the rule should be TRUE or FALSE when it equals the value. When used with lower bound and upper bound values, logic results in Greater than or equals. |
| valueUpperBoundInclusives | No | Boolean | false | true/false | Indicates whether the rule should be TRUE or FALSE when it equals the value. When used with lower bound and upper bound values, logic results in Less than or equals. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|---------------------|---|
| actionTag | The tags associated with the rule. |
| clientId | |
| displayOrder | The order the rule is shown in the challenge. |
| filter | The rule's filter type. |
| goal | The action value. |

| Response | Description |
|-------------------------------|--|
| id | The rule's ID. |
| itemName | The action, challenge, or group class name. |
| Metadata | Indicates if the rule contains metadata. Returns true or false. |
| MetricRuleFilterDetail | When the rule is of type metricGoal: <ul style="list-style-type: none"> ▪ countGoal - The number of times the action was completed. ▪ lowerBoundInclusive - If the rule is equal to the lower bound value. Returns true or false. ▪ name - The user preference name. ▪ upperBoundInclusive - If the rule is equal to the upper bound value. Returns true or false. ▪ value - The user preference value. ▪ valueLowerBound - The rule's greater than value. ▪ valueUpperBound - The rule's less than value. |
| operator | The rule's operator type. Returns GT (greater than), GE (greater than or equal to), EQ (equal to), LT (less than), or LE (less than or equal to). |
| prereqOperator | Indicates if one or all prerequisites must be met. Returns 'and' or 'or'. |
| prereqRuleIds | The ID of the rule that must be completed before this rule (for multi-rule challenges). |
| serviceActionType | Previously used for social actions. |
| serviceType | Previously used for social actions. |
| sortOrder | An integer value indicating the rule's order within the challenge. |
| tagId | The ID of the tag the rule is associated with. |
| type | The rule type. Returns sum, count, value, none, metricGoal, timeRange, or withinFolder. |

Sample Request

```
method=admin.createRule&type=metricGoal&tagId=2734215856&prereqOperator=or&operator=GE&displayOrder=0&challengeId=4802299&countGoals=1,1&valueLowerBounds=50,50&valueUpperBounds=100,100&valueLowerBoundInclusives=true,true&valueUpperBoundInclusives=true,true&names=league%7Cleague&values=west%7Ceast&filter=segment&newApi=true
```

Sample Response

```
{ "response": { "rule": { "metadata": {}, "metricRuleFilter": {
"metricRuleFilterDetails": { "metricRuleFilterDetail": [ { "name":
"league", "value": "west", "countGoal": 1, "valueLowerBound": "50.0",
"valueUpperBound": "100.0", "lowerBoundInclusive": true,
"upperBoundInclusive": true }, { "name": "league", "value": "east",
"countGoal": 1, "valueLowerBound": "50.0", "valueUpperBound":
"100.0", "lowerBoundInclusive": true, "upperBoundInclusive": true } ]
}, "filter": "segment" }, "id": 6999469, "clientId": "150401|-
661|1403066704424633", "tagId": 2734215856, "actionTag":
"DeployRelease", "operator": "GE", "goal": 1, "type": "metricGoal",
"sortOrder": 0, "displayOrder": 0, "serviceType": "nitro",
"serviceActionType": "none", "prereqRuleIds": "", "prereqOperator":
"or", "itemName": "DeployRelease" } }, "responseType": {}, "isError":
false, "httpError": 0, "httpsStatusCode": 200 }
```


Sample Request - Prerequisites

```
method=admin.createRule&type=preferencePrereq&challengeId=6&custom=le
ague%7Cnational&newApi=true&sessionKey=NXw4fDZny
```

Then, add the returned ruleID (10 in this example) into the action rule creation as a prereqRuleId.

```
method=admin.createRule&itemName=winlastgame&itemType=action&type=cou
nt&tagId=14&prereqOperator=or&operator=GE&goal=1&description=&display
Order=0&prereqRuleIds=10&challengeId=6&newApi=true&sessionKey=NXw4fDZ
ny
```

admin.createSitePointCategory Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Creates a point category. Only returns the new point category, not all point categories.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|------------|----------|--------|---------------|---------------|--|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used |

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------------------------|----------|---------|---------------|------------------------|---|
| | | | | | to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| dailyMaxCredits | No | Integer | 0 | 0 - 2 ³¹ -1 | The maximum number of points a user can earn in this point category in a single day. |
| dailyMaxCreditsNotificationID | No | String | None | 1 - 255 chars | The ID of the notification that displays when the user reaches the daily max points limit. |
| expirationFrequency | No | String | None | 1 - 255 chars | How often the user's balance in this point category is reset to 0. Accepts never, monthly, or yearly. |
| iconUrl | No | String | None | 1 - 255 chars | The URL to the icon that displays for the point category. |
| initialPoints | No | Integer | 0 | 0 - 2 ³¹ -1 | The number of points each new user starts with. |
| isDefault | No | Boolean | false | true/false | Indicates if this is the default point category. Accepts true or false. |
| maxBalance | No | Integer | 0 | 0 - 2 ³¹ -1 | The lifetime maximum balance a user can have in this point category. |

| Name | Required | Type* | Default Value | Range | Comments |
|---------------------------------------|----------|---------|---------------|------------------------|--|
| maxBalanceNotificationId | No | String | None | 1 - 255 chars | The URL to the notification that displays when the user reaches the maximum points balance for the first time. |
| name | No | String | None | 1 - 255 chars | The point category name. |
| pointCategoryId | No | Integer | 0 | 0 - 2 ³¹ -1 | The point category ID. |
| shortName | No | String | None | 1 - 255 chars | The abbreviated category name. |
| yearlyMaxCredits | No | Integer | 0 | 0 - 2 ³¹ -1 | The maximum number of points a user can earn in this point category in a single year. |
| yearlyMaxCreditsNotificationID | No | String | None | 1 - 255 chars | The URL to the notification that displays when the user reaches the yearly max points limit. |

[*Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------------------------------|---|
| dailyMaxCredits | The maximum number of points a user can earn in this point category in a single day. |
| dailyMaxCreditsNotificationId | The ID of the notification that displays when the user reaches the daily max points limit. |
| expirationFrequency | How often the user's balance in this point category is reset to 0. Returns never, monthly, or yearly. |
| iconUrl | The URL to the icon that displays for the point category in widgets where you show the user's level. |
| id | The point category's ID. |
| initialPoints | The number of points each new user starts with. |

| Response | Description |
|---------------------------------------|---|
| isDefault | Indicates if this is the default point category. Returns true or false. |
| maxBalance | The lifetime maximum balance a user can have in this point category. |
| maxBalanceNotificationId | The URL to the notification that displays when the user reaches the maximum points balance for the first time. |
| name | The point category name. |
| premium | Indicates if this is a premium points category: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| shortName | The abbreviated category name. |
| yearlyMaxCredits | The maximum number of points a user can earn in this point category in a single year. |
| yearlyMaxCreditsNotificationId | The URL to the notification that displays when the user reaches the yearly max points limit. |


Sample Request

method=admin.createSitePointCategory&isDefault=false&name=ServicePoints

Sample Response

```
{ "response": { "pointCategory": { "id": 591, "clientId": "1|530|14378783102979", "name": "ServicePoints", "premium": 0, "iconUrl": "", "shortName": "", "maxBalance": 0, "initialPoints": 0, "dailyMaxCredits": 0, "expirationFrequency": "never", "isDefault": false, "yearlyMaxCredits": 0 } }, "responseType": {}, "isError": false, "httpError": 0, "httpStatusCode": 200 }
```

admin.deleteMetadatum Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Deletes the metadata of an existing rule.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|--------|---------------|------------------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| metadatumId | YES | Long | None | 0 - 2 ³¹ -1 | The unique ID for the rule's metadata. Use admin.getChallenges with includeRules= true to return metadata IDs. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-------------|--|
| id | The metadata ID. |
| name | The name half of the metadata name:value pair. |


Sample Request

```
method=admin.deleteMetadatum&metadatumId=524543
```

Sample Response

```
{ "response": { "deleteSuccess": { "object": "Metadatum", "name": "quiz_id", "id": 524543 } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.deleteUser Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Deletes a user.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| userId | YES | String | None | 0 - 255 chars | The ID of the user to be deleted. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|---------------|---|
| object | The type of Nitro object to be deleted. Returns "User". |
| name | The user ID. |
| id | The user's internal Nitro ID. |


Sample Request

```
method=admin.deleteUser&userId=6001
```

Sample Response

```
{ "response": { "deleteSuccess": { "object": "User", "name": "6001", "id": 3857887042 } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.exportLocaleTranslations Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Exports an XML translation file for a given locale.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|--|
| locale | YES | String | None | 0 - 255 chars | Locale that has been activated in Nitro Studio (Configuration > Localization). |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| locale | YES | String | None | 1 - 255 chars | The locale to export. |
| version | No | String | 2 | 1 - 2 chars | The version of the translation file. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------------|--|
| apiKey | The site's API key. |
| element | <p>The list of elements to be translated such as challenge names, level names, and catalog items.</p> <ul style="list-style-type: none"> ▪ active - Indicates if the element is active. Returns 0 (not active) or 1 (active). ▪ baseCopy - The text to be translated. ▪ id - The challenge ID. ▪ parentId - The ID of the parent catalog item. |
| locale | The name of the exported locale. |
| site | The Nitro site. |
| version | The export file version. |
| value | The translated value. |

Sample Request


method=admin.exportLocaleTranslations&locale=es-ES

Sample Response

```
{ "response": { "root": { "data": [], "element": [ { "text": { "value": "" }, "id": "Challenge.name.1316225", "baseCopy": "11111111", "active": 0 }, { "text": { "value": "" }, "id": "Challenge.name.1213825", "baseCopy": "View Training Videos2016-05-06_13-32-06", "active": 0 }, { "text": { "value": "" }, "id":
```

```
"Rule.description.2047104", "baseCopy": "serving the southwest",
"parentId": 1415924 } ], "site": "Testing", "apiKey": "TEST",
"locale": "es-ES", "version": 2 } }, "responseType": {}, "isError":
false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.getActionTags Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Gets a list of all action tags.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| sortBy | No | String | None | 1 - 255 chars | The order to sort the items in the response (date, name). |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------------|---|
| description | The action's description. |
| folderId | The ID of the folder the action is located in. |
| iconUrl | The URL to the icon used for the action. |
| id | The action's ID. |
| isActionTag | Indicates if the tag is set on an action: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| isEditable | Indicates if the action is editable: |

| Response | Description |
|--------------------------|---|
| | <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| lowSecurity | Indicates if the action has low security. <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| name | The action name. |
| notificationId | The ID of the notification assigned to the action. |
| prefixMatch | Indicates if prefix matching is enabled: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| rateLimit | The maximum number of times this action can be logged in seconds, minutes, hours, or days. |
| serviceActionType | Previously used for social actions. |
| serviceType | Previously used for social actions. |
| status | Indicates if the action is active. |


Sample Request

method=admin.getActionTags

Sample Response

```
{ "response": { "tags": { "tag": [ { "id": 2703201665, "clientId": "150401|-206|13430067112101023", "name": "post_document", "description": "Post a document", "rateLimit": 0, "isActionTag": 1, "lowSecurity": 0, "prefixMatch": 0, "category": 1, "serviceType": 1, "serviceActionType": 1, "isEditable": 0, "folderId": 573281, "order": 0, "status": "active" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.getAdminLog Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Gets the history of operations admin users have executed in Nitro Studio.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|---------|---------------|------------------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| endTime | No | Integer | None | 0 - 2 ³¹ -1 | The end of the date range you want to show records for (UNIX timestamp). |
| startTime | No | Integer | None | 0 - 2 ³¹ -1 | The start of the date range you want to show records for (UNIX timestamp). |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------------|---|
| description | The operation performed by the admin. |
| id | The ID of the operation. |
| ipAddress | The IP address of the user logging the operation. |
| ts | The date and time the operation was performed (UNIX timestamp). |
| userName | The name of the user completing the operation. |

Sample Request

method=admin.getAdminLog


Sample Response

```
{ "response": { "adminLogRecord": { "adminLogEntry": [ {
"adminLogDetails": {}, "id": 85285992, "ts": 1507244222, "ipAddress":
"11.11.111.111", "userName": "judi.ingram@bunchball.com",
"description": "Login by admin user judi.ingram@bunchball.com" }, {
"adminLogDetails": { "adminLogDetail": [ { "detailName": "challenge",
"detailValue": "Test Challenge for Reordering", "detailOldValue": ""
}, { "detailName": "description", "detailValue": "column 2 rule 3" },
{ "detailName": "Tag", "detailValue": "COURSE_COMPLETED",
"detailOldValue": "" }, { "detailName": "goal", "detailValue": "2",
"detailOldValue": "0" }, { "detailName": "displayOrder",
```



```
"detailValue": "2", "detailOldValue": "0" }, { "detailName":
"timeRange", "detailValue": "3600", "detailOldValue": "0" }, {
"detailName": "prereqOperator", "detailValue": "or",
"detailOldValue": "or" }, { "detailName": "operator", "detailValue":
"GE", "detailOldValue": "" } ] } }, "id": 85284508, "ts": 1507226664,
"ipAddress": "111.11.1.1", "userName": "judi.ingram@bunchball.com",
"description": "Created Rule for Challenge Test Challenge for
Reordering" } ] } } }, "responseType": {}, "isError": false,
"httpError": 0, "httpsStatusCode": 200 }
```

admin.getAdminUsers Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Gets a list of all admin users.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------|--|
| adminType | The admin user type. Returns "2" for Full Admin. |
| userId | The user's email address. |


