Nitro Web Components

The features in this guide have been deprecated and replaced with Nitro Impact. While still supported, no updates or fixes are expected for web components. Contact Bunchball Support for assistance.

Contents

Web Components

Bunchball's web components help you personalize your site with rich functionality. Web components are graphical elements which can be pre-configured by administrators in Nitro Studio, and then embedded within websites. To end users, these elements look native to your application. The functional benefits of web components are:

- Plug-in gamification elements
- Easy JavaScript embed code
- Administrative control over many web component behaviors from Nitro Studio
- Flexibility over the web component styling, labeling, and functional configuration
- Bunchball evolves the web component capabilities in line with the overall platform
- Bunchball provides technical support for web components

OAuth Web Components

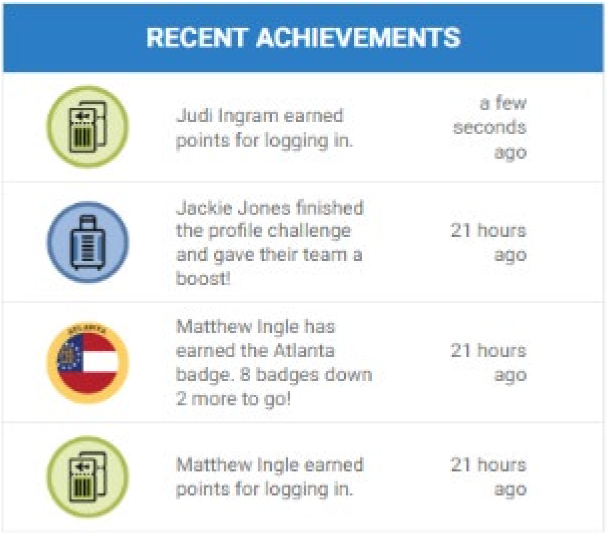
OAuth web components are responsive and adjust to the size of the container they sit within so they are mobile and desktop compatible without additional development. Web components use authentication through OAuth.

Name	Description
Events	Displays the achievements earned by users in the program.
Leaders	Displays a leaderboard of user or group actions, missions, or points.
Missions	Displays the missions a user can complete and includes powerful filters.
OTS	Displays an interface where users can enter a claim code to redeem an On The Spot Card.
Profile	Displays the user's image, name, role, point balance, and level.
Quiz	Displays an interface where users can complete quizzes, surveys, or polls.
Recognitions	Displays an interface where users can recognize others for contributions and accomplishments.
Trophy Case	Displays the user's earned trophies (badges).

Events Web Component

Displays the achievements earned by users in the program. You can show the activity of one user, a group of users, or all users. The Action Phrase field in missions and levels and the Community Phrase field in a recognition category defines what information shows in the web component. Use tokens in the action phrase to replace the token at runtime with the identifier they represent (such as the user's name).

The length of time an item remains in the events list depends on how many items you've set to show and how much activity there is to report. When the information is refreshed, older items remain or are removed depending on the amount of new activity. Adjusting the number of items to show helps you target how long an item remains in the events list for the amount of activity you anticipate.



RECENT ACHIEVEMENTS

Judi Ingram earned points for logging in.	a few seconds ago
Jackie Jones finished the profile challenge and gave their team a boost!	21 hours ago
Matthew Ingle has earned the Atlanta badge. 8 badges down 2 more to go!	21 hours ago
Matthew Ingle earned points for logging in.	21 hours ago

Configure a Web Component

1. Open Nitro Studio > **Integrate** > **Web Components**.
2. Click **New** > **Events**.
3. Enter the web component settings.
   Parameters are used when working directly with a web component embed tag (HTML) or editing the config.json file for Bunchball Go.

Studio Field	Description	Parameters			
		Web Component Embed HTML	Bunchball Go config.json	Valid Values	Required
Name	The web component name. ▪ **web component** - set the component tag as bunchball-events. ▪ **iframe** - must match exactly with the name of the web component created in Nitro Studio.	name	name		Yes
Width	The web component's width.	width	width		Yes
Height	The web component's height.	height	height		Yes
Custom Styles	The link to your custom stylesheet(s). Accepts a comma delimited list of URLs.				
Embed Type	Specifies the type of embed code to generate. ▪ **web component** - Recommended. Web components improve performance due to faster loading. Height and width settings are not required as the web component will fill space on the page like any other HTML element. ▪ **iframe** - The source code is a div tag and the rendered web component will be inside an iframe.	data-embed-type	embed-type	iframe	Yes
Group Class Filter	Filters events by group within the selected group class. For example, user A is in group “Phoenix” and user B is in group “Des Moines”. Both groups are of class “Office”. If the filter is set to “Office”, user A will only see events happening in Phoenix while user B will only see events happening in Des Moines. The selected group class must be a segment.	data-group-class	group-class	the group class name (not ID)	No

Studio Field	Description	Parameters			
		Web Component Embed HTML	Bunchball Go config.json	Valid Values	Required
	Note: When filtering by group, the User ID parameter is also required.				
Event Types Filter	Filters events by type. You can set the filter to show missions completed, level changes, interactions (recognitions), or group missions. Default is to show every event except group missions. Use the pipe separator in your embed code to filter by multiple types. For example, data-type="challenges_completed|interactions" **Note:** When the Community Phrase is blank within an recognition category, that recognition event does not display in the web component.	data-type	type	challenges_completed, level_changes, interactions, group_challenges	No
Number of Events to Display	The number of items to display.	data-return-count	return-count		No
Title	The title to display in the header. Default is Recent Achievements.	data-title	title		No
Subtitle	The subtitle to display below the header.	data-subtitle	subtitle		No
User ID	The ID of the user to get events for. The web component will filter events based on the user ID. If blank, all users' events are returned.	data-user-id	user-id		No
Language	Add a language code to preview the web component in a Nitro supported default language. Default languages are: en, de, hi, zh, it, es, and fr.				No
Custom Language Reference	Add a language code and then add a custom translation key to the browser console to preview the web component in a non-default language.				No
	If the type of the web component is known (for example, EVENTS), the web	data-element-type	element-type		No

		Parameters			
Studio Field	Description	Web Component Embed HTML	Bunchball Go config.json	Valid Values	Required
	component object doesn't need to be fetched using the embed name.				
	Marks the root body element inside the web component with a given class(es) name. This makes it easier to style web component sharing the same custom CSS file with theme differences.	data-theme	theme		No

4. Click **Save & Finish** and reopen the web component.

5. Click **Get Web Component Embed Code**. Copy the code to the clipboard or a file.

6. Include the code to generate an OAuth token in your app.

7. Insert the embed code into the page/div where you want to display the web component in your app or web page and update the variable for the OAuth token.

Sample Embed Code

See Embed an OAuth Web Component for more information and the Web Components view for the most up-to-date embed code. The following is an example embed for an Events web component filtered by group.

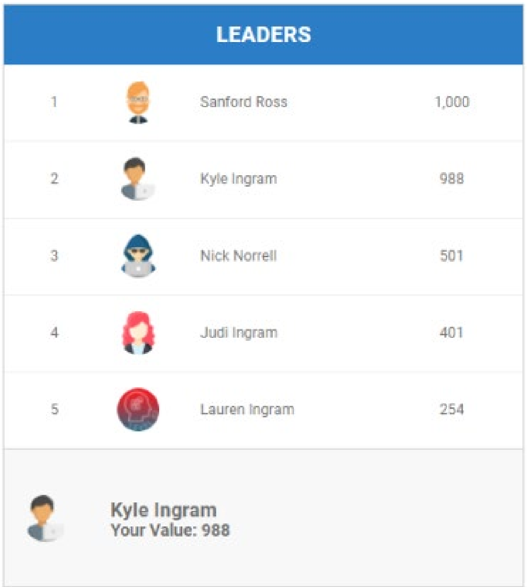
web component	iframe
<bunchball-events style="display:block; width:400px;" data-user-id="AutoTest" data-token="abcdefg1234567" data-export-navigation='false' data-group-class='Team' data-return-count='10' data-title='Recent Achievements' data-custom-css='[""]'> </bunchball-events>	<div name="Events" class="nitro-widget" width="400" height="450" data-name='Events' data-custom-css='[""]' data-embed-type='iframe' data-export-navigation='false' data-height='450' data-return-count='10'

web component	iframe

	data-title='Recent Achievements'
	data-group-class='Team'
	data-width='400'>
	</div>

Leaders Web Component

Displays a leaderboard of user or group actions, missions, or points. You can configure the web component to show one or multiple leaderboards.

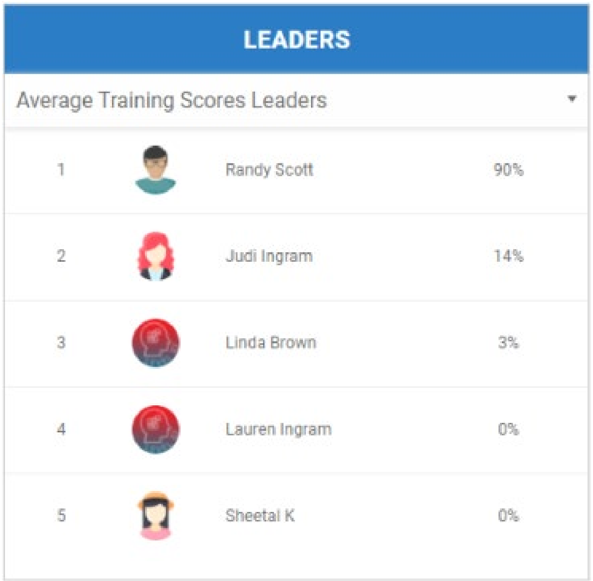


LEADERS

1	Sanford Ross	1,000
2	Kyle Ingram	988
3	Nick Norrell	501
4	Judi Ingram	401
5	Lauren Ingram	254

Kyle Ingram
Your Value: 988

One leaderboard with profile



LEADERS

Average Training Scores Leaders

1	Randy Scott	90%
2	Judi Ingram	14%
3	Linda Brown	3%
4	Lauren Ingram	0%
5	Sheetal K	0%

Multiple leaderboards with no profile

Configure a Web Component

1. Create a leaderboard in the Leaderboards view or using the /leaderboards APIs.
2. (Optional) To show an image in a group or aggregate leaderboard, add a group image.
3. (Optional) To set the leaderboard to display only the first initial of the user's last name, set the displayLeadersLastInitial site preference to true.

4. Open Nitro Studio > **Integrate** > **Web Components**.

5. Click **New** > **Leaders**.

6. Enter the web component settings.
   Parameters are used when working directly with a web component embed tag (HTML) or editing the config.json file for Bunchball Go.

Studio Field	Description	Parameters				
		Web Component Embed HTML	**Bunchball Go config.json**	**Valid Values**	**Required (web component)**	**Required (iframe)**
Name	The web component name. ▪ **web component** - set the component tag as bunchball-leaders. ▪ **iframe** - must match exactly with the name of the web component created in Nitro Studio.	name	name		No	Yes
Width	The web component's width.	width	width		No	Yes
Height	The web component's height.	height	height		No	Yes
Custom Styles	The link to your custom stylesheet(s). Accepts a comma delimited list of URLs.				No	No

Studio Field	**Description**	**Parameters**				
		Web Component Embed HTML	**Bunchball Go config.json**	**Valid Values**	**Required (web component)**	**Required (iframe)**
Embed Type	Specifies the type of embed code to generate. ▪ **web component** - Recommended. Web components improve performance due to faster loading. Height and width settings are not required as the web component will fill space on the page like any other HTML element. ▪ **iframe** - The source code is a div tag and the rendered web component will be inside an iframe.	data-embed-type	embed-type	iframe	No	Yes
Basic Settings						
Add Leaderboard To Web Component	Add one or more leaderboards to the web component. When multiple leaderboards are					

Studio Field	Description	Parameters				
		Web Component Embed HTML	Bunchball Go config.json	Valid Values	Required (web component)	Required (iframe)
	added, a dropdown list displays in the web component so users can toggle between leaderboards.					
Name (Leaderboard)	The name of the leaderboard. When the web component contains multiple leaderboards, the name displays in the dropdown list.				Yes	Yes
Value Format	The format for the values in the leaderboard. Use "d" to represent the number precision and optionally, add a one character prefix or suffix. For example, $d.dd or d%	data-valueFormat	valueFormat		No	No
Show Values with Commas	If true, displays the leaderboard values with commas. For example, 6,500 instead of 6500.	data-showValuesWithCommas	showValuesWithCommas	true, false	No	No
Leaderboard ID	The ID of the leaderboard to show data for. Leaderboards are created using the Leaderboards view	data-leaderboardId	leaderboardId		Yes	Yes

Studio Field	Description	Parameters				
		Web Component Embed HTML	Bunchball Go config.json	Valid Values	Required (web component)	Required (iframe)
	or the /leaderboards API.					
Group Class IDs	Filters the leaderboard by the specified group class ID(s). The selected group class must be a segment. If no ID is specified, all groups are shown. ❗Aggregate user leaderboards and group leaderboards must have only one group class filter. Other user leaderboards can be filtered by multiple group classes. **Note:** The leaderboard displays the points/missions/actions based on the user's current group membership.	data-groupClassId	groupClassId		No	No
Show Profile	If true, shows the current value, not rank, of the logged in user (user leaderboard) or user's group (group	data-showProfile	showProfile	true, false	No	No

Studio Field	Description	Parameters				
		Web Component Embed HTML	Bunchball Go config.json	Valid Values	Required (web component)	Required (iframe)
	leaderboard) at the bottom of the web component. **Note:** In the web component preview, the leaderboard may appear empty if the user you're testing with does not belong to a group within the selected group class(es).					
Criteria	Select how to calculate the leader based on the leaderboard type. For Actions leaderboards: ▪ **count** - Returns leaders with the most number of actions logged. ▪ **sum** - Returns leaders with the highest cumulative action values logged. ▪ **min** - Returns leaders with the lowest action value	data-criteria	criteria	count, sum, min, max, average, lifetimeBalance, creditBalance, debitBalance, completionCount	Yes	Yes

Studio Field	Description	Parameters				
		Web Component Embed HTML	Bunchball Go config.json	Valid Values	Required (web component)	Required (iframe)
	logged. Shows leaders in ascending order. ▪ **max** - Returns leaders with the highest action value logged. ▪ **average** - Returns leaders with the highest average of the action values logged. For Points leaderboards: ▪ **lifetimeBalance** - Returns leaders with the highest balance after calculating point credits plus debits. ▪ **creditBalance** - Returns leaders who have the most point credits. ▪ **debitBalance** - Returns leaders who have the most point debits.					