Sample Request

```
method=admin.getAdminUsers
```

Sample Response

```
{ "response": { "users": { "user": [ { "userId":
"tom.smith@mail.com", "adminType": 2 }, { "userId":
"david.brown@mail.com", "adminType": 2 }, { "userId":
"betsyross@mail.com", "adminType": 2 } ] } }, "responseType": {},
"isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.getAllCatalogContents Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Gets a list of all catalogs, categories, and items.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-------------------|---|
| catalogs | The catalog details: <ul style="list-style-type: none"> ▪ id - The catalog ID. ▪ name - The catalog name. ▪ type - The catalog type: <ul style="list-style-type: none"> ▪ 1 - avatar ▪ 3 - store |
| categories | The category details within the catalog: <ul style="list-style-type: none"> ▪ catalogId - The catalog's ID number. |

| Response | Description |
|---------------------|--|
| | <ul style="list-style-type: none"> ▪ childCategoryIds - The child categories' IDs (when nested categories exist). ▪ createTime - The date/time the category was created (UNIX timestamp). ▪ id - The category ID. ▪ name - The category name. ▪ order - The order the category appears in the catalog. |
| catalogItems | <p>The item details within the catalog:</p> <ul style="list-style-type: none"> ▪ active - Indicates if the item is active within the catalog: <ul style="list-style-type: none"> ▪ 0 - Not active ▪ 1 - Active ▪ background - Indicates if the item is a background item: <ul style="list-style-type: none"> ▪ 0 - Not a background item ▪ 1 - A background item ▪ catalogItemId - The item's ID number. ▪ catalogName - The name of the catalog the item is in. ▪ category - The names of the categories the item is in. ▪ categoryIds - The ID's of the category the item is in. ▪ countRemaining - The number of items remaining for purchase. ▪ createTime - The date/time the item was created in the catalog (UNIX timestamp). ▪ customData - The custom data associated with the item. ▪ description - The item's description. ▪ doCallback - Indicates if a callback is triggered when a user purchases the item: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes ▪ endTime - The date/time the item is no longer available (UNIX timestamp). ▪ fullUrl - The URL to the full size image of the item. ▪ id - The item's ID number. ▪ lastUpdated - The date/time the item was last updated (UNIX timestamp). ▪ maxOwnedCountPerUser - The maximum number of the item a user can purchase. ▪ name - The item's name. ▪ orders - The order the item appears in the catalog. ▪ ownedCount - The number of items of this type that users own. |

| Response | Description |
|----------|---|
| | <ul style="list-style-type: none"> ▪ pointCategoryId - The ID of the point category that points are deducted from when the user purchases this item. ▪ pointCategoryName - The name of the point category that points are deducted from when the user purchase this item. ▪ price - The cost of the item (in points). ▪ realItem - Indicates if the item shows in the user's Owned Items list in widgets: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes ▪ startTime - The date/time the item becomes available (UNIX timestamp). ▪ tags - The tags associated with the item. ▪ thumbUrl - The URL to the thumbnail size image of the item. ▪ zOrder - The item's Z-Order number. |

Sample Request


method=admin.getAllCatalogContents

Sample Response

```
{ "response": { "catalogRecord": [ { "catalogs": { "catalog": [ {
"id": 433025, "clientId": "150401|-478|14032517927625626", "name":
"Store", "type": 3 }, { "id": 433205, "clientId": "150401|-
958|174346326905493", "name": "Sports", "type": 1 } ] },
"categories": { "category": [ { "id": 2895873, "clientId":
"150401|307|14032097292573483", "name": "Shopping", "catalogId":
433025, "order": 0, "canUnselect": "true", "childCategoryIds": "",
"createTime": 1472752128 }, { "id": 2896057, "clientId": "150401|-
726|177460674451836", "name": "Cheer", "catalogId": 433205, "order":
1, "canUnselect": "true", "childCategoryIds": "", "createTime":
1482429952 }, { "id": 2896060, "clientId":
"150401|305|183992298300530", "name": "Basketball", "catalogId":
433205, "order": 3, "canUnselect": "true", "childCategoryIds": "",
"zOrder": 2, "createTime": 1482436224 }, { "id": 2896061, "clientId":
"150401|754|1380908821526849", "name": "Football", "catalogId":
433205, "order": 1, "canUnselect": "true", "childCategoryIds": "",
"createTime": 1483633152 } ] }, "catalogItems": { "catalogItem": [ {
"id": 31211596, "clientId": "150401|-579|1380877875326593",
"description": "", "customData": "", "background": 0, "designer": "",
"zOrder": 0, "countRemaining": -1, "name": "Shoes", "catalogName":
"Sports", "price": 0, "categoryIds": "2896057", "category": "Cheer",
"tags": "", "canUnselect": "null", "active": 1, "realItem": 0,
```

```
"doCallback": 0, "passExtraData": 0, "ownedCount": 0, "orders": "0",
"createTime": 1483633408, "pointCategoryId": 295169,
"pointCategoryName": "Points", "lastUpdated": "1483633408",
"catalogItemId": 31211596, "startTime": 0, "endTime": 0,
"maxOwnedCountPerUser": 0 }, { "id": 31208834, "clientId":
"150401|639|326288024360054", "description": "Company tee",
"customData": "", "background": 0, "thumbUrl":
"https://www.bunchball.com/sites/files/badge_logo_web.png",
"designer": "", "zOrder": 0, "countRemaining": -1, "name": "Company
T-shirt", "catalogName": "Store", "fullUrl":
"https://www.bunchball.com/sites/files/badge_logo_web.png", "price":
0, "categoryIds": "2895873", "category": "Shopping", "tags": "",
"canUnselect": "null", "active": 1, "realItem": 1, "doCallback": 0,
"passExtraData": 0, "ownedCount": 6, "orders": "0", "createTime":
1473191808, "pointCategoryId": 295682, "pointCategoryName": "Funny
Money", "lastUpdated": "1473191808", "catalogItemId": 31208834,
"startTime": 0, "endTime": 0, "maxOwnedCountPerUser": 0 } ] } } ] },
"responseType": {}, "isError": false, "httpError": 0,
"httpsStatusCode": 200 }
```

admin.getCatalog Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Gets all the items in the specified catalog. To return all items in all catalogs, use [admin.getCatalogs](#).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| catalogId | YES | Long | None | N/A | The ID of the catalog to get. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|---------------------|--|
| catalogs | <p>The catalog details:</p> <ul style="list-style-type: none"> ▪ id - The catalog ID. ▪ name - The catalog name. ▪ type - The catalog type: <ul style="list-style-type: none"> ▪ 1 - avatar ▪ 3 - store |
| categories | <p>The category details within the catalog:</p> <ul style="list-style-type: none"> ▪ canUnselect - Returns true or false. ▪ catalogId - The catalog's ID number. ▪ childCategoryIds - The child categories' IDs (when nested categories exist). ▪ createTime - The date/time the category was created (UNIX timestamp). ▪ id - The category ID. ▪ name - The category name. |
| catalogItems | <p>The item details within the catalog:</p> <ul style="list-style-type: none"> ▪ active - Indicates if the item is active within the catalog: <ul style="list-style-type: none"> ▪ 0 - Not active ▪ 1 - Active ▪ background - Indicates if the item is a background item: <ul style="list-style-type: none"> ▪ 0 - Not a background item ▪ 1 - A background item ▪ catalogItemId - The item's ID number. ▪ catalogName - The name of the catalog the item is in. ▪ category - The names of the categories the item is in. ▪ categoryIds - The ID's of the category the item is in. ▪ countRemaining - The number of items remaining for purchase. ▪ createTime - The date/time the item was created in the catalog (UNIX timestamp). ▪ customData - The custom data associated with the item. ▪ description - The item's description. ▪ doCallback - Indicates if a callback is triggered when a user purchases the item: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |

| Response | Description |
|----------|---|
| | <ul style="list-style-type: none"> ▪ endTime - The date/time the item is no longer available (UNIX timestamp). ▪ fullUrl - The URL to the full size image of the item. ▪ id - The item's ID number. ▪ lastUpdated - The date/time the item was last updated (UNIX timestamp). ▪ maxOwnedCountPerUser - The maximum number of the item a user can purchase. ▪ name - The item's name. ▪ orders - The order the item appears in the catalog. ▪ ownedCount - The number of items of this type that users own. ▪ pointCategoryId - The ID of the point category that points are deducted from when the user purchases this item. ▪ pointCategoryName - The name of the point category that points are deducted from when the user purchase this item. ▪ price - The cost of the item (in points). ▪ realItem - Indicates if the item shows in the user's Owned Items list in widgets: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes ▪ startTime - The date/time the item becomes available (UNIX timestamp). ▪ tags - The tags associated with the item. ▪ thumbUrl - The URL to the thumbnail size image of the item. ▪ zOrder - The item's Z-Order number. |

Sample Request


```
method=admin.getCatalog&catalogId=433025
```

Sample Response

```
{ "response": { "catalog": { "categories": { "category": [ { "id": 2895873, "clientId": "150401|307|14032097292573483", "name": "Shopping", "catalogId": 433025, "canUnselect": "true", "childCategoryIds": "", "createTime": 1472752128 } ] }, "catalogItems": { "catalogItem": [ { "id": 31208834, "clientId": "150401|639|326288024360054", "description": "Company tee", "customData": "", "background": 0, "thumbUrl": "https://www.bunchball.com/assets/bb_logo_tee.png", "designer": "", "zOrder": 0, "countRemaining": -1, "name": "Company T-shirt", "catalogName": "Store", "fullUrl":
```

```
"https://www.bunchball.com/assets/bb_logo_tee.png", "price": 0,
"categoryIds": "2895873", "category": "Shopping", "tags": "",
"canUnselect": "null", "active": 1, "realItem": 1, "doCallback": 0,
"passExtraData": 0, "ownedCount": 7, "orders": "0", "createTime":
1473191808, "pointCategoryId": 295682, "pointCategoryName": "Funny
Money", "lastUpdated": "1473191808", "catalogItemId": 31208834,
"startTime": 0, "endTime": 0, "maxOwnedCountPerUser": 0 } ] }, "id":
433025, "clientId": "150401|-478|14032517927625626", "name": "Store",
"type": 3 } }, "responseType": {}, "isError": false, "httpError": 0,
"httpsStatusCode": 200 }
```

admin.getCatalogs Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Gets all catalogs and the items in the catalogs. To find items in an individual catalog, use [admin.getCatalog](#).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-----------------|---|
| catalogs | The catalog details: <ul style="list-style-type: none"> ▪ id - The catalog ID. ▪ name - The catalog name. ▪ type - The catalog type: <ul style="list-style-type: none"> ▪ 1 - avatar |

| Response | Description |
|---------------------|--|
| | <ul style="list-style-type: none"> ▪ 3 - store |
| categories | <p>The category details within the catalog:</p> <ul style="list-style-type: none"> ▪ canUnselect - Returns true or false. ▪ catalogId - The catalog's ID number. ▪ childCategoryIds - The child categories' IDs (when nested categories exist). ▪ createTime - The date/time the category was created (UNIX timestamp). ▪ id - The category ID. ▪ name - The category name. |
| catalogItems | <p>The item details within the catalog:</p> <ul style="list-style-type: none"> ▪ active - Indicates if the item is active within the catalog: <ul style="list-style-type: none"> ▪ 0 - Not active ▪ 1 - Active ▪ background - Indicates if the item is a background item: <ul style="list-style-type: none"> ▪ 0 - Not a background item ▪ 1 - A background item ▪ catalogItemId - The item's ID number. ▪ catalogName - The name of the catalog the item is in. ▪ category - The names of the categories the item is in. ▪ categoryIds - The ID's of the category the item is in. ▪ countRemaining - The number of items remaining for purchase. ▪ createTime - The date/time the item was created in the catalog (UNIX timestamp). ▪ customData - The custom data associated with the item. ▪ description - The item's description. ▪ doCallback - Indicates if a callback is triggered when a user purchases the item: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes ▪ endTime - The date/time the item is no longer available (UNIX timestamp). ▪ fullUrl - The URL to the full size image of the item. ▪ id - The item's ID number. ▪ lastUpdated - The date/time the item was last updated (UNIX timestamp). ▪ maxOwnedCountPerUser - The maximum number of the item a user can purchase. |

| Response | Description |
|----------|---|
| | <ul style="list-style-type: none"> ▪ name - The item's name. ▪ orders - The order the item appears in the catalog. ▪ ownedCount - The number of items of this type that users own. ▪ pointCategoryId - The ID of the point category that points are deducted from when the user purchases this item. ▪ pointCategoryName - The name of the point category that points are deducted from when the user purchase this item. ▪ price - The cost of the item (in points). ▪ realItem - Indicates if the item shows in the user's Owned Items list in widgets: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes ▪ startTime - The date/time the item becomes available (UNIX timestamp). ▪ tags - The tags associated with the item. ▪ thumbUrl - The URL to the thumbnail size image of the item. ▪ zOrder - The item's Z-Order number. |

Sample Request


method=admin.getCatalogs

Sample Response