Studio Field	Description	Parameters				
		Web Component Embed HTML	**Bunchball Go config.json**	**Valid Values**	**Required** **(web component)**	**Required** **(iframe)**
	For Missions leaderboards: ▪ **completionCount** - Returns leaders who have the most number of missions completed.					
User ID	The ID of the user to get leaderboards for.	data-user-id	user-id		Yes	Yes
Title	The title to display in the header. Defaults to Leaders.	data-title	title		No	No
Subtitle	The subtitle to display below the header.	data-subtitle	subtitle		No	No
Number of rows on first display impression	The number of rows to display in the leaderboard. Defaults to 100.	data-page-size	page-size		No	No
Language	Add a language code to preview the web component in a Nitro supported default language. Default languages are: en, de, hi, zh, it, es, and fr.				No	No

Studio Field	Description	Parameters				
		Web Component Embed HTML	Bunchball Go config.json	Valid Values	Required (web component)	Required (iframe)
Custom Language Reference	Add a language code and then add a custom translation key to the browser console to preview the web component in a non-default language.				No	No
	The message that displays when there is no leaderboard data. Defaults to "There are no leaders to show".	data-empty-message	empty-message		No	No
	Marks the root body element inside the web component with a given class(es) name. This makes it easier to style web components sharing the same custom CSS file with theme differences.	data-theme	theme		No	No

7. Click **Save & Finish** and reopen the web component.
8. Click **Get Web Component Embed Code**. Copy the code to the clipboard or a file.
9. Include the code to generate an OAuth token in your app.

10. Insert the embed code into the page/div where you want to display the web component in your app or web page and update the variable for the OAuth token.

Sample Embed Code

See Embed an OAuth Web Component for more information and the Web Components view for the most up-to-date embed code. The following is an example embed for a web component containing two leaderboards.

web component	iframe

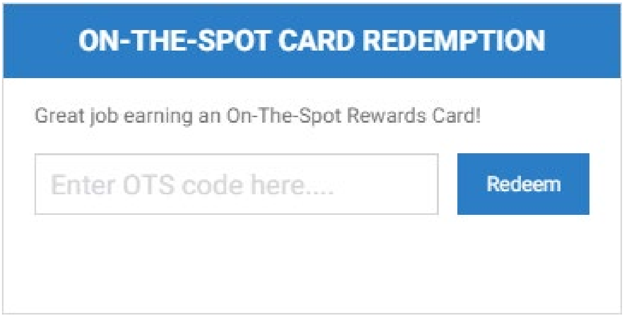
<bunchball-leaders style="display:block; width:400px;"
data-token="abcdefg1234567"
data-user-id="AutoTest"
data-custom-css='[""]'
data-leaderboard-configs='[
{
"name":"Points Leaders",
"showValuesWithCommas":"true",
"leaderboardId":"3",
"groupClassIds":"",
"showProfile":"true",
"criteria":"lifetimeBalance"
},
{
"name":"Quiz Leaders",
"leaderboardId":"1",
"groupClassIds":"",
"showProfile":"true",
"criteria":"count"
}
]'
data-leaderboard-configs-collection='[
{
"name":"Points Leaders",
"showValuesWithCommas":"true",
"leaderboardId":"3",
	<div name="TeamMissions"
	class="nitro-widget"
	width="400"
	height="500"
	data-user-id="AutoTest"
	data-name='TeamMissions'
	data-custom-css='[""]'
	data-embed-type='iframe'
	data-height='500'
	data-leaderboard-configs='[
	{
	"name":"Points Leaders",
	"showValuesWithCommas":"true",
	"leaderboardId":"3",
	"groupClassIds":"",
	"showProfile":"true",
	"criteria":"lifetimeBalance"
	},
	{
	"name":"Quiz Leaders",
	"leaderboardId":"1",
	"groupClassIds":"",
	"showProfile":"true",
	"criteria":"count"
	}
	]'

web component	iframe

"groupClassIds":"",
"showProfile":"true",
"criteria":"lifetimeBalance"
},
{
"name":"Quiz Leaders",
"leaderboardId":"1",
"groupClassIds":"",
"showProfile":"true",
"criteria":"count"
}
]'
data-page-size='10'
data-title='Leaders'
data-custom-css='[""]'>
</bunchball-leaders>
	data-leaderboard-configs-collection='[
	{
	"name":"Points Leaders",
	"showValuesWithCommas":"true",
	"leaderboardId":"3",
	"groupClassIds":"",
	"showProfile":"true",
	"criteria":"lifetimeBalance"
	},
	{
	"name":"Quiz Leaders",
	"leaderboardId":"1",
	"groupClassIds":"",
	"showProfile":"true",
	"criteria":"count"
	}
	]'
	data-page-size='10'
	data-title='Leaders'
	data-width='400'>
	</div>

OTS (On The Spot) Web Component

Displays an interface to your users where they can enter a claim code to redeem an On The Spot Card. When a user claims a card, an action is logged to Nitro and the award amount is applied to the user's Global Rewards Marketplace account for immediate use.



ON-THE-SPOT CARD REDEMPTION

Great job earning an On-The-Spot Rewards Card!

Configure a Web Component

1. Open Nitro Studio > **Integrate** > **Web Components**.
2. Click **New** > **OTS**.
3. Enter the web component settings.
   Parameters are used when working directly with a web component embed tag (HTML) or editing the config.json file for Bunchball Go.

Studio Field	**Description**	**Parameters**				
		Web Component Embed HTML	**Bunchball Go config.json**	**Valid Values**	**Required (web component)**	**Required (iframe)**
Name	The web component name. ▪ **web component** - set the component tag as bunchball-ots. ▪ **iframe** - must match exactly with the name of the web component created in Nitro Studio.	name	name		No	Yes
Width	The web component's width.	width	width		No	Yes
Height	The web component's height.	height	height		No	Yes
Custom Styles	The link to your custom stylesheet(s). Accepts a comma delimited list of URLs.					
Embed Type	Specifies the type of embed code to generate.	data-embed-type	embed-type	iframe	No	Yes

Studio Field	Description	Parameters				
		Web Component Embed HTML	**Bunchball Go config.json**	**Valid Values**	**Required (web component)**	**Required (iframe)**
	▪ **web component** - Recommended. Web components improve performance due to faster loading. Height and width settings are not required as the web component will fill space on the page like any other HTML element. ▪ **iframe** - The source code is a div tag and the rendered web component will be inside an iframe.					
User ID	The ID of the user to apply the award to.	data-user-id	user-id		Yes	Yes
Title	The title to display in the web component's header. Default is On The Spot Card Redemption.	data-title	title		No	No
Subtitle	The subtitle to display below the web component header.	data-subtitle	subtitle		No	No
Welcome Message	The text to display above the redemption field.	data-welcome-message	welcome-message		No	No
Language	Add a language code to preview the web component in a Nitro supported default language. Default languages are: en, de, hi, zh, it, es, and fr.				No	No
Custom Language Reference	Add a language code and then add a custom translation key to the browser console to preview the web component in a non-default language.				No	No

4. Click **Save & Finish** and reopen the web component.

5. Click **Get Web Component Embed Code**. Copy the code to the clipboard or a file.

6. Include the code to generate an OAuth token in your app.

7. Insert the embed code into the page/div where you want to display the web component in your app or web page and update the variable for the OAuth token.

Sample Embed Code

See Embed an OAuth Web Component for more information and the Web Components view for the most up-to-date embed code. The following is an example embed for an OTS web component.

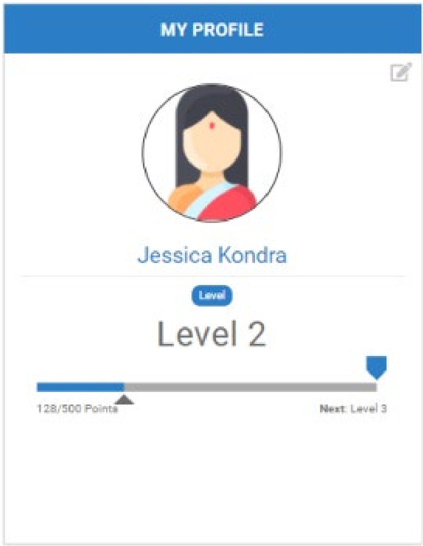
web component	iframe

<bunchball-ots style="display:block; width:400px;"
data-token="abcdefg1234567"
data-user-id="AutoTest"
data-subtitle='Enter your 16 digit code.'
data-title='On-The-Spot Card Redemption'
data-welcome-message='Great job earning an On-The-Spot Rewards Card!'
</bunchball-ots>
	<div name="OTS"
	class="nitro-widget"
	width="400"
	height="200"
	data-name='OTS'
	data-user-id="AutoTest"
	data-embed-type='iframe'
	data-height='200'
	data-subtitle='Enter your 16 digit code.'
	data-title='On-The-Spot Card Redemption'
	data-welcome-message='Great job earning an On-The-Spot Rewards Card!'
	data-width='400'>
	</div>

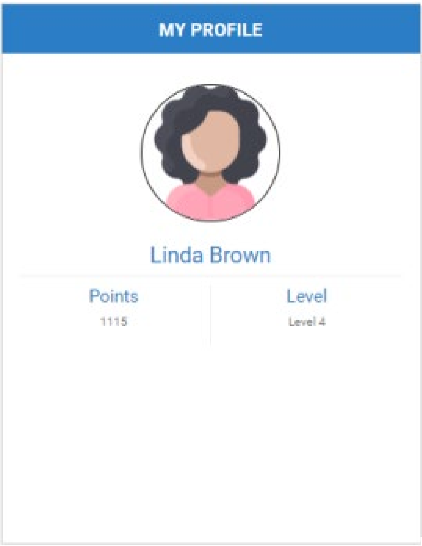
Profile Web Component

Displays the user's name, role, level, and point balance. You can customize the profile by showing or hiding different elements.

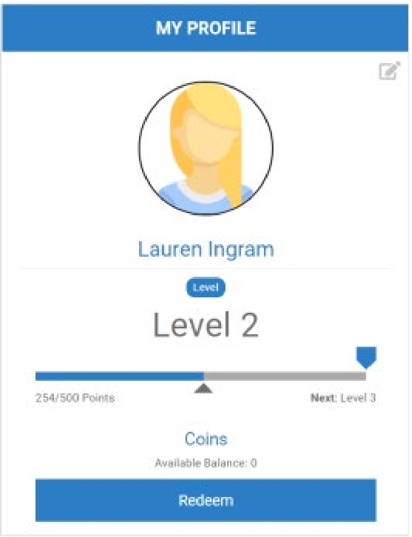
If your program is integrated with the Global Rewards Marketplace, a user's enrollment and verification status determines if they can view their available balance and/or access the Marketplace.



Level progress bar



Points balance and no progress bar



Points balances and access to the Global Rewards Marketplace

Configure a Web Component

1. Open Nitro Studio > **Integrate** > **Web Components**.
2. Click **New** > **Profile**.
3. Enter the web component settings.
   Parameters are used when working directly with a web component embed tag (HTML) or editing the config.json file for Bunchball Go.

Studio Field	**Description**	**Parameters**				
		Web Component Embed HTML	**Bunchball Go config.json**	**Valid Values**	**Required** (web component)	**Required** (iframe)
Name	The web component name. ▪ **web component** - set the component tag as bunchball-profile. ▪ **iframe** - must match exactly with the name of the web component created in Nitro Studio.	name	name		No	Yes
Width	The web component's width.	width	width	relative (%) or absolute value (px)	No	Yes
Height	The web component's height.	height	height	relative (%) or absolute value (px)	No	Yes
Custom Styles	The link to your custom stylesheet(s). Accepts a comma delimited list of URLs.					
Embed Type	Specifies the type of embed code to generate. ▪ **web component** - Recommended. Web components improve performance due to faster loading. Height and width settings are not required as the web component will fill space	data-embed-type	embed-type	iframe	No	Yes

Studio Field	Description	Parameters				
		Web Component Embed HTML	**Bunchball Go config.json**	**Valid Values**	**Required** (web component)	**Required** (iframe)
	on the page like any other HTML element. ▪ **iframe** - The source code is a div tag and the rendered web component will be inside an iframe.					
Web Component						
User ID	The ID of the user to get information for.	data-user-id	user-id		Yes	Yes
Title	The title to display as the web component's header. Defaults to My Profile.	data-title	title		No	No
Subtitle	The subtitle to display below the web component header.	data-subtitle	subtitle		No	No
Point Category	The point category to display.	data-point-category	point-category		No	No
Point Category Title	The title to display for the user's lifetime points balance. Defaults to the name of the selected point category.	data-point-category-title	point-category-title		No	No
Catalog Navigation	Defines if the Global Rewards Marketplace catalog should open inline or in a new tab. Defaults to new tab. Most programs, particularly Bunchball Go, will want to set this to new tab.	data-navigation	navigation	inline, new tab	No	No

Studio Field	Description	Parameters				
		Web Component Embed HTML	Bunchball Go config.json	Valid Values	Required (web component)	Required (iframe)
Balance Display Type	The type of points balance to display in the balance section. ▪ **None** - Shows the user's lifetime points balance for the point category selected in the Point Category field. ▪ **Current Balance** - Shows the user's lifetime and current points balances for the point category selected in the Point Category field. ▪ **Award Balance** - Shows the user's lifetime point balance for the point category selected in the Point Category field, the current points balance for the point category tied to your award account, and a Redeem Points button to access the Global Rewards Marketplace.	data-balance-display-type	balance-display-type	None, Current, Award Balance	No	No
Balance Title	The title to display when showing the user's current points balance. Defaults to Award Account.	data-balance-title	balance-title		No	No

Studio Field	Description	Parameters				
		Web Component Embed HTML	Bunchball Go config.json	Valid Values	Required (web component)	Required (iframe)
Award account exclusion group class	Defines the groups that will not see their award balance or the redemption button. Allows you to hide the award account section for specific groups.	data-award-account-exclusion-group-class-id	award-account-exclusion-group-class-id		No	No
Hide Level Progress Bar	Specifies if the level progress bar section should be hidden. We recommend hiding this section when using custom levels.	data-hide-level-progress	hide-level-progress	true, false	No	No
Max Level Achieved Message	The message to display in place of the next level when the highest level is reached. By default, the 💯 icon displays if no message is set.	data-max-level-achieved-message	max-level-achieved-message		No	No
Unenrolled Message	The message to display when a user is not enrolled in an award account. See Award Account Enrollment and Verification Examples for more details.	data-unenrolled-message	unenrolled-message		No	No
Unverified Message	The message to display when a user is unverified. Only applies to programs with an award account.	data-unverified-message	unverified-message		No	No
Unverified URL	The URL you want to use for the unverified button. This can take the user to the location where they can verify their status or link to a browse only version of the	data-unverified-url	unverified-url		No	No

Studio Field	Description	Parameters				
		Web Component Embed HTML	Bunchball Go config.json	Valid Values	Required (web component)	Required (iframe)
	catalog. Only applies to programs with an award account.					
Editable Items	Allows users to edit their name and/or image from their profile. These edits update the user's preferences (firstName, lastName, and userPhotoUrl). The web component only supports image uploads in png or jpeg format. **Note:** If users remove their profile image, it clears the userPhotoUrl preference.	data-editable-items	editable-items	name, image	No	No
Language	Add a language code to preview the web component in a Nitro supported default language. Default languages are: en, de, hi, zh, it, es, and fr.				No	No
Custom Language Reference	Add a language code and then add a custom translation key to the browser console to preview the web component in a non-default language.				No	No
	Marks the root body element inside the web component with a given class(es) name. This makes it easier to style web components sharing the	data-theme	theme		No	No

Studio Field	Description	Parameters				
		Web Component Embed HTML	**Bunchball Go config.json**	**Valid Values**	**Required** (web component)	**Required** (iframe)
	same custom CSS file with theme differences.					