```
{ "response": { "catalog": { "categories": { "category": [ { "id":
2895745, "clientId": "150401|869|13780055334802303", "name":
"Voting", "catalogId": 432897, "canUnselect": "true",
"childCategoryIds": "", "createTime": 1472500096 } ] },
"catalogItems": { "catalogItem": [ { "id": 31208577, "clientId":
"150401|-892|13780338085366301", "description": "Logo1",
"customData": "", "background": 0, "thumbUrl":
"C:\\Files\\logo1.png", "designer": "", "zOrder": 0,
"countRemaining": -1, "name": "Logo1", "catalogName": "LogoVote",
"fullUrl": "C:\\Files\\logo1.png", "price": 0, "categoryIds":
"2895745", "category": "Voting", "tags": "", "canUnselect": "null",
"active": 1, "realItem": 0, "doCallback": 0, "passExtraData": 0,
"ownedCount": 0, "orders": "0", "createTime": 1472500352,
"pointCategoryId": 295169, "pointCategoryName": "Points",
"lastUpdated": "1472500096", "catalogItemId": 31208577, "startTime":
0, "endTime": 0, "maxOwnedCountPerUser": 0 }, { "id": 31208705,
"clientId": "150401|741|13780820691909443", "description": "Logo2",
"customData": "", "background": 0, "thumbUrl":
```

```
"C:\\Files\\logo2.png", "designer": "", "zOrder": 0,
"countRemaining": -1, "name": "Logo2", "catalogName": "LogoVote",
"fullUrl": "C:\\Files\\logo2.png", "price": 0, "categoryIds":
"2895745", "category": "Voting", "tags": "", "canUnselect": "null",
"active": 1, "realItem": 0, "doCallback": 0, "passExtraData": 0,
"ownedCount": 4, "orders": "0", "createTime": 1472500480,
"pointCategoryId": 295169, "pointCategoryName": "Points",
"lastUpdated": "1472500096", "catalogItemId": 31208705, "startTime":
0, "endTime": 0, "maxOwnedCountPerUser": 0 } ] }, "id": 432897,
"clientId": "150401|-830|13780026663526760", "name": "LogoVote",
"type": 2 } }, "responseType": {}, "isError": false, "httpError": 0,
"httpsStatusCode": 200 }
```

admin.getChallengeDependencyTree Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Allows you to pass in a challenge ID and receive a dependency tree response showing which challenges have the specified challenge as a prerequisite and which challenges are a prerequisite of each prerequisite.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|---------|---------------|------------------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| challengeId | YES | Integer | 0 | 0 - 2 ³¹ -1 | The ID of the challenge that you want to get a dependency list for. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------|---|
| dependsOn | The challenge that is a prerequisite to the listed challenge. |

| Response | Description |
|-------------|---------------------|
| id | The challenge ID. |
| name | The challenge name. |


Sample Request

```
method=admin.getChallengeDependencyTree&challengeId=11316
```

Sample Response

```
{ "response": [ { "id": 11309, "name": "Become Keepersite Certified",
"dependsOn": [ { "id": 9792, "name": "Pass the Site Knowledge Quiz",
"dependsOn": [ ] } ] } ], "responseType": {}, "isError": false,
"httpError": 0, "httpStatusCode": 200 }
```

admin.getChallenges Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Gets a list of all challenges.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|------------------------|----------|---------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| includeRules | No | Boolean | true | true/false | If true, returns all rules that belong to the returned challenges. |
| showMetricGoals | No | Boolean | true | true/false | Shows/hides the metric rule goals associated with the challenge. |
| sortBy | No | String | None | 1 - 255 chars | The order to sort the items in the response (date, name). |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------------------|---|
| actionPhrase | The challenge's action phrase. |
| actionUrl | The URL associated with the challenge (the page where the user must go to complete the challenge). |
| activeFlag | Indicates if the challenge is active: <ul style="list-style-type: none"> ▪ 0 - Not active ▪ 1 - Active |
| applyMultiplier | The points multiplier in use for the challenge. |
| callbackFlag | Indicates if the challenge is enabled to trigger an achievement callback when completed: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| ChallengeReward | The rewards associated with the challenge: <ul style="list-style-type: none"> ▪ type - Indicates the reward type. Returns point, item, or notification. ▪ details - Additional reward information. For points, the details may be something like "285424 50" with the number of points awarded after the pipe (). In this example, 50 points are awarded to the user after completing the challenge. ▪ itemName - The name of the point category, catalog item, or notification. |
| customData | The challenge's custom data. |
| dailyAchievementLimit | The number of times the challenge can be completed in a day. Returns 0 for unlimited. |
| dateIssued | The date the challenge becomes active (UNIX timestamp). |
| description | The challenge's description. |
| endTime | The date the challenge ends (UNIX timestamp). Returns 0 if no end date is set. Typically used to display the expiration date of the challenge to the user. |
| folderId | The ID of the folder the challenge is located in. |
| fullUrl | URL to the full size image associated with the challenge. |
| groupFlag | Indicates if it's a group challenge: <ul style="list-style-type: none"> ▪ 0 - Not a group challenge |

| Response | Description |
|-------------------------|---|
| | <ul style="list-style-type: none"> ▪ 1 - A group challenge |
| groupPointAward | The number of points awarded to the group for completing the challenge. |
| hideUntilEarned | <p>Indicates if the challenge is hidden until complete:</p> <ul style="list-style-type: none"> ▪ 0 - Not hidden ▪ 1 - Hidden <p>Repeatable, hidden challenges show in both complete and incomplete once initially completed.</p> |
| iconUrl | The URL to the icon used for the challenge. |
| id | The challenge ID. |
| name | The challenge name. |
| order | <p>Indicates if the challenge rules must be completed in order:</p> <ul style="list-style-type: none"> ▪ 0 - Do not have to be completed in order. ▪ 1 - Must be completed in order. |
| pointAward | The number of points awarded for completing the challenge. |
| pointCategory | For challenges with point rewards, returns the name of the point category the points are awarded to. |
| pointCategoryId | The ID of the point category. |
| preferences | Used for metric goals where a segment determines which value to use in the goal. |
| preferenceValues | The values for the preferences list. |
| prereqMatchType | Indicates if the prerequisite is and or or. |
| repeatable | <p>Indicates if the challenge is repeatable:</p> <ul style="list-style-type: none"> ▪ 0 - Not repeatable ▪ 1 - Repeatable <p>Repeatable, hidden challenges show in both complete and incomplete once initially completed.</p> |
| rules | <p>The rules associated with the challenge:</p> <ul style="list-style-type: none"> ▪ actionTag - The name of the action associated with the rule. ▪ displayOrder - The rule's order within the column. ▪ description - The rule's action description. ▪ goal - The action value goal required by the rule. For Less than scenarios, the action value lists this value as the number configured in Nitro Studio. Must be used with operator for Less than scenarios. |

| Response | Description |
|----------|---|
| | <ul style="list-style-type: none"> ▪ id - The rule ID. ▪ itemName - The action name. ▪ metadata - The metadata associated with the rule. Returns name and value pair. ▪ metricRuleFilter - The metric goals associated with the rule. <ul style="list-style-type: none"> ▪ name - The type of metric rule. ▪ value - Indicates if the metric rule is site or user based. ▪ countGoal - The number of times the action goal is to be satisfied. ▪ valueLowerBound - Returns the greater than value of the lower bound. ▪ valueUpperBound - Returns the less than value of the upper bound. ▪ lowerBoundInclusive - Indicates if the rule should be true or false when it equals the lower bound value. ▪ upperBoundInclusive - Indicates if the rule should be true or false when it equals the upper bound value. ▪ info - Information specified in the rule's info field. ▪ operator - The action goal value: <ul style="list-style-type: none"> ▪ GE - Greater than or equal to ▪ GT - Greater than ▪ LT - Less than ▪ LE - Less than or equal to ▪ EQ - Equal to ▪ prereqOperator - Indicates if one or all prerequisites must be met. Returns and or or. ▪ prereqRuleIds - The ID of the rule that must be completed before this rule (for multi-rule challenges). ▪ serviceActionType - Previously used for social actions. ▪ serviceType - Previously used for social actions. ▪ sortOrder - An integer value indicating the rules' order within the challenge. ▪ tagId - The ID of the tag associated with the challenge. ▪ type - Indicates the action goal type: <ul style="list-style-type: none"> ▪ sum - Running total ▪ count - Number of calls |

| Response | Description |
|----------------------|--|
| ruleMatchType | Indicates if the rule operator is "and" or "or". |
| serviceType | Previously used for social actions. |
| startTime | The challenge start date (UNIX timestamp). |
| tags | The challenge's tags. |
| thumbUrl | For challenges with badge rewards, returns the URL of the thumb badge image. |
| version | For internal use and backward compatibility. |


Sample Request

method=admin.getChallenges

Sample Response

```
{ "response": { "challenges": { "challenge": [ { "rules": { "rule": [
{ "metadata": {}}, "metricRuleFilter": { "metricRuleFilterDetails": {
"metricRuleFilterDetail": [ { "name": "site", "value": "site",
"countGoal": 1, "valueLowerBound": "1.0", "valueUpperBound": "5.0",
"lowerBoundInclusive": true, "upperBoundInclusive": true, "info":
"info field input" } ] }, "filter": "site" }, "id": 6996687, "tagId":
2734243631, "actionTag": "add_sale", "operator": "GE", "description":
"Close 5 Sales", "goal": 1, "timeRange": 3600, "type": "metricGoal",
"sortOrder": 0, "displayOrder": 0, "serviceType": "nitro",
"serviceActionType": "none", "prereqRuleIds": "", "prereqOperator":
"or", "custom": "2734243631", "itemName": "add_sale" }, ] },
"challengeRewards": { "challengeReward": [ { "type": "notification",
"details": "709121", "itemName": "Good Job" } ] }, "id": 4805220,
"type": "user", "name": "Close 5 Sales", "description": "Close 5
Sales this Month", "activeFlag": 1, "featuredFlag": 0,
"callbackFlag": 0, "serviceType": "1", "order": 0, "pointCategory":
"Points", "notificationId": 709121, "thumbUrl":
"https://s3.amazonaws.com/bunchball/7283d2deb/userUploadpng158d72.png",
"fullUrl":
"https://s3.amazonaws.com/bunchball/7283df78eb/userUploadpng1589bd.png",
"pointAward": 0, "dateIssued": "1483712260", "applyMultiplier":
0, "repeatable": 0, "hideUntilEarned": 0, "tags": "", "startTime":
1483712260, "endTime": 0, "ruleMatchType": 0, "prereqMatchType":
"or", "preferences": "", "preferenceValues": "",
"preferencesMatchType": "or", "groupFlag": 0, "groupPointAward": 0,
"dailyAchievementLimit": 1, "version": "5.2" } ] } }, "responseType":
{}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```


admin.getFolders Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Gets a list of all folders of a specified type.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| type | YES | String | None | 1 - 255 chars | The folder type. Accepts action, challenge, or group. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-----------------|--|
| id | The folder's ID. |
| name | The folder name. |
| type | The folder type. |
| order | The order the folder appears in the action, challenge, or group view list. |
| created | The date the folder was created (UNIX timestamp). |
| parentId | The ID of the parent folder. |


Sample Request

```
method=admin.getFolders&type=action
```

Sample Response

```
{ "response": { "folders": { "folder": [ { "id": 573281, "name":
"Collaborating", "type": "action", "order": 0, "created": 1485554959,
"clientId": "150401|-888|331620241011772" }, { "id": 573282, "name":
"Company Events", "type": "action", "order": 0, "created":
1485554959, "clientId": "150401|156|331632305121358" }, { "id":
573280, "name": "Learning", "type": "action", "order": 0, "created":
1485554959, "clientId": "150401|-432|331602628635415" } ] } },
"responseType": {}, "isError": false, "httpError": 0,
"httpsStatusCode": 200 }
```

admin.getGroups Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Returns all groups for the site or only groups within a specified folder.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|--|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| folderName | No | String | None | 1 - 255 chars | The folder that contains the groups you want to return. |
| returnAll | No | String | None | 1 - 255 chars | Specifies whether to return everything as a flat result. When set to true, returns all groups regardless of the values in the returnCount and startIndex parameters. For example, if you have 200 groups and want to see all groups, set returnAll to true. If you only want to get groups 100-200, set returnCount to 100 and startIndex to 100. If you only want to |