4. Click **Save & Finish** and reopen the web component.

5. Click **Get Web Component Embed Code**. Copy the code to the clipboard or a file.

6. Include the code to generate an OAuth token in your app.

7. Insert the embed code into the page/div where you want to display the web component in your app or web page, and update the variable for the OAuth token.

Quiz Web Component

Displays an interface to your users where they can take quizzes, surveys, and polls and receive instant feedback on their responses. Results are directly integrated into analytics.

The web component only displays quizzes/surveys/polls that are associated with a mission and meet user eligibility. Unless you set the web component to only display one quiz or to receive a specific quiz ID, the web component presents all quizzes that are available one by one as determined by mission eligibility and ordering.

✓**Best Practice** Implement the Quiz web component as a pop up or place it on a page that is not part of the program navigation. Users should access a quiz through a mission. When accessing quizzes only through missions, the user has context around the quiz, insight into rewards associated with completing it, and doesn't see an empty web component when no quizzes are available.

Answer these questions to see how well you understand the org hierarchy batch load process.

Configure and Run the Quiz Web Component

The following are the high-level steps needed to configure the Quiz web component.

1. Create one or more quizzes.
2. Activate the quiz mission.
3. Create the Quiz web component and extract the embed code.
4. Include the code to generate an OAuth token in your app.
5. Insert the embed code into the page/div where you want to display the web component in your app or web page, and update the variable for the OAuth token.

Create a Quiz Web Component

1. Open Nitro Studio > **Integrate** > **Web Components**.
2. Click **New** > **Quiz**.
3. Enter the web component settings.
   Parameters are used when working directly with a web component embed tag (HTML) or editing the config.json file for Bunchball Go.

Studio Field	**Description**	**Parameters**				
		Web Component Embed HTML	**Bunchball Go config.json**	**Valid Values**	**Required (web component)**	**Required (iframe)**
Name and Display Settings						
Name	The web component name.	name	name		No	Yes

Studio Field	Description	Parameters				
		Web Component Embed HTML	Bunchball Go config.json	Valid Values	Required (web component)	Required (iframe)
	▪ **web component** - set the component tag as bunchball-quiz. ▪ **iframe** - must match exactly with the name of the web component created in Nitro Studio.					
Width	The web component's width.	width	width	relative (%) or absolute value (px)	No	No
Height	The web component's height.	height	height	relative (%) or absolute value (px)	No	No
Custom Styles	The link to your custom stylesheet(s). Accepts a comma delimited list of URLs.					
Embed Type	Specifies the type of embed code to generate. ▪ **web component** - Recommended. Web components improve performance due to faster loading. Height and width settings are not required as the web component will fill space on the page like any other HTML element. ▪ **iframe** - The source code is a div tag and the rendered web component will be inside an iframe.	data-embed-type	embed-type	iframe	No	Yes

Studio Field	Description	Parameters				
		Web Component Embed HTML	Bunchball Go config.json	Valid Values	Required (web component)	Required (iframe)
Web Component Settings						
Subtitle	The subtitle to display below the web component header.	data-subtitle	subtitle		No	No
User ID	The ID of the user to get quizzes for.	data-user-id	user-id		Yes	Yes
Quiz ID	Set a quiz ID to limit the web component to display only one quiz. By default, the web component displays all available quizzes based on user and mission eligibility. This attribute only supports a single value. **Note:** If an ID is set, and the user has already passed the quiz, or is not eligible to see the quiz (i.e. because of segmentation), the message set in the Empty Data Message field will appear.	data-quiz-id	quiz-id		No	No
Empty Data Message	The message that displays when there are no quizzes that the user is eligible for. Defaults to "No quizzes available at the moment...".	data-empty-message	empty-message		No	No
Bypass Resume	Use to set whether or not to bypass the Resume Quiz screen. When set to true, the user will resume the quiz at the point where they previously exited rather than having to click the Resume Quiz button.	data-bypass-resume	bypass-resume	true, false	No	No

Studio Field	**Description**	**Parameters**				
		Web Component Embed HTML	**Bunchball Go config.json**	**Valid Values**	**Required (web component)**	**Required (iframe)**
	Use to customize the confetti that displays when a user passes a quiz. You can add the following parameters to your embed code to replace the default confetti with emoji confetti. These settings will apply to all quizzes that use the web component. ▪ **emoji** - Set one or more emojis to display as confetti. For multiple emojis, use a comma separated list. ▪ **size** - Set the size of the emojis. ▪ **number** - Set the number of emojis to display. Show me an example embed	data-confetti-emojis data-confetti-emoji-size data-confetti-emoji-number	confetti-emojis confetti-emoji-size confetti-emoji-number	Any standard emoji. Size and number accept an integer value.	No	No
Language	Add a language code to preview the web component in a Nitro supported default language. Default languages are: en, de, hi, zh, it, es, and fr.				No	No
Custom Language Reference	Add a language code and then add a custom translation key to the browser console to preview the web component in a non-default language.				No	No
	Marks the root body element inside the web component with a given	data-theme	theme		No	No

Studio Field	Description	Parameters				
		Web Component Embed HTML	Bunchball Go config.json	Valid Values	Required (web component)	Required (iframe)
	class(es) name. This makes it easier to style web components sharing the same custom CSS file with theme differences.					

Note: By default, the web component header displays the name of the active quiz.

4. Click **Save & Finish** and reopen the web component.
   You can use the preview pane to take a quiz.

5. Click **Get Web Component Embed Code**. Copy the code to the clipboard or a file.

Configure the Web Component to Receive a Specific Quiz ID

The following embed code can be used for the Quiz web component to receive a specific quiz ID (or no ID). This is recommended in non-Bunchball Go programs to open a specific quiz from the mission's task list.

Note: The example code works when the Quiz web component is embedded as an iframe.

1. Save the following embed code to a .html file.

2. Replace the data-user-id and data-token parameters with values from your environment.

3. Replace the name parameter with the name of the Quiz web component you created in Nitro Studio.

4. Double-click the file (or put it on a web server) to show the web page. Then, in a mission rule's Rule URL field:

   - Call the code without a parameter (http://.../quizembed.html) to show the next available quiz.

   - Call the code with a parameter (http://.../quizembed.html?quizid=123) to show just quiz 123 (or whatever quiz would match your environment).

```
<!DOCTYPE html>
<html>
<head>
  <title>Sample Quiz Embed</title>
</head>
<body>
  <div id="quiz_to_change"
   name="MyQuiz"
   class="nitro-widget"
   width="400px"
   height="400px"
   data-user-id="user.name@email.com">
  </div>
  <script id="nitro-js"
   data-token="095cxxxxxxxxxxxxxxxc08d776"
   data-version="v6.1"
   src="https://widgets.bunchball.net/nitro/v6.1/nitro.min.js">
  </script>
  <script>
   var quizid = getUrlParameter("quizid");
   if (quizid != null) {
    document.querySelector("#quiz_to_change").setAttribute("data-quiz-id", quizid);
   }
   function getUrlParameter(name) {
    name = name.replace(/[\[]/, '\\[').replace(/[\]]/, '\\]');
    var regex = new RegExp('[\\?&]' + name + '=([^&#]*)');
    var results = regex.exec(location.search);
    return results === null ? '' : decodeURIComponent(results[1].replace(/\+/g, ' '));
    };
  </script>
</body>
</html>
```

Sample Embed Code

See Embed an OAuth Web Component for more information and the Web Components view for the most up-to-date embed code. The following is an example embed for a Quiz web component.

web component	iframe

<bunchball-quiz style="display:block; width:400px;"
data-user-id="AutoTest"
data-token="abcdefg1234567"
data-embed-id='0'
data-help-text='Interact with your users via a quiz.'
data-oauth-client='{}'
data-title='QUIZ'
data-empty-message='No quizzes available at the moment...'
data-bypass-resume='false'
data-custom-css=''>
</bunchball-quiz>
	<div name="Quiz"
	class="nitro-widget"
	width="400"
	height="400"
	data-user-id="AutoTest"
	data-name='Quiz'
	data-embed-id='0'
	data-height='400px'
	data-help-text='Interact with your users via a quiz.'
	data-locale='EN'
	data-oauth-client='{}'
	data-title='QUIZ'
	data-width='400px'
	data-custom-css='[""]'
	data-embed-type='iframe'
	data-empty-message='No quizzes available at the moment...'
	data-bypass-resume='false'>
	</div>

Recognitions Web Component

Overview

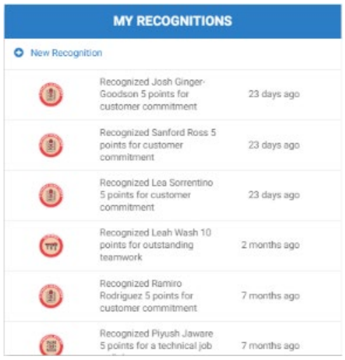
The Recognitions web component displays an interface where participants can recognize one another. Depending on your recognition setup, anyone can recognize anyone, managers recognize team members, or only people in the same group recognize each other for contributions and accomplishments and grant rewards based on a predetermined budget.

The web component has two views: a form and a list. The form is used to submit a recognition. The list shows either the recognitions the user has given or the recognitions the user has received. By default, the web component shows recognitions given along with a link to the form view. You can use multiple web component to show different combinations of information.

Recipient Lookup

The search in the "Who do you want to recognize" field is a prefix search. Typing "jo" will search for "jo*" which will match joe, josh, etc. The search supports "or" and "and", although "or" isn't necessary. Typing "jo mi"will find josh and michael in the search results. Typing "jo and fi" will find anyone with both of the letters as the beginning of words in their name, such as "josh finnis".

The recipient lookup searches firstName and lastName. If necessary, you can use the additionalName preference to add a preferred name to differentiate between users with the same or similar names. Show me



Recognized Josh Ginger-Goodson 5 points for customer commitment — 23 days ago

Recognized Sanford Ross 5 points for customer commitment — 23 days ago

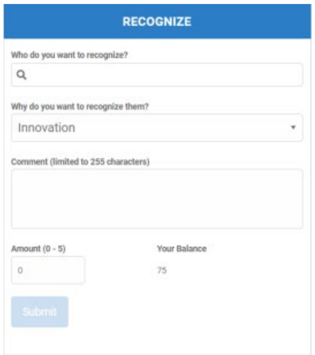
Recognized Lea Sorrentino 5 points for customer commitment — 23 days ago

Recognized Leah Wash 10 points for outstanding teamwork — 2 months ago

Recognized Ramiro Rodriguez 5 points for customer commitment — 7 months ago

Recognized Piyush Jaware 5 points for a technical job — 7 months ago

Main view



Who do you want to recognize?

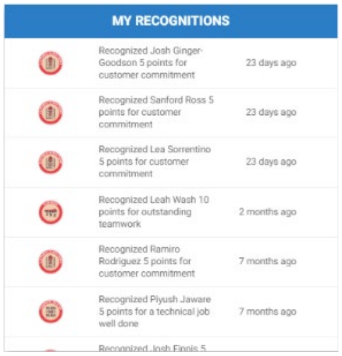
Why do you want to recognize them?
Innovation

Comment (limited to 255 characters)

Amount (0 - 5)
0

Your Balance
75

Form view with points



Recognized Josh Ginger-Goodson 5 points for customer commitment — 23 days ago

Recognized Sanford Ross 5 points for customer commitment — 23 days ago

Recognized Lea Sorrentino 5 points for customer commitment — 23 days ago

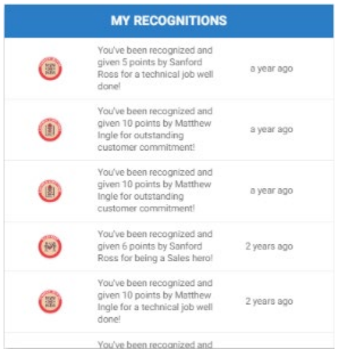
Recognized Leah Wash 10 points for outstanding teamwork — 2 months ago

Recognized Ramiro Rodriguez 5 points for customer commitment — 7 months ago

Recognized Piyush Jaware 5 points for a technical job well done — 7 months ago

Recognized Josh Finnis 5

Giver view



You've been recognized and given 5 points by Sanford Ross for a technical job well done! — a year ago

You've been recognized and given 10 points by Matthew Ingle for outstanding customer commitment! — a year ago

You've been recognized and given 10 points by Matthew Ingle for outstanding customer commitment! — a year ago

You've been recognized and given 6 points by Sanford Ross for being a Sales hero! — 2 years ago

You've been recognized and given 10 points by Matthew Ingle for a technical job well done! — 2 years ago

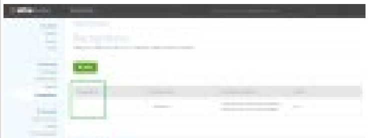
You've been recognized and

Recipient view

Configure a Web Component

1. Before adding a web component, configure recognition for your program.
2. Open Nitro Studio > **Integrate** > **Web Components**.
3. Click **New** > **Recognitions**.
4. Enter web component settings.
   Parameters are used when working directly with a web component embed tag (HTML) or editing the config.json file for Bunchball Go.

Studio Field	**Description**	**Parameters**				
		Web Component Embed HTML	**Bunchball Go config.json**	**Valid Values**	**Required (web component)**	**Required (iframe)**
Name and Display						
Name	The web component name. ▪ **web component** - set the component tag as bunchball-interaction. ▪ **iframe** - must match exactly with the name of the web component created in Nitro Studio.	name	name		No	Yes
Width	The web component's width.	width	width	relative (%) or absolute value (px)	No	No
Height	The web component's height.	height	height	relative (%) or absolute value (px)	No	No
Custom Styles	The link to your custom stylesheet(s). Accepts a comma delimited list of URLs.					
Embed Type	Specifies the type of embed code to generate. ▪ **web component** - Recommended. Web components improve	data-embed-type	embed-type	iframe	No	Yes

Studio Field	Description	Parameters				
		Web Component Embed HTML	**Bunchball Go config.json**	**Valid Values**	**Required (web component)**	**Required (iframe)**
	performance due to faster loading. Height and width settings are not required as the web component will fill space on the page like any other HTML element. - **iframe** - The source code is a div tag and the rendered web component will be inside an iframe.					
Basic						
User ID	The ID of the user to get recognitions for.	data-user-id	user-id		Yes	Yes
Exchange	The name of the exchange to use for recognitions. **Note:** The name will be converted to the exchange ID which is used in the embed. Click the image to see where to find the exchange ID. 	data-exchange-id	exchange-id		Yes	Yes
Return Count	The number of items to display.	data-return-count	return-count		No	No
View to Display	The view the web component should render. Defaults to main. - **recipient** - Shows recognitions received by the user. Uses the phrase set in the Recipient Phrase in the recognition category.	data-display	display	giver, recipient, main, form	No	No