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|---------|---------------|------------------------|--|
| | | | | | get groups 150-200, set returnCount to 50 and startIndex to 150. |
| returnCount | No | Integer | 10 | 1 - 100 | The number of records to return. |
| sortBy | No | String | None | 1 - 255 chars | The order to sort the items in the response. Accepts date or name. |
| startIndex | No | Integer | 0 | 0 - 2 ³¹ -1 | A 0-based index specifying which row from the results to return data from. Can be used to page through the result set. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------------|---|
| count | The number of users in the group. |
| folderId | The ID of the group's folder. |
| folderName | The group's folder name. |
| id | The group ID. |
| name | The group name. |
| preferences | Lists the group's preferences: <ul style="list-style-type: none"> ▪ name - The preference name. ▪ value - The preference value. ▪ groupId - The group ID. |

Sample Request


method=admin.getGroups

Sample Response

```
{ "response": { "groups": { "group": [ { "preferences": {}, "id": 1959809, "clientId": "1|-682|69348968600293", "name": "country:Canada", "count": 0 }, { "preferences": {}, "id": 1959937, "clientId": "1|178|69388934983063", "name": "country:Chile", "count": 0 }, { "preferences": {}, "id": 1958426, "clientId": "1|-170|63509444506059084", "name": "g1", "count": 0 }, { "preferences": { "groupPreference": [ { "name": "testPref", "value": "testPref", "groupId": 1958430 } ] }, "id": 1958430, "clientId": "1|-
```

```
669|7926449624452763", "name": "test3", "count": 0 } ] } },
"responseType": {}, "isError": false, "httpError": 0,
"httpsStatusCode": 200 }
```

admin.getGroups2 Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Returns a specific group. Use [admin.getGroups](#) to return all groups.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|---------|---------------|------------------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| groupId | No | Integer | 0 | 0 - 2 ³¹ -1 | The group ID to query for. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-----------------------|---|
| count | The number of users in the group. |
| groupClassId | The ID of the group class. |
| groupClassName | The group class name. |
| id | The group ID. |
| name | The group name. |
| preferences | Lists the group's preferences: <ul style="list-style-type: none"> ▪ name - The preference name. ▪ value - The preference value. ▪ groupId - The group ID. |


Sample Request

```
method=admin.getGroups2&asyncToken=&groupId=2189
```

Sample Response

```
{ "response": { "detailedGroups": { "folders": {}, "groups": {
"group": [ { "preferences": {}, "id": 2189, "clientId": "1|-
119|5206865493961", "name": "nick_gc_test_group", "count": 0,
"groupClassId": 780, "groupClassName": "Nick_test_gc" } ] } },
"returnCount": 1, "startIndex": 0, "totalRecords": 6 },
"responseType": {}, "isError": false, "httpError": 0,
"httpStatusCode": 200 }
```

admin.getNotifications Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Gets a list of notifications configured in Nitro Studio, along with the notification ID and other configuration information.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|---------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| returnAll | No | String | None | 1 - 255 chars | Indicates if all notifications should be returned. Accepts true or false. |
| returnCount | No | Integer | 10 | 1 - 100 | The number of records to return. |
| sortBy | No | String | None | 1 - 255 chars | The order to sort the items in the response. Accepts date or name. |

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|---------|---------------|---------------|--|
| startIndex | No | Integer | 0 | N/A | A 0-based index specifying which row from the results to return data from. Can be used to page through the result set. |
| type | No | String | None | 0 - 255 chars | The notification type. Accepts pop-up and email. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------------|--|
| active | Indicates if the notification is active: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| challenges | The challenges that use the notification: <ul style="list-style-type: none"> ▪ name - The challenge name. ▪ id - The challenge ID. |
| corner | The position on the screen where the notification appears (toast only). |
| delay | The length of time before the notification appears (toast only). |
| description | A description of the notification. Not user facing. |
| direction | How the notification appears on the screen (toast only). <ul style="list-style-type: none"> ▪ Vertical - Rises from the bottom or descends from the top of the screen. ▪ Horizontal - Slides in from the side of the screen. |
| doFeed | Indicates if the notification is included in your site's newsfeed. <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| doToast | Indicates if the notification displays as a pop-up on your site. <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| duration | The length of time, in seconds, the notification displays on the screen (toast only). |
| endTime | The date/time the notification stops appearing. |
| frequency | How often the notification displays (toast only). If a Start date and End date are set, the notification displays at the selected frequency within the time/date range. <ul style="list-style-type: none"> ▪ Unlimited - Displays every time the action starts or the challenge completes. |

| Response | Description |
|-----------------------------|---|
| | <ul style="list-style-type: none"> ▪ Once Per Day - Displays every time the action starts or the challenge completes. If the action or challenge is performed multiple times a day, then only shows the notification once per day. ▪ Only Once - Displays only once. |
| id | The notification ID. |
| name | The notification name. |
| notificationSettings | The notification display settings: <ul style="list-style-type: none"> ▪ name - The name of the field. ▪ value - The value of the field. ▪ defaultValue - The default field value. ▪ type - The field type. ▪ id - The field ID. |
| order | The order the notification should display if multiple notifications are pending. |
| startTime | The date/time the notification starts appearing. |
| styleId | The ID of the notification style assigned to this notification. |
| styleName | The name of the notification style assigned to this notification. |
| tags | The tags used to identify this notification in filtering and reporting. |
| type | The notification type. Returns popup or feed. |
| updated | The date the notification was updated. |
| version | For internal use and backward compatibility. |

Sample Request


method=admin.getNotifications&returnCount=1&sortBy=name

Sample Response

```
{ "response": { "notifications": { "notification": [ { "challenges":
{ "minis": [ { "name": "Add a New Beneficiary", "id": 4805220 }, {
"name": "AttendWorkshop", "id": 4802302 }, { "name": "Collaborate
with others", "id": 4802234 }, { "name": "Log in to the mobile app",
"id": 4805135 }, { "name": "Post a Document Contest", "id": 4753793
}, { "name": "TrainingVideos", "id": 4799489 } ] },
"notificationSettings": { "notificationSetting": [ { "name": "90x90
Image", "value": "", "defaultValue":
"http://bunchball.com/assets/avatar/thumb.png?size=90", "type":
"HTML", "id": 0 }, { "name": "Title", "value": "WINNER WINNER",
```

```
"defaultValue": "Well done!", "type": "string", "id": 501505 }, {
"name": "Subtitle", "value": "Good Job!", "defaultValue": "Nice
work.", "type": "string", "id": 501633 } ] }, "id": 709121,
"clientId": "150401|211|13960786292242992", "name": "Good Job",
"type": "popup", "description": "Good job completing your challenge",
"updated": 1472681666, "styleId": 1996289, "active": 1, "doFeed": 1,
"doToast": 1, "doShare": 0, "version": 0, "styleName": "Badge of
Honor", "startTime": 0, "endTime": 0, "tags": "", "order": 0,
"duration": "5", "delay": "0", "frequency": "Unlimited", "corner":
"Bottom Right", "direction": "Vertical" } ] }, "returnCount": 1,
"startIndex": 0, "totalRecords": 7 }, "responseType": {}, "isError":
false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.getNotificationStyles Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Returns all notification styles for the site.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|---------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| returnAll | No | String | None | 1 - 255 chars | Indicates if all notifications should be returned. Accepts true or false. |
| returnCount | No | Integer | 10 | 0 - 100 chars | The number of records to return. |
| sortBy | No | String | None | 1 - 255 chars | The order to sort the items in the response. Accepts date or name. |
| startIndex | No | Integer | 0 | N/A | A 0-based index specifying which row from the results to return data from. Can be used to page through the result set. |

| Name | Required | Type* | Default Value | Range | Comments |
|-------------|----------|--------|---------------|---------------|---|
| type | No | String | None | 0 - 255 chars | The notification type. Accepts popup and email. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------------------|--|
| corner | The position on the screen where the notification appears (popup only). |
| css | The css applied to notification's using this style. |
| delay | The length of time before the notification appears (popup only). |
| description | The description of the notification style. |
| direction | How the notification appears on the screen (popup only). <ul style="list-style-type: none"> ▪ Vertical - Rises from the bottom or descends from the top of the screen. ▪ Horizontal - Slides in from the side of the screen. |
| duration | The length of time, in seconds, a notification displays on the screen (popup only). |
| frequency | How often a notification displays (toast only). If a Start date and End date are set, the notification displays at the selected frequency within the time/date range. <ul style="list-style-type: none"> ▪ Unlimited - Displays every time the action starts or the challenge completes. ▪ Once Per Day - Displays every time the action starts or the challenge completes. If the action or challenge is performed multiple times a day, then only shows the notification once per day. ▪ Only Once - Displays only once. |
| html | The HTML of the notification. |
| id | The ID of the notification. |
| isDefault | Indicates if this is the default notification. Returns true or false. |
| name | The notification name. |
| notification settings | The name, value, defaultValue, type, and ID of each of the elements within the notification. |
| returnCount | The number of notification styles returned. |
| startIndex | The row from the results data was returned from. |
| type | The notification type. Returns email or popup. |
| thumbUrl | The URL to the notification's thumbnail image. |
| totalRecords | The total number of notification styles on the site. |

Sample Request

method=admin.getNotificationStyles&type=popup&returnCount=1

Sample Response

```
{ "response": { "notificationStyles": { "notificationStyle": [ {
"notificationSettings": { "notificationSetting": [ { "name": "90x90
Image", "value": "", "defaultValue":
"http://dynamic.bunchball.com/assets/{nitro_apiKey}/{nitro_userId}/th
umb.png?size=90", "type": "HTML", "id": 0 }, { "name": "Title",
"value": "", "defaultValue": "AENEAN TRISTIQUÉ ARCU EGET QUAMON",
"type": "string", "id": 0 }, { "name": "Subtitle", "value": "",
"defaultValue": "Lorem ipsum dolor sit amet, consectetur adip iscing
elit sagittis.", "type": "string", "id": 0 }, { "name": "Message",
"value": "", "defaultValue": "Lorem ipsum dolor sit ame. ", "type":
"string", "id": 0 } ] }, "id": 1996289, "clientId": "150401|-
651|13961858310824057", "name": "Badge of Honor", "type": "popup",
"html": "", "css": "div.main01 { clear: both; margin: 0px; width:
329px; text-align: left; line-height: 16px; } div.row { clear: both;
line-height: 0; } div.cell { display: table-cell; float: left; }
div.bg01 { clear: both; background-image:
url(http://assets.bunchball.com/scripts/media/notifications/popup01/i
mages/bg01.png) !important; background-repeat: repeat-y; background-
image: none; filter: none !important; filter:
progid:DXImageTransform.Microsoft.AlphaImageLoader(src='http://assets
.bunchball.com/scripts/media/notifications/popup01/images/bg01.png',
sizingMethod='scale'); display: table; width: 329px; } div.image {
float: left; margin-left: 29px; position: relative; background-image:
url(http://assets.bunchball.com/scripts/media/notifications/popup01/i
mages/image-bg.jpg); background-repeat: no-repeat; width: 122px;
height: 102px; } div.image img { position: absolute; top: 2px; left:
13px; } div.imagetxt { float: left; width: 153px; background-image:
url(http://assets.bunchball.com/scripts/media/notifications/popup01/i
mages/imagetxt-bg.jpg); background-repeat: no-repeat; background-
position: top right; } div.imagetxt h1 { color: white; font-size:
13px; font-weight: bold; padding: 0 0 5px 0; margin: 0; width: 135px;
line-height: 19px; font-family: \"Trebuchet MS\"; } div.imagetxt p {
color: white; font-size: 11px; font-weight: bold; width: 135px; line-
height: 16px; padding: 0; margin: 0; font-family: \"Trebuchet MS\"; }
div.bg02 { clear: both; background-image:
url(http://assets.bunchball.com/scripts/media/notifications/popup01/i
mages/bg02.png) !important; background-repeat: repeat-y; background-
image: none; filter: none !important; filter:
progid:DXImageTransform.Microsoft.AlphaImageLoader(src='http://assets
.bunchball.com/scripts/media/notifications/popup01/images/bg02.png',
sizingMethod='scale'); display: table; width: 329px; } div.bg02 p {
```

```
width: 252px; margin-left: 44px; padding: 10px 0; color: #c0c0c0;
font-family: \"Trebuchet MS\"; font-size: 11px; }\", \"description\":
\"Simple, elegant, refined, timeless. Features a picture, title,
subtitle, and message area.\", \"thumbUrl\":
\"http://assets.bunchball.com/scripts/media/notifications/popup01/scre
enshot.png\", \"isDefault\": false, \"order\": 0, \"duration\": \"5\",
\"delay\": \"0\", \"frequency\": \"Unlimited\", \"corner\": \"Bottom Left\",
\"direction\": \"Vertical\" } ] }, \"returnCount\": 1, \"startIndex\": 0,
\"totalRecords\": 2 }, \"responseType\": {}, \"isError\": false,
\"httpError\": 0, \"httpsStatusCode\": 200 }
```

admin.getSiteLevels Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Gets all the [levels](#) defined for the site. Each site level tag contains a type attribute which can be either custom or points. A level of type "custom" has an additional order attribute.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|--------------------------------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| type | No | String | Level type specified in Nitro Studio | 0 - 255 chars | The type of level to get (Points or Custom). Defaults to the site level type selected in Nitro Studio (Configuration > Site Settings > Levels > Leveling Type). |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|---------------------|--|
| actionPhrase | The action phrase that displays in the newsfeed when a user reaches the level. |
| description | The level description. |

| Response | Description |
|-----------------------|---|
| iconUrl | The URL to the image associated with the level. |
| id | The level's ID number. |
| name | The level name. |
| notificationId | The ID of the notification associated with the level. |
| points | The number of points needed to reach the level. |
| type | The level type (points or custom). |


Sample Request

method=admin.getSiteLevels&type=custom

Sample Response

```
{ "response": { "siteLevels": { "siteLevel": [ { "id": 744465,
"clientId": "150401|-946|16358593102805", "order": 4, "name":
"Basic", "description": "Level 1 - No training", "type": "custom",
"actionPhrase": "" }, { "id": 744466, "clientId":
"150401|469|16409099144932", "order": 5, "name": "Intermediate",
"description": "Level 2 - 10 or fewer training sessions", "type":
"custom", "actionPhrase": "" } ] } }, "responseType": {}, "isError":
false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.getSiteLocales Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Gets a list of all locales and their enabled/disabled status.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |

| Name | Required | Type* | Default Value | Range | Comments |
|------------|----------|--------|---------------|---------------|--|
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------|--|
| enabled | Indicates if the locale is enabled on the site. Returns true or false. |
| font | The default font for the locale. |
| name | The language code for the locale. |


Sample Request

```
method=admin.getSiteLocales
```

Sample Response

```
{ "response": { "locales": { "locales": [ { "name": "en-US", "font": "Arial", "enabled": "false" }, { "name": "da-DK", "font": "Arial", "enabled": "false" }, { "name": "de-AT", "font": "Arial", "enabled": "false" }, { "name": "es-LAM", "font": "Arial", "enabled": "false" } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.getSitePointCategories Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Gets one or all point categories.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|------------|----------|--------|---------------|---------------|--|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the |

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|---------|---------------|------------------------|--|
| | | | | | response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| id | No | Integer | 0 | 0 - 2 ³¹ -1 | The point category ID to query for. |
| name | No | String | None | 1 - 255 chars | The point category name to query for. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--------------------------------------|---|
| dailyMaxCredits | The maximum number of points a user can earn in this point category in a single day. |
| dailyMaxCreditsNotificationId | The ID of the notification that displays when the user reaches the daily max points limit. |
| expirationFrequency | How often the user's balance in this point category is reset to 0. Returns Never, Monthly, or Yearly. |
| iconUrl | The URL to the icon that displays for the point category in widgets where you show the user's level. |
| id | The point category's ID. |
| initialPoints | The number of points each new user starts with. |
| isDefault | Indicates if this is the default point category. Returns true or false. |
| maxBalance | The lifetime maximum balance a user can have in this point category. |
| maxBalanceNotificationId | The URL to the notification that displays when the user reaches the maximum points balance for the first time. |
| name | The point category name. |
| premium | Indicates if this is a premium points category: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| shortName | The abbreviated category name. |

| Response | Description |
|---------------------------------------|--|
| yearlyMaxCredits | The maximum number of points a user can earn in this point category in a single year. |
| yearlyMaxCreditsNotificationId | The URL to the notification that displays when the user reaches the yearly max points limit. |


Sample Request

```
method=admin.getSitePointCategories
```

Sample Response

```
{ "response": { "pointCategories": { "pointCategory": [ { "id": 295686, "clientId": "150401|767|5091766271837", "name": "Points", "premium": 0, "iconUrl": "https://amazonaws.com/bunchball_images/7283d24f78eb/userUploadpng15752.png", "shortName": "Points", "maxBalance": 0, "initialPoints": 0, "dailyMaxCredits": 0, "dailyMaxCreditsNotificationId": 87, "expirationFrequency": "never", "isDefault": true, "yearlyMaxCredits": 0, } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpStatusCode": 200 }
```

admin.login and admin.loginAdmin Methods

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Note: The admin.login and admin.loginAdmin methods are identical.

Logs in an admin user and returns an admin session key that can be used in future admin requests. Session keys are valid for 30 minutes. The time period is not configurable.

admin.loginAdmin requires a Nitro Studio Administrator user ID and password. Administrator passwords typically expire every 90 days, and can be reset via the “Forgot your password?” link on the Nitro Studio login page. Consider implementing a simple way to update the password used in your code that does not require a code change or deployment.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| apiKey | YES | String | None | 1 - 255 chars | The site's API key. |
| userId | YES | String | None | 1 - 255 chars | The admin user's user name. This is typically the administrator's email address. |
| password | YES | String | None | 0 - 255 chars | The admin user's password. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|---------------------|---|
| sessionKey | The sessionKey value. |
| bucketed | For internal use. |
| siteName | The Bunchball site name. |
| isSuperAdmin | Indicates if the admin has super admin privileges. Returns true or false. |
| version | For internal use and backward compatibility. |


Sample Request

method=admin.login&userId=myname&password=mypassword

Sample Response

```
{ "response": { "login": { "sessionKey": "NXwxNTAF8Mzg1NMg==",
"type": 2, "isSuperAdmin": true, "siteName": "Sandbox", "mode":
"nitro", "version": "5.2", "bucketed": false } }, "responseType": {},
"isError": false, "httpError": 0, "httpsStatusCode": 200 }
```


admin.logoutAdmin Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Logs out the admin user.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from user.login . |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------|--|
| logout | Indicates if the admin was successfully logged out. Returns true or false. |
| isError | Indicates if the call was successful. Returns true or false. |
| server | The Nitro server name. |


Sample Request

```
method=admin.logoutAdmin
```

Sample Response

```
{ "response": { "logout": true, }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.removeChallengeFromUser Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Removes the challenge completion, but not points associated with the challenge from a user. Use [user.debitPoints](#) to remove points from a user.

You cannot remove a repeatable challenge.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| challengeId | YES | Long | None | N/A | The Nitro-generated unique ID for the challenge. Use admin.getChallenges to determine the challenge ID. |
| userId | YES | String | None | 0 - 255 chars | The ID of the user to remove the challenge from. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------|--------------------------------|
| ok | The challenge was removed. |
| err | The challenge was not removed. |

Sample Request

```
admin.removeChallengeFromUser&userId=1000&challengeId=4802529
```




```
'postman-token: 6bbbb66b-b6bb-b6b6-b6b6-b666b6bb666b' \ -d
'type=0&data=40620'
```

Example Restore by XML file

```
curl -X POST \ http://localhost:3003/restore \ -H 'authorization:
Bearer 6ce780cdde3a91c646d293583b74b9b499ac8721' \ -H 'cache-control:
no-cache' \ -H 'content-type: application/x-www-form-urlencoded' \ -H
'postman-token: ef5611b2-d6f2-0219-b0e0-205306ccf2f7' \ -d
'type=1&data=%3C%3Fxml%20version%3D%221.0%22%20encoding%3D%22UTF-
8%22%20standalone%3D%22yes%22%3F%3E%20%3CNitro%20method%3D%22admin.ba
ckupSiteConfig%22%20server%3D%22bc8c01d15a88%2Fnitro%22%20res%3D%22ok
%22%3E%20%20%20%20%20%3CSiteConfiguration%20name%3D%22Test%20Site%20u
sed%20by%20unit%20test%20script.%20No%20limits!%22%20userlinkUrl%3D%2
2http%3A%2F%2Ftest.bunchball.com%2FuserId%22%20achievementCallbackUrl
%3D%22%22%20levelType%3D%221%22%20whitelistDisabled%3D%221%22%20authe
nticationDisabled%3D%221%22%20userFilterPrefix%3D%22%22%20catalogBuyB
ackPercent%3D%220%22%20ownedItemFeedDescription%3D%22%22%20giftItemFe
edDescription%3D%22%22%20giftNotifyUrl%3D%22test%22%20itemPurchaseNot
ifyUrl%3D%22http%3A%2F%2Fnicknorrell.com%2Ftemp%2F%22%20subdomain%3D%
22test.nitro.bunchball.net%22%20enforceSubdomain%3D%220%22%20version%
3D%221.1%22%2...
```

admin.setPointsBalance Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Sets a user's points balance.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |

| Name | Required | Type* | Default Value | Range | Comments |
|-----------------------|----------|---------|---------------|------------------------|---|
| desiredBalance | YES | Integer | 0 | 0 - 2 ³¹ -1 | The points balance to set for the user. Positive numbers only. |
| userId | YES | String | None | 1 - 255 chars | The ID of the user to set the points balance for. |
| pointCategory | No | String | None | 1 - 255 chars | The point category to set the balance for. If not specified, assigns the point balance to the default point category. |
| pointsId | YES | Integer | 0 | 0 - 2 ³¹ -1 | The user's point category points ID. user.getPointsBalance returns the points ID (if the user has points in that point category). |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------------|---|
| iconUrl | The URL to the icon that displays for the point category in widgets where you show the user's level. |
| id | The point category's ID. |
| isDefault | Indicates if this is the default point category. Returns true or false. |
| lifetimeBalance | The number of lifetime points earned by the user. |
| name | The point category name. |
| points | The user's current points balance in the selected point category. |
| pointsId | The user's point category points ID to query for. |
| premium | Indicates if this is a premium points category: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| shortName | The abbreviated category name. |
| userId | The user's ID. |


Sample Request

```
method=admin.setPointsBalance&desiredBalance=5000&userId=judi.ingram&pointsId=6320
```

Sample Response

```
{ "response": { "balance": { "pointCategories": { "pointCategory": [
{ "pointsId": 6320, "id": 1, "name": "Points", "premium": 0,
"points": 5000, "lifetimeBalance": 6220, "iconUrl": "https://s3-us-
west-2.amazonaws.com/bunchball-public/point_category_icon-
1?15235271", "shortName": "Pts", "isDefault": true } ] }, "userId":
"judi.ingram", "points": 5000, "lifetimeBalance": 6220 } },
"responseType": {}, "isError": false, "httpError": 0,
"httpStatusCode": 200 }
```

admin.updateChallenge Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Updates an existing challenge. Challenge rules are updated separately using the [admin.updateRule](#) method.

Arguments

| Name | Require d | Type* | Defaul t Value | Range | Comments |
|---------------------|-----------|---------|----------------|------------------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| actionPhrase | No | String | None | 0 - 255 chars | The message that displays in the newsfeed when the challenge is completed. |
| actionUrl | No | String | None | 0 -255 chars | The URL that displays in the newsfeed when the challenge is completed. |
| activeFlag | No | Integer | None | 0 - 2 ³¹ -1 | Indicates if the challenge is active (visible) or hidden (not visible) to the user: <ul style="list-style-type: none"> ▪ 0 - Not active |

| Name | Required | Type* | Default Value | Range | Comments |
|------------------------------|----------|---------|---------------|------------------------|---|
| | | | | | <ul style="list-style-type: none"> 1 - Active |
| applyMultiplier | No | Integer | None | 0 - 2 ³¹ -1 | If set, the point award is multiplied by the value specified in the log action. |
| catalogItemId | No | Long | None | N/A | If set, rewards the user with the specified catalog item for completing the challenge. |
| challengeId | YES | Long | None | N/A | The Nitro-generated unique ID for the challenge. Use admin.getChallenges to determine the challenge ID. |
| customData | No | String | None | 0 - 10000 chars | The metadata to associate with the challenge. |
| dailyAchievementLimit | No | Integer | None | 0 - 2 ³¹ -1 | The maximum number of times the challenge can be completed in one day. |
| dependentChallengeId | No | Long | None | N/A | The ID of the prerequisite challenge, if one exists. |
| description | No | String | None | 0 - 2048 chars | The challenge description. |
| end | No | Integer | 0 | 0 - 2 ³¹ -1 | UNIX timestamp. Sets the date the challenge ends. All times are Eastern Standard Time. |
| folderId | No | Long | None | N/A | The folder the challenge is located in. |
| fullUrl | No | String | None | 0 - 255 chars | The URL for the trophy or badge associated with the challenge. |
| groupChallenge | No | Boolean | None | true/false | Indicates if the challenge is a group challenge. |
| hideBeforeEarned | No | Integer | None | 0 - 2 ³¹ -1 | Indicates if the challenge should be hidden until the user completes it: <ul style="list-style-type: none"> 0 - Not hidden 1 - Hidden |
| iconUrl | No | String | None | 0 -255 chars | The URL for the icon of the trophy or badge associated with the challenge. |
| levelId | No | Long | None | N/A | For internal use only. |