Studio Field	Description	Parameters				
		Web Component Embed HTML	Bunchball Go config.json	Valid Values	Required (web component)	Required (iframe)
	▪ **giver** - Shows recognitions given by the user. Uses the phrase set in the Giver Phrase in the recognition category. ▪ **form** - Shows the recognition submission form. ▪ **main** - Shows both the submission form and recognitions given. Uses the phrase set in the Giver Phrase in the recognition category.					
Title (List View)	The title to display as the header of the list view. Defaults to My Recognitions.	data-title	title		No	No
Subtitle	The subtitle to display below the web component header.	data-subtitle	subtitle		No	No
Title (Form View)	The title to display as the header of the form view. Defaults to Recognize.	data-form-title	form-title		No	No
Start Time	The recognition feed's start date. Recognitions before this date will not appear in the web component. If not set, will show recognitions for the last three months. **Note:** The difference between the start and end time must be less than one year.	data-start-time	start-time	a UNIX timestamp	No	No
End Time	The recognition feed's end date. Recognitions after this date will not appear in the web component. If not set, will show recognitions up to the current date.	data-end-time	end-time	a UNIX timestamp	No	No

Studio Field	Description	Parameters				
		Web Component Embed HTML	Bunchball Go config.json	Valid Values	Required (web component)	Required (iframe)
Language	Add a language code to preview the web component in a Nitro supported default language. Default languages are: en, de, hi, zh, it, es, and fr.				No	No
Custom Language Reference	Add a language code and then add a custom translation key to the browser console to preview the web component in a non-default language.				No	No

5. Click **Save & Finish** and reopen the web component.
6. Click **Get Web Component Embed Code**. Copy the code to the clipboard or a file.
7. Include the code to generate an OAuth token in your app.
8. Insert the embed code into the page/div where you want to display the web component in your app or web page, and update the variable for the OAuth token.

Sample Embed Code

See Embed an OAuth Web Component for more information and the Web Components view for the most up-to-date embed code. The following is an example embed for a Recognitions web component.

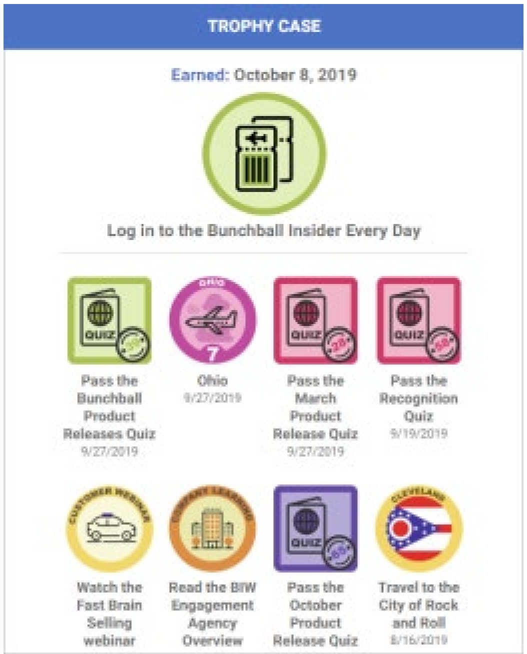
web component	iframe
`<bunchball-interaction style="display:block; width:400px;"`	`<div name="recognition"`
`data-token="abcdefg1234567"`	`class="nitro-widget"`
`data-user-id="AutoTest"`	`width="400"`
`data-exchange-id='1'`	`height="400"`
`data-form-title='Recognize'`	`data-name='recognition'`
`data-title='My Recognitions'`	`data-user-id="AutoTest"`
`data-custom-css=''>`	`data-exchange-id='1'`
`</bunchball-interaction>`	`data-form-title='Recognize'`

web component	iframe

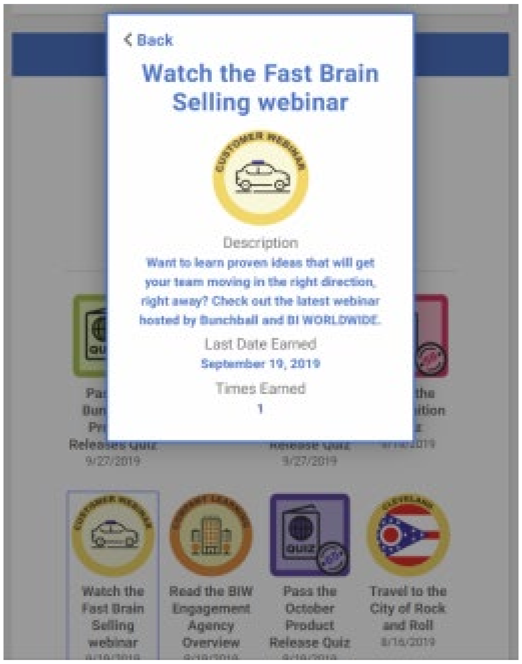
	data-height='400'
	data-title='My Recognitions'
	data-width='400'
	data-custom-css='[""]'
	data-embed-type='iframe'>
	</div>

Trophy Case Web Component

The trophy case showcases the badges/trophies earned by the user. The top section shows the most recently earned badge. The bottom section shows all badges earned by the user. From the main view, you can drill-down into a detail view showing the mission's description, the last time the badge was earned and how many times the badge was earned (for repeatable missions).


TROPHY CASE
Earned: October 8, 2019
Log in to the Bunchball Insider Every Day
Pass the Bunchball Product Releases Quiz 9/27/2019
Ohio 9/27/2019
Pass the March Product Release Quiz 9/27/2019
Pass the Recognition Quiz 9/19/2019
Watch the Fast Brain Selling webinar
Read the BIW Engagement Agency Overview
Pass the October Product Release Quiz
Travel to the City of Rock and Roll 8/16/2019

Main view


< Back

Watch the Fast Brain Selling webinar

Description

Want to learn proven ideas that will get your team moving in the right direction, right away? Check out the latest webinar hosted by Bunchball and BI WORLDWIDE.

Last Date Earned

September 19, 2019

Times Earned

1

Watch the Fast Brain Selling webinar
Read the BIW Engagement Agency Overview
Pass the October Product Release Quiz
Travel to the City of Rock and Roll 8/16/2019

Detail view

Configure a Web Component

1. Open Nitro Studio > **Integrate** > **Web Components**.
2. Click **New** > **Trophy Case**.
3. Enter the web component settings.
   Parameters are used when working directly with a web component embed tag (HTML) or editing the config.json file for Bunchball Go.