| Name | Required | Type* | Default Value | Range | Comments |
|-----------------------------|----------|----------------------|---------------|--|---|
| limit | No | Integer | None | 1 - 2 ³¹ -1 | The maximum number of points to be awarded or the number of times the challenge can be completed. |
| limitType | No | String | None | 0 - 255 chars | The limit type. Choices are 'points' and 'challenges' depending on which type of limit was specified. |
| name | No | String | None | 0 - 255 | The challenge name. |
| newApi | No | String | None | 0 - 255 chars Truncated | Indicates if the request is using the rule API. |
| | | | | | |
| order | No | Integer | None | 0 - 2 ³¹ -1 | Determines the order the challenge should display in the trophy case. |
| pointAward | No | Integer | None | 1 - 2 ³¹ -1 | The number of points a user should receive for completing the challenge. |
| | | | | | |
| preferences | No | Comma-separated list | None | 1 - 3 items Truncated | A comma-separated list of preferences for the challenge. |
| preferencesMatchType | No | String | or | 0 - Unlimited chars | Indicates how preferences associated with the challenge must be matched to complete it. Accepts and, all, or any. |
| preferenceValues | No | Comma-separated list | None | 0 - 3 items Truncated | A comma-separated list of the preferences' values. The number of values should match the number of preferences, otherwise values are truncated. |
| prereqMatchType | No | String | None | 0 - Unlimited chars | Indicates how prerequisites associated with the challenge must be matched to complete it. Accepts and, all, or any. |
| repeatable | No | Integer | None | 0 - 2 ³¹ -1 | Indicates if the challenge is repeatable: |

| Name | Required | Type* | Default Value | Range | Comments |
|----------------------|----------|----------------------|---------------|---|---|
| | | | | | <ul style="list-style-type: none"> ▪ 0 - Not repeatable ▪ 1 - Repeatable |
| rewards | No | Comma-separated list | None | 0 - 10 items Truncated | A list of points, notifications, and catalog items that are awarded to a user when they complete the challenge. |
| ruleMatchType | No | Integer | None | 0 - 2 ³¹ -1 | Indicates how rules associated with the challenge must be matched to complete it. Options are: <ul style="list-style-type: none"> ▪ 0 - All ▪ 1 - Any ▪ 2 - Order listed |
| start | No | Integer | 0 | 0 - 2 ³¹ -1 | UNIX timestamp. Sets the start date after which the challenge can be completed. All times are Eastern Standard Time. |
| tags | No | Comma-separated list | None | 0 - 10 items | <p>Used to pass the action name. A comma-separated list of actions and contextual tags to log. Mainly used to power leaderboards.</p> <p>The parameter is case-sensitive and includes both the action and contextual tag. For example, <code>method=logAction&tags=action_tag,contextual1,contextual2,contextual3</code>.</p> <p>The parameter supports up to a total of 10 items (10 contextual tags, or 10 action tags, or 1 action tag and 9 contextual tags). However, leaderboards and reporting can only support up to 5 contextual tags.</p> <p>Note: Bunchball discourages logging actions with multiple comma-delimited tags and encourages using metadata and preferences.</p> |

| Name | Required | Type* | Default Value | Range | Comments |
|----------|----------|--------|---------------|---------------|---|
| thumbUrl | No | String | None | 0 - 255 chars | The URL for the thumbnail of the trophy or badge associated with the challenge. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|-----------------------|---|
| activeFlag | Indicates if the challenge is active: <ul style="list-style-type: none"> 0 - Not active 1 - Active |
| applyMultiplier | The points multiplier in use for the challenge. |
| callbackFlag | Indicates if the challenge is enabled to trigger an achievement callback when completed: <ul style="list-style-type: none"> 0 - No 1 - Yes |
| ChallengeRewards | The rewards associated with the challenge: <ul style="list-style-type: none"> details - The reward ID itemName - The reward name type - The reward type |
| dailyAchievementLimit | The number of times the challenge can be completed in a day. Returns 0 for unlimited. |
| dateIssued | The date the challenge was updated (UNIX timestamp). |
| description | The challenge's description. |
| endTime | The date/time the challenge ends (UNIX timestamp). |
| featuredFlag | Indicates if the challenge is featured within the Missions widget: <ul style="list-style-type: none"> 0 - Not featured 1 - Featured |
| fullUrl | The URL to the full size image associated with the challenge. |
| groupFlag | Indicates if the challenge is a group challenge: <ul style="list-style-type: none"> 0 - Not a group challenge 1 - A group challenge |

| Response | Description |
|-----------------------------|--|
| groupPointAward | The number of points awarded to the group for this challenge. |
| hideUntilEarned | Indicates if the challenge is hidden until earned: <ul style="list-style-type: none"> ▪ 0 - Not hidden ▪ 1 - Hidden |
| id | The challenge ID. |
| name | The challenge name. |
| notificationId | The ID of the notification awarded to the user after completing the challenge. |
| order | Indicates if the challenge rules must be completed in order: <ul style="list-style-type: none"> ▪ 0 - Do not have to be completed in order ▪ 1 - Must be completed in order |
| pointAward | The number of points awarded for completing the challenge. |
| pointCategory | For challenges with point rewards, returns the name of the point category the points are awarded to. |
| preferences | Used for metric goals where a segment determines which value to use in the goal. |
| preferencesMatchType | Indicates if the prerequisite is and or or. |
| preferenceValues | The values for the preferences list. |
| prereqMatchType | Indicates if the prerequisite is and or or. |
| repeatable | Indicates if the challenge is repeatable: <ul style="list-style-type: none"> ▪ 0 - Not repeatable ▪ 1 - Repeatable <p>Repeatable, hidden challenges show in both complete and incomplete once initially completed.</p> |
| ruleMatchType | Indicates if the rule operator is and or or. |
| rules | The rules within the challenge: <ul style="list-style-type: none"> ▪ actionTag - The name of the action associated with the rule. ▪ clientId - ▪ custom - ▪ description - The rule's description (user-facing). ▪ displayOrder - The order the rule is shown in the challenge. ▪ goal - The action value goal required by the rule. For Less than scenarios, the action value lists this value as the number configured in Nitro Studio. Must be used with operator for Less than scenarios. |

| Response | Description |
|--------------------|--|
| | <ul style="list-style-type: none"> ▪ id - The rule's ID. ▪ itemName - The action name. ▪ Metadata - The metadata associated with the rule. Returns name and value pair. ▪ operator - The action goal value: <ul style="list-style-type: none"> ▪ GE - Greater than or equal to ▪ GT - Greater than ▪ LT - Less than ▪ LE - Less than or equal to ▪ EQ - Equal to ▪ prereqOperator - Indicates if one or all prerequisites must be met. Returns and or or. ▪ prereqRuleIds - The ID of the rule that must be completed before this rule (for multi-rule challenges). ▪ serviceActionType - Previously used for social actions. ▪ serviceType - Previously used for social actions. ▪ sortOrder - An integer value indicating the rules' order within the challenge. ▪ tagId - ▪ timeRange - ▪ type - Indicates the action goal type: <ul style="list-style-type: none"> ▪ sum - Running total ▪ count - Number of calls |
| serviceType | Previously used for social actions. |
| startTime | The date/time the challenge starts (UNIX timestamp). |
| tags | The tags associated with the challenge. |
| thumbUrl | The URL to the thumbnail size image associated with the challenge. |
| version | For internal use and backward compatibility. |


Sample Request

```
method=admin.updateChallenge&challengeId=4805220&dailyAchievementLimit=1&description=Add a new beneficiary&name=Add a New Beneficiary&notificationId=709121
```

Sample Response

```
{ "response": { "challenges": { "challenge": [ { "rules": { "rule": [
{ "metadata": {}, "id": 6996687, "clientId":
"150401|535|851410495386751", "tagId": 2734243631, "actionTag":
"add_beneficiary", "operator": "GE", "description": "Add a ben",
"goal": 1, "timeRange": 3600, "type": "count", "sortOrder": 0,
"displayOrder": 0, "serviceType": "nitro", "serviceActionType":
"none", "prereqRuleIds": "", "prereqOperator": "or", "custom":
"2734243631", "itemName": "add_beneficiary" } ] },
"challengeRewards": { "challengeReward": [ { "type": "notification",
"details": "709121", "itemName": "Good Job" } ] }, "id": 4805220,
"name": "Add a New Beneficiary", "description": "Add a new
beneficiary", "activeFlag": 1, "featuredFlag": 0, "callbackFlag": 0,
"serviceType": "1", "order": 0, "pointCategory": "Points",
"notificationId": 709121, "thumbUrl":
"https://bunchball_images/7283f78eb/userUploadpng158d72.png",
"fullUrl":
"https://bunchball_images/7283d4f78eb/userUploadpng15bd.png",
"pointAward": 0, "dateIssued": "1483712259", "applyMultiplier": 0,
"repeatable": 0, "hideUntilEarned": 0, "tags": "", "startTime":
1483712259, "endTime": 0, "ruleMatchType": 0, "prereqMatchType":
"or", "preferences": "", "preferenceValues": "",
"preferencesMatchType": "or", "groupFlag": 0, "groupPointAward": 0,
"dailyAchievementLimit": 1, "version": "5.2" } ] } }, "responseType":
{}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.updateMetadatum Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Updates metadata for an existing rule. When you add [metadata](#) name:value pairs to a rule, the metadata logged in the [/users/{userId}/actions](#) API must match the rule metadata exactly for the rule to be completed by a user. Keep in mind that metadata is case sensitive, and as such, the best practice is to use simple strings without special characters and/or numeric IDs as metadata names and values. This minimizes the risk of typos and incorrect formatting from completing rules.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|--------|---------------|------------------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| metadatumId | YES | Long | None | 0 - 2 ³¹ -1 | The unique ID for the rule's metadata. Use admin.getChallenges with includeRules= true to return metadata IDs. |
| name | No | String | None | 0 - 255 chars | The name half of the metadata name:value pair. |
| value | No | String | None | 0 - 255 chars | The value half of the metadata name:value pair. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|------------------------|---|
| activeFlag | Indicates if the challenge is active: <ul style="list-style-type: none"> ▪ 0 - Not active ▪ 1 - Active |
| applyMultiplier | The points multiplier in use for the challenge. |
| callbackFlag | Indicates if the challenge is enabled to trigger an achievement callback when completed: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| ChallengeReward | The rewards associated with the challenge: <ul style="list-style-type: none"> ▪ type - Indicates the reward type. Returns point, item, or notification. ▪ details - Additional reward information. For points, the details may be something like "285424 50" with the number of points awarded after the pipe (). In this example, 50 points are awarded to the user after completing the challenge. ▪ itemName - The name of the point category, catalog item, or notification. |

| Response | Description |
|------------------------------|--|
| custom | The challenge's custom data. |
| dailyAchievementLimit | The number of times the challenge can be completed in a day. Returns 0 for unlimited. |
| dateIssued | The date the challenge becomes active (UNIX timestamp). |
| description | The challenge's description. |
| endTime | The date the challenge ends (UNIX timestamp). Returns 0 if no end date is set. Typically used to display the expiration date of the challenge to the user. |
| featuredFlag | Indicates if the challenge is featured within the Missions widget: <ul style="list-style-type: none"> ▪ 0 - Not featured ▪ 1 - Featured |
| folderId | The ID of the folder the challenge is located in. |
| fullUrl | URL to the full size image associated with the challenge. |
| groupFlag | Indicates if it's a group challenge: <ul style="list-style-type: none"> ▪ 0 - Not a group challenge ▪ 1 - A group challenge |
| groupPointAward | The number of points awarded to the group for completing the challenge. |
| hideUntilEarned | Indicates if the challenge is hidden until complete: <ul style="list-style-type: none"> ▪ 0 - Not hidden ▪ 1 - Hidden <p>Repeatable, hidden challenges show in both complete and incomplete once initially completed.</p> |
| id | The rule ID. |
| metadata | The metadata associated with the rule: <ul style="list-style-type: none"> ▪ id - The ID of the rule. ▪ name - The name half of the metadata name:value pair. ▪ value - The value half of the metadata name:value pair. |
| name | The challenge name. |
| order | Indicates if the challenge rules must be completed in order: <ul style="list-style-type: none"> ▪ 0 - Do not have to be completed in order. ▪ 1 - Must be completed in order. |
| pointAward | The number of points awarded for completing the challenge. |

| Response | Description |
|-----------------------------|--|
| pointCategory | For challenges with point rewards, returns the name of the point category the points are awarded to. |
| pointCategoryId | The ID of the point category. |
| preferences | Used for metric goals where a segment determines which value to use in the goal. |
| preferencesMatchType | Indicates if the preference operator is "and" or "or". |
| preferenceValues | The values for the preferences list. |
| prereqMatchType | Indicates if the prerequisite operator is "and" or "or". |
| repeatable | <p>Indicates if the challenge is repeatable:</p> <ul style="list-style-type: none"> ▪ 0 - Not repeatable ▪ 1 - Repeatable <p>Repeatable, hidden challenges show in both complete and incomplete once initially completed.</p> |
| rules | <p>The rules associated with the challenge:</p> <ul style="list-style-type: none"> ▪ actionTag - The name of the action associated with the rule. ▪ displayOrder - The rule's order within the column. ▪ description - The rule's action description. ▪ goal - The action value goal required by the rule. For Less than scenarios, the action value lists this value as the number configured in Nitro Studio. Must be used with operator for Less than scenarios. ▪ id - The rule ID. ▪ itemName - The action name. ▪ operator - The action goal value: <ul style="list-style-type: none"> ▪ GE - Greater than or equal to ▪ GT - Greater than ▪ LT - Less than ▪ LE - Less than or equal to ▪ EQ - Equal to ▪ prereqOperator - Indicates if one or all prerequisites must be met. Returns "and" or "or". ▪ prereqRuleIds - The ID of the rule that must be completed before this rule (for multi-rule challenges). ▪ serviceActionType - Previously used for social actions. ▪ serviceType - Previously used for social actions. |

| Response | Description |
|----------------------|--|
| | <ul style="list-style-type: none"> ▪ sortOrder - An integer value indicating the rules' order within the challenge. ▪ type - Indicates the action goal type: <ul style="list-style-type: none"> ▪ sum - Running total ▪ count - Number of calls |
| ruleMatchType | Indicates if the rule operator is "and" or "or". |
| serviceType | Previously used for social actions. |
| startTime | The challenge start date (UNIX timestamp). |
| tags | The challenge's tags. |
| thumbUrl | URL to the thumbnail image associated with the challenge. |
| version | For internal use and backward compatibility. |

Sample Request


method=admin.updateMetadatum&metadatumId=533236&name=quiz_id&value=72

Sample Response

```
{ "response": { "challenges": { "challenge": [ { "rules": { "rule": [
{ "metadata": { "metadatum": [ { "id": 533236, "derived": false,
"name": "quiz_id", "value": "72" } ] }, "id": "7031283", "tagId":
"2734269999", "actionTag": "QUIZ_COMPLETED", "operator": "GE",
"description": "Take the meaning of life quiz", "goal": "1",
"timeRange": "3600", "type": "count", "sortOrder": "0",
"displayOrder": "0", "serviceType": "nitro", "serviceActionType":
"none", "prereqRuleIds": "", "prereqOperator": "or", "custom":
"2734269999", "itemName": "QUIZ_COMPLETED" }, { "Metadata": {}, "id":
"7031284", "operator": "", "goal": "0", "type": "operator",
"sortOrder": "0", "displayOrder": "0", "prereqRuleIds": "7031283",
"prereqOperator": "or" } ] }, "challengeRewards": {
"challengeReward": [ { "type": "point", "details": "295169|11",
"itemName": "Points" } ] }, "id": "4820279", "name": "Know the
meaning of life", "description": "Know the meaning of life (by taking
this quiz)", "activeFlag": "1", "featuredFlag": "0", "callbackFlag":
"0", "serviceType": "1", "order": "0", "pointCategory": "Points",
"thumbUrl":
"https://amazonsaws.com/images/7283d78eb/userUploadpng15e9f.png",
"fullUrl":
"https://amazonsaws.com/images/728330ac4/userUploadpng15e96.png",
"pointAward": "11", "dateIssued": "1505768646", "applyMultiplier":
"0", "repeatable": "0", "hideUntilEarned": "0", "pointCategoryId":
```

```
"295169", "tags": "", "startTime": "1505768646", "endTime": "0",
"ruleMatchType": "0", "prereqMatchType": "or", "preferences": "",
"preferenceValues": "", "preferencesMatchType": "or", "groupFlag":
"0", "groupPointAward": "0", "dailyAchievementLimit": "0",
"folderId": "575670", "version": "5.2" } ] } }, "responseType": {},
"isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.updateRule Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Updates an existing challenge rule.

Note: Parameters not specified during an update result in those parameters being set to blank/empty values.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------|----------|---------|---------------|------------------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| description | No | String | None | 0 - 255 chars | The rule description. |
| folderId | No | Long | None | N/A | The ID of the folder the rule should be associated with. |
| goal | No | Integer | None | 0 - 2 ³¹ -1 | The numerical value for how much or how often the action must be logged. |
| isNegated | No | Boolean | false | true/false | Set to false to preserve backward compatibility. Only group prerequisite rules support a true value to allow a group prerequisite exclusion. |
| operator | No | String | None | 0 - 255 chars | The comparison operator to use to determine if the rule goal has been met: |

| Name | Required | Type* | Default Value | Range | Comments |
|-----------------------|----------|--------|---------------|---------------|---|
| | | | | | <ul style="list-style-type: none"> ▪ GE - Greater than or equal to ▪ GT - Greater than ▪ LT - Less than ▪ LE - Less than or equal to ▪ EQ - Equal to |
| preReqOperator | No | String | None | 0 - 5 chars | Must be either 'and' or 'or'. |
| ruleId | YES | Long | None | N/A | The Nitro generated unique ID for the challenge rule. Use admin.getChallenges to get the ruleId. |
| tagId | No | Long | None | N/A | The ID of the tag the rule should be associated with. |
| type | No | String | None | 0 - 255 chars | The action goal for this rule (what to measure): <ul style="list-style-type: none"> ▪ count - The number of times a user must perform the action. ▪ sum - The total value a user must accumulate. ▪ timeRange - The number of times a user must perform the action within the specified time frame. ▪ value - The value for the action that a user must achieve. ▪ withinFolder - Users complete a number of actions within a folder. |
| url | No | String | None | 0 - 255 chars | The URL where users begin to complete this rule. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|---------------------|--|
| actionTag | The name of the action associated with the rule. |
| description | The rule's description (user-facing). |
| displayOrder | The order the rule is shown in the challenge. |

| Response | Description |
|--------------------------|---|
| goal | The action value goal required by the rule. For Less than scenarios, the action value lists this value as the number configured in Nitro Studio. Must be used with operator for Less than scenarios. |
| id | The rule's ID. |
| itemName | The action name. |
| Metadata | The metadata associated with the rule. Returns name and value pair. |
| operator | The action goal value: <ul style="list-style-type: none"> ▪ GE - Greater than or equal to ▪ GT - Greater than ▪ LT - Less than ▪ LE - Less than or equal to ▪ EQ - Equal to |
| prereqOperator | Indicates if one or all prerequisites must be met. Returns and or or. |
| prereqRuleIds | The ID of the rule that must be completed before this rule (for multi-rule challenges). |
| serviceActionType | Previously used for social actions. |
| serviceType | Previously used for social actions. |
| sortOrder | An integer value indicating the rules' order within the challenge. |
| type | Indicates the rule type: <ul style="list-style-type: none"> ▪ count - Number of calls ▪ sum - Running total |

Sample Request


method=admin.updateRule&ruleId=6999480&description=Research motorcoach&operator=GE&goal=2

Sample Response

```
{ "response": { "rule": { "metadata": {}, "id": 6999480, "clientId": "150401|-126|1458682661823470", "tagId": 2734215237, "actionTag": "learn_more", "operator": "GE", "description": "Research motorcoach", "goal": 2, "timeRange": 3600, "type": "count", "sortOrder": 0, "displayOrder": 0, "serviceType": "nitro", "serviceActionType": "none", "prereqRuleIds": "", "prereqOperator": "or", "custom":
```

```
"2734215237", "itemName": "learn_more" } }, "responseType": {},
"isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

admin.updateSitePointCategory Method

 As you manage your program, [RESTful APIs](#) should be used before Traditional APIs. Use Traditional APIs only when a REST version is unavailable. Admin APIs are extremely complex and should only be used after consulting [Bunchball Support](#).

Updates a point category. Only returns the updated point category, not all point categories.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|--------------------------------------|----------|---------|---------------|------------------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The sessionKey response value returned from admin.loginAdmin . |
| dailyMaxCredits | No | Integer | 0 | 0 - 2 ³¹ -1 | The maximum number of points a user can earn in this point category in a single day. |
| dailyMaxCreditsNotificationID | No | String | None | 1 - 255 chars | The ID of the notification that displays when the user reaches the daily max points limit. |
| expirationFrequency | No | String | None | 1 - 255 chars | How often the user's balance in this point category is reset to 0. |

| Name | Required | Type* | Default Value | Range | Comments |
|---------------------------------------|----------|---------|---------------|------------------------|--|
| | | | | | Accepts never, monthly, or yearly. |
| iconUrl | No | String | None | 1 - 255 chars | The URL to the icon that displays for the point category. |
| initialPoints | No | Integer | 0 | 0 - 2 ³¹ -1 | The number of points each new user starts with. |
| isDefault | No | Boolean | false | true/false | Indicates if this is the default point category. Accepts true or false. |
| maxBalance | No | Integer | 0 | 0 - 2 ³¹ -1 | The lifetime maximum balance a user can have in this point category. |
| maxBalanceNotificationId | No | String | None | 1 - 255 chars | The URL to the notification that displays when the user reaches the maximum points balance for the first time. |
| name | No | String | None | 1 - 255 chars | The point category name. |
| pointCategoryId | YES | Integer | 0 | 0 - 2 ³¹ -1 | The point category ID. |
| shortName | No | String | None | 1 - 255 chars | The abbreviated category name. |
| yearlyMaxCredits | No | Integer | 0 | 0 - 2 ³¹ -1 | The maximum number of points a user can earn in this point category in a single year. |
| yearlyMaxCreditsNotificationID | No | String | None | 1 - 255 chars | The URL to the notification that displays when the user reaches the yearly max points limit. |

[*Type Descriptions](#)

Response Parameters

| Response | Description |
|---------------------------------------|---|
| dailyMaxCredits | The maximum number of points a user can earn in this point category in a single day. |
| dailyMaxCreditsNotificationId | The ID of the notification that displays when the user reaches the daily max points limit. |
| expirationFrequency | How often the user's balance in this point category is reset to 0. Returns never, monthly, or yearly. |
| iconUrl | The URL to the icon that displays for the point category in widgets where you show the user's level. |
| id | The point category's ID. |
| initialPoints | The number of points each new user starts with. |
| isDefault | Indicates if this is the default point category. Returns true or false. |
| maxBalance | The lifetime maximum balance a user can have in this point category. |
| maxBalanceNotificationId | The URL to the notification that displays when the user reaches the maximum points balance for the first time. |
| name | The point category name. |
| premium | Indicates if this is a premium points category: <ul style="list-style-type: none"> ▪ 0 - No ▪ 1 - Yes |
| shortName | The abbreviated category name. |
| yearlyMaxCredits | The maximum number of points a user can earn in this point category in a single year. |
| yearlyMaxCreditsNotificationId | The URL to the notification that displays when the user reaches the yearly max points limit. |

Sample Request

```
method=admin.updateSitePointCategory&pointCategoryId=591&expirationFrequency=yearly
```

Sample Response

```
{ "response": { "pointCategory": { "id": 591, "clientId":
"1|530|14378783102979", "name": "ServicePoints", "premium": 0,
"iconUrl": "", "shortName": "", "maxBalance": 0, "initialPoints": 0,
"dailyMaxCredits": 0, "expirationFrequency": "never", "isDefault":
false, "yearlyMaxCredits": 0 } }, "responseType": {}, "isError":
false, "httpError": 0, "httpStatusCode": 200 }
```

batch.run Method

This method can be used to call multiple methods on the server with a single call. This helps reduce the round trip time when multiple calls need to be made one after another. If some of the server calls fail, subsequent operations are still performed.

The parameters are the same as in a normal call to the method. The `sessionKey` for each of the method calls is required if no call to `user.login` is made. However, if `user.login` is called first, subsequent calls within `methodFeed` need not specify the `sessionKey`. In such instances, the `sessionKey` from the login is used.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|-------------------|----------|----------------------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |
| sessionKey | YES | String | None | 1 - 255 chars | The <code>sessionKey</code> response value returned from user.login . |
| methodFeed | YES | Comma-separated list | None | 0 - 100 items | A JSON encoded string that specifies the list of methods to call. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|--|---|
| Nitro / content / responseType(n) | Returns one response per method embedded. |

Sample Request

```
method=batch.run&methodFeed=%5B%22method%3Duser.logAction%26userId%3Dtest-user-1%26tags%3Dtest_action%26ts%3D1484862838%26sessionKey%3DNXNzE1YWNOWF8MA%3D%3D%22%2C%22method%3Duser.logAction%26userId%3Dtest-user-1%26tags%3Dbbtest%26ts%3D1484862838%26sessionKey%3DNXNzE1YWNOWF8MA%3D%3D%22%5D
```

Sample Response

```
{ "response": { "Nitro": { "content": [ { "responseType": {}, "isError": false, "response": {}, "httpError": 0, "httpStatusCode": 200 } ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpStatusCode": 200 }
```

hello Method

Checks if the server is responding. For a deeper check, use [server.status](#).

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------|---|
| isError | true - Server is not responding to requests. false - Server is responding to requests. |

Sample Request

```
method=hello
```

Sample Response

```
{ "response": {}, "responseType": {}, "isError": false, "httpError": 0, "httpStatusCode": 200 }
```

server.getErrorCodes Method

Returns a list of [error codes](#) that may be returned by Nitro API calls.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------|------------------------|
| code | The error number. |
| string | The error description. |

Sample Request

```
method=server.getErrorCodes
```

Sample Response

```
{ "response": { "errorCodeList": { "error": [ { "code": 179, "string": "Invalid achievement callback url" }, { "code": 196, "string": "Already exists" }, { "code": 189, "string": "EHCACHE error" }, [...] ] } }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

server.status Method

Checks if the server is running. Verifies the database is available and working, and the distributed cache is functioning.