Studio Field	**Description**	**Parameters**				
		Web Component Embed HTML	**Bunchball Go config.json**	**Valid Values**	**Required (web component)**	**Required (iframe)**
Display Settings						
Name	The web component name. ▪ **web component** - set the component tag as bunchball-trophy. ▪ **iframe** - must match exactly with the name of the web component created in Nitro Studio.	name	name		No	Yes
Width	The web component's width.	width	width		No	Yes
Height	The web component's height.	height	height		No	Yes
Custom Styles	The link to your custom stylesheet(s). Accepts a comma delimited list of URLs.					
Embed Type	Specifies the type of embed code to generate. ▪ **web component** - Recommended. Web components improve performance due to faster loading. Height and width settings are not required as the web component will fill	data-embed-type	embed-type	iframe	No	Yes

Studio Field	Description	Parameters				
		Web Component Embed HTML	**Bunchball Go config.json**	**Valid Values**	**Required (web component)**	**Required (iframe)**
	space on the page like any other HTML element. ▪ **iframe** - The source code is a div tag and the rendered web component will be inside an iframe.					
Web Component Settings						
User ID	The ID of the user to get badges for.	data-user-id	user-id		Yes	Yes
Title	The title to display in the web component's header. Default is Trophy Case.	data-title	title		No	No
Subtitle	The subtitle to display below the web component header.	data-subtitle	subtitle		No	No
Empty Data Message	The message that displays when no badges have been earned. Defaults to "Any trophies you earn will appear here."	data-empty-message	empty-message		No	No
Latest Badge Only	If true, only shows the last badge the user earned.	data-latest-only	latest-only	true, false	No	No
Language	Add a language code to preview the web component in a Nitro supported default language. Default languages are: en, de, hi, zh, it, es, and fr.				No	No
Custom Language Reference	Add a language code and then add a custom translation key to the browser console to preview the web component in a non-default language.				No	No

Studio Field	Description	Parameters				
		Web Component Embed HTML	**Bunchball Go config.json**	**Valid Values**	**Required (web component)**	**Required (iframe)**
	The height of the trophy container below the last earned badge. Setting the height enables the scroll bar for that section. Default is 300.	data-trophy-container-height	trophy-container-height		No	No
	The number of badges that will be retrieved per infinite scrolling page. Defaults to 10.	data-page-size	page-size		No	No

4. Click **Save & Finish** and reopen the web component.
5. Click **Get Web Component Embed Code**. Copy the code to the clipboard or a file.
6. Include the code to generate an OAuth token in your app.
7. Insert the embed code into the page/div where you want to display the web component in your app or web page, and update the variable for the OAuth token.

Sample Embed Code

See Embed an OAuth web component for more information and the Web Components view for the most up-to-date embed code. The following is an example embed for a Trophy Case web component.

web component	iframe

`<bunchball-trophy style="display:block; width:400px;"`
`data-user-id="AutoTest"`
`data-token="abcdefg1234567"`
`data-empty-message='Any trophies you earn will appear here.'`
`data-latest-only='false'`
`data-title='Trophy Case'`
	`<div name="TrophyCase"`
	`class="nitro-widget"`
	`width="400"`
	`height="400"`
	`data-user-id="AutoTest"`
	`data-name='TrophyCase'`
	`data-embed-type='iframe'`
	`data-empty-message='Any trophies you earn will`

web component	iframe

data-custom-css=''>
</bunchball-trophy>
	appear here.'
	data-height='400'
	data-latest-only='false'
	data-title='Trophy Case'
	data-width='400'
	data-custom-css='[""]'>
	</div>

Embed an OAuth Web Component

Nitro JavaScript Library

The Nitro JavaScript Library is required for all web components. It identifies the web components you want to load and loads them. Do not locally store the library.

For each page where you want to display web components, Nitro JavaScript only needs to be included on the page one time. Once on the page, the script finds your web component embed tags and loads the resources they need to run.

Dynamic Updates

For programs where web components are not regularly refreshed via page navigation, you can use the "update" function to ensure you get the latest data from Nitro. We recommend you leverage this function on components where it's appropriate to get new data in response to a user action. It should not be used to poll our APIs on a timer. Show me example scenarios and code samples

Embed Steps

❗Embed code used in sandbox must be changed before being used in production. Some embed parameters, as outlined in the following steps, vary between sandbox and production environments.

The following steps describe a generic overview of embedding OAuth web components. Not all web components produce the same embed code. Follow the specific instructions and tips provided with the web component preview and resulting embed code to embed a web component on your site. To embed a block, see Embed a Block.

1. Use the Web Component Builder to configure, save, and get the embed code for your web component. **Note:** Embed code is not available until you save the web component.

2. Add the Nitro web component tag on your page in the location where you want the web component to display.

   - **For Web Components:**

     - Set the web component tag name: bunchball-profile, bunchball-missions, bunchball-events, bunchball-interaction, bunchball-leaders, bunchball-ots, bunchball-quiz, or bunchball-trophy.

     - Where web components support a data-user-id attribute, it should be set, along with using a 3-legged OAuth token.

Sandbox Example	Production Example

<bunchball-events style="display:block; width:400px; height:450px;"
data-token=%oauth_token%
data-server="https://api.sandbox.bunchball.com"
data-user-id="AutoTest"
data-export-navigation='false'
data-return-count='10'
data-title='Recent Achievements'
data-custom-css='[""]'>
</bunchball-events>
	<bunchball-events style="display:block; width:400px; height:450px;"
	data-token=%oauth_token%
	data-user-id="AutoTest"
	data-export-navigation='false'
	data-return-count='10'
	data-title='Recent Achievements'
	data-custom-css=''>
	</bunchball-events>

   - **For iframes**

     - Make sure the "name" attribute in the embed code div name matches the name of the web component you created in Nitro Studio. This uses the web component parameters defined in Nitro Studio.
       To bypass pulling the web component data from Nitro Studio, hardcode the name using the data-element-type parameter. Element

types include: QUIZ, MINI_PROFILE, EVENTS, MISSIONS, OTS, LEADERS, TROPHY.

- The "class" attribute allows nitro.js to recognize the tag as a web component embed tag.
- Where web components support a data-user-id attribute, it should be set, along with using a 3-legged OAuth token.

Sandbox and Production Example

```
<div name="Events"
  class="nitro-widget"
  width="400"
  height="450"
  data-user-id="AutoTest"
  data-name='Events'
  data-custom-css='[""]'
  data-embed-type='iframe'
  data-export-navigation='false'
  data-return-count='10'
  data-title='Recent Achievements'
  data-width='400'>
</div>
```

3. You can place text or other content between the opening and closing tags to provide loading text or a loading image until the content loads.
   You can also provide alternate content to display to your users instead of Nitro content. You may want to do this in case the Nitro servers can't be reached. To provide alternate content, use the <nitro:block>, <nitro:content> and <nitro:alt> tags.

4. Add Nitro JavaScript to your page as a direct child of the <body> element or after the </body> tag. You should not nest this script under another element, like a <div>. The code that gets executed upon inclusion appends a new <div> to the <body> tag, which cannot happen from within a nested element. This code only needs to be included on the page once. After adding the script, change the values for your environment.

   - **web component** - Add component.min.js BEFORE all web component elements.

Sandbox

```
<script src="https://components.bunchball.com/web/latest/component.min.js"></script>
```

Production

```
<script src="https://components.bunchball.com/web/v6.1/component.min.js"></script>
```

- **iframe** - Add nitro.min.js AFTER all web component tags. You must use an "id" attribute of "nitro-js" as well as the src, and data-token attributes. We also recommend you use a data-version attribute which differs in Sandbox and Production environments.

Sandbox

```
<script id="nitro-js"

    data-token=%oauth_token%

    data-server="https://api.sandbox.bunchball.com"

    data-version="latest"

    src="https://components.bunchball.com/nitro/latest/nitro.min.js"

    data-custom-css='[""]'></script>
```

Production

```
<script id="nitro-js"

    data-token=%oauth_token%

    data-version="v6.1"

    src="https://components.bunchball.com/nitro/v6.1/nitro.min.js"

    data-custom-css=''></script>
```

Embed Parameters

Parameter	Description	Sandbox	Production
src *(required)*	The URL that the nitro.js script is loaded from (Nitro JavaScript Library).	"https://components.bunchball.com/nitro/latest/nitro.min.js"	▪ **web component** - https://components.bunchball.com/web/v6.1/component.min.js ▪ **iframe** - https://components.bunchball.com/nitro/6.1/nitro.min.js
id *(required for iframe only)*	The id attribute of the script element. Must be set to nitro-js. Required for iframe