Bunchball server status can be found at <http://status.bunchball.com>.

Arguments

| Name | Required | Type* | Default Value | Range | Comments |
|------------|----------|--------|---------------|---------------|---|
| asyncToken | No | String | None | 0 - 255 chars | Returns any value passed in the response. Often used to identify which request this is the response for. Typically a single request and response. |

*[Type Descriptions](#)

Response Parameters

| Response | Description |
|----------|----------------------------|
| ok | The server is running. |
| err | The server is not running. |

Sample Request

```
method=server.status
```

Sample Response

```
{ "response": { "method": "server.status", "server": "nitro-sandbox-i-06d5a85b/nitro", "res": "ok" }, "responseType": {}, "isError": false, "httpError": 0, "httpsStatusCode": 200 }
```

Nitro Error Codes

The following list contains the most common Nitro error codes. 'OK' is the only normal response. Any other response indicates an error. The [server.getErrorCodes](#) method returns error codes.

Note: Some API response attributes, such as `httpStatusCode`, return standard HTTP status codes. Those standard codes are not listed here.

If you're receiving timeout errors, check <http://status.bunchball.com> to see if there are any issues with your environment. While most of our APIs return within one second, there are times when load or other variables could cause timeout errors. If you have a short timeout period, increasing that time might resolve the issue.

| Code | Notes | How to Diagnose |
|------|--------------------------|---|
| 101 | Invalid API Key | Open Nitro Studio > Configuration > Site Settings > Overview . Verify the affected method call's API key is exactly the same as the one shown in the API key field. Check for extra characters, such as leading or trailing blanks, in either key. |
| 102 | Invalid Signature | Check the MD5 hash that your server generates when the user.login method is called. When a site is running at Medium or High security, the user.login method requires a signature. If you need to call user.login without a signature, set your site to Low security (Nitro Studio > Configuration > Site Settings > Security). |
| 103 | Transaction Not Verified | |
| 104 | Invalid Timestamp | Verify you are using a 10 digit UNIX timestamp. Common errors include using a 13 digit UNIX timestamp (including milliseconds) and using a date/time string. |
| 105 | Invalid Number of Points | |
| 106 | Limit Exceeded | |
| 107 | User Doesn't Exist | Verify the user whose data you tried to retrieve exists in the system by calling the user.exists method. A new user can be registered by calling the user.login method. |
| 108 | Invalid Category | |
| 109 | Invalid Sub-Category | |
| 110 | Invalid Return Count | |
| 111 | Invalid Start Date | Verify you are using a 10 digit UNIX timestamp. Common errors include using a 13 digit UNIX timestamp (including milliseconds) and using a date/time string. |
| 112 | Invalid End Date | Verify you are using a 10 digit UNIX timestamp. Common errors include using a 13 digit UNIX timestamp (including milliseconds) and using a date/time string. |
| 113 | Invalid Request | |

| Code | Notes | How to Diagnose |
|------|-------------------------------------|---|
| 114 | Session Expired | If your session key expires, generate a new one by calling the user.login method to log in the user. By default, session keys are valid for up to 30 days. To set the duration of your session key, open Nitro Studio > Configuration > Site Settings > Security > Session Timeout. |
| 115 | Module does not exist | |
| 116 | No Levels Defined | Verify levels are defined in Nitro Studio > Configuration > Site Settings > Levels. |
| 118 | Not Authorized | |
| 119 | Record Not Found | |
| 120 | Invalid Parameter | |
| 121 | Invalid Parameter - Number expected | |
| 122 | Action Not Found | Verify the action is defined in Nitro Studio > Integrate > Actions. |
| 123 | Action Tag Not Found | |
| 124 | Invalid tag name | |
| 125 | Invalid action name | |
| 126 | Invalid Criteria | |
| 127 | Invalid Session Key | Make sure the session key you are trying to use is being stored correctly. Try re-generating the session key by calling user.login ; then use the session key returned in subsequent API calls. |
| 128 | Invalid Items | |
| 129 | Invalid Item Id | |
| 130 | Item not available | |
| 131 | Insufficient points balance | |
| 132 | Item already exists | |
| 134 | Invalid recordId | |
| 135 | Invalid gameld | |
| 136 | Invalid userNames | |

| Code | Notes | How to Diagnose |
|------|--|-----------------|
| 137 | Invalid places | |
| 138 | Invalid values | |
| 139 | Invalid orderBy | |
| 140 | Invalid userId | |
| 141 | Client authentication failed from | |
| 142 | Invalid type | |
| 143 | Invalid IP address | |
| 144 | Invalid counter | |
| 145 | Catalog item does not exists in My Item List | |
| 146 | Invalid return type | |
| 147 | Operation does not exists for Compute | |
| 148 | Operator must be one of EQ, GT, GE, LT, LE | |
| 149 | Signature already used | |
| 150 | New UserID already exists | |
| 151 | Invalid tournament request | |
| 152 | Down for Maintenance | |
| 153 | Invalid gift certificate | |
| 154 | Gift certificate expired | |
| 155 | Invalid API Keys | |
| 156 | No method0 defined | |
| 157 | Invalid Release Id | |
| 158 | Release id is not valid for this user | |
| 159 | Invalid Password | |
| 160 | User is not a superuser | |
| 161 | Not a superuser session | |
| 162 | You do not have access to perform this operation | |
| 163 | Not a valid old user id | |
| 164 | Not a valid new user id | |
| 165 | The method called is not recognized | |
| 166 | Missing required parameter | |

| Code | Notes | How to Diagnose |
|------|---|-----------------|
| 167 | Invalid Delta | |
| 168 | Missing allowNaked Parameter | |
| 169 | Invalid Position | |
| 170 | Invalid Reserve Id | |
| 171 | Invalid IP Address or Host | |
| 172 | Cannot add friend before being invited | |
| 173 | Action Value Greater Than Max - | |
| 174 | Action Value Less Than Min - | |
| 175 | Action Rate Exceeded - | |
| 176 | Invalid secretKey | |
| 177 | Invalid old apiKey | |
| 178 | Invalid userlink url | |
| 179 | Invalid achievement callback url | |
| 180 | Invalid tags | |
| 181 | Invalid action tags | |
| 182 | Invalid level number - already used | |
| 183 | Invalid duration: unknown value | |
| 184 | Invalid request - can't set levels when using points-based leveling | |
| 185 | Invalid challenge name | |
| 186 | Unimplemented report | |
| 187 | Database error | |
| 188 | Memcached error | |
| 189 | EHCACHE error | |
| 190 | JMS error | |
| 191 | Server under maintenance | |
| 192 | Rule exists | |
| 193 | Challenge exists | |
| 194 | Name already in use | |
| 195 | Invalid rule configuration | |
| 196 | Already exists | |

| Code | Notes | How to Diagnose |
|------|---|-----------------|
| 197 | Life Time Limit Exceeded | |
| 198 | Invalid number of game types | |
| 199 | Reserved Value | |
| 200 | <ul style="list-style-type: none"> ▪ RESTful APIs: Success ▪ Traditional APIs: Invalid point category | |
| 201 | Invalid catalog item name | |
| 202 | This action can be used only in dev mode | |
| 203 | Source user does not exist, userId | |
| 204 | API Key Expired. Please contact Bunchball for support. | |
| 205 | Cannot buy inactive items. | |
| 206 | No default catalog exists | |
| 207 | At least one tag has to be an Action Tag predefined in the Admin Console. | |
| 208 | No longer used | |
| 209 | A challenge id must be specified when creating a rule | |
| 210 | User is at a level lower than the catalog item level | |
| 211 | Invalid group name | |
| 212 | Cannot clear levels when using points-based levels | |
| 213 | The size of file uploaded does not match the request size. Please try again. | |
| 214 | Cannot specify both friends and a group | |
| 215 | Cannot specify both friends and a group without withRank | |
| 216 | Cannot specify more than 100 user ids | |
| 217 | Cannot specify both and , | |
| 218 | Invalid recipient | |
| 219 | No valid notification names specified | |
| 220 | Another level with same point value already exists | |
| 221 | Cannot credit to a premium point category | |

| Code | Notes | How to Diagnose |
|------|--|-----------------|
| 222 | Size exceeded for parameter | |
| 223 | Invalid Recipient API Key | |
| 224 | Cyclic Dependency found for challenges | |
| 225 | Invalid rule operator specified | |
| 226 | Log Action error | |
| 227 | No tags specified. | |
| 228 | A group challenge must be specified | |
| 229 | Invalid value | |
| 230 | SSL is required for this site | |
| 231 | Invalid competition instance | |
| 232 | At least one userId must be specified | |
| 233 | Invalid Filter | |
| 234 | All available instances of this item are already placed on other catalog instances | |
| 235 | Invalid Color | |
| 236 | That admin username already exists | |
| 237 | All available instances of this item are already placed on this catalog instance | |
| 240 | Premium items cannot be sold back | |
| 241 | One of the specified action tags was not low security | |
| 242 | The token did not exist or was already fulfilled | |
| 243 | Invalid recipient emails | |
| 244 | Invalid email address stored for the user trying to send an email | |
| 245 | An email has already been sent to this user for this action | |
| 246 | User does not exist, userId | |
| 247 | Restore has failed. Your site has not been changed. Please contact Bunchball support | |
| 248 | This locale has already been added to this site | |
| 249 | This locale is not enabled for this site | |

| Code | Notes | How to Diagnose |
|------|--|-----------------|
| 250 | No valid groups were specified | |
| 251 | Invalid url | |
| 252 | Invalid src api key specified | |
| 253 | Invalid version path | |
| 254 | Invalid XML | |
| 255 | Zookeeper initialization error | |
| 256 | bit.ly error | |
| 257 | Invalid verification code | |
| 258 | User already verified | |
| 259 | This user has not verified their account | |
| 260 | Cannot upload the Image, Invalid Image Type | |
| 261 | Elements Payment Error | |
| 262 | There are too many folder levels | |
| 263 | Invalid preferences | |
| 264 | The number of preferences and the number of preference values do not match | |
| 265 | Failed to reset password | |
| 267 | Search error | |
| 268 | Admin user is currently locked out due to too many incorrect password attempts | |
| 269 | Sync Failed: | |
| 270 | Sync is in progress. Modifications are not allowed. | |
| 271 | Tag is not associated with a datasift query. | |
| 272 | Request was throttled | |
| 273 | Cannot specify both userIds and a groupName. | |
| 500 | Server Error | |

Argument Types

The following information details the values expected for each parameter type.

| Type | Notes | Example |
|-----------------------------|--|--|
| String | A string of letters, numbers, and other printable characters. | this is a string |
| Integer | A number with no decimal points. The maximum value for an integer is 2,147,483,647. | 62353 |
| Long | A number with no decimal points. The maximum value for a long is 1.84467441 10 ¹⁹ | 62353 |
| Float | A number with or without decimal points but limited to within its precision range. | 235.23 |
| Boolean | The string "true" or "false". Everything except "true" is interpreted as false. | true |
| Operator | The string "OR" or "AND". | OR |
| Generic List | A string of items separated by pipes or commas, but not both. Pipes have the highest precedence. There can be spaces between and within items. For example the list: "Dark,Knight Batman The Incredible Hulk " is converted to the following list containing 3 items: "Dark,Knight Batman The Incredible Hulk" | item1 item2 item3 item1 , item2 , item3 |
| Comma-separated list | A string of items separated by commas. There can be spaces between and within items. For example the list: "Dark Knight, Batman, The Incredible Hulk ," is converted to the following list containing 3 items: "Dark Knight,Batman,The Incredible Hulk" | item1, item2, item3 |
| Pipe-separated list | A string of items separated by pipes. There can be spaces between and within items. For example the list: "Dark Knight Batman The Incredible Hulk " is converted to the following list containing 3 items: "Dark Knight Batman The Incredible Hulk" | item1 item2 item3 |

Other Parameter Constraints and Flags

| Parameter Flag | Notes |
|------------------|---|
| Truncated | If the input string or list is larger than the maximum allowed range, the input is truncated without returning an error message. Lists are truncated to the max allowed number of whole items and strings are truncated to their maximum length. |

| Parameter Flag | Notes |
|-----------------|---|
| Now | Uses the current UNIX timestamp at the time when the request is received. |
| Required | This item must appear in the request. If it doesn't, the server returns an error message. |

Disclaimer

Bunchball logos, the product, and service names mentioned herein are registered trademarks or trademarks of Bunchball, Inc. All other trademarks are the property of their respective owners.

Every effort has been made to ensure the accuracy of this information. However, Bunchball makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability and fitness for a particular purpose. The information in this document is subject to change without notice.

8/5/2020

Bunchball's Nitro platform is covered by one or more of U.S. Patent 8,768,764 and U.S. Patent 9,779,421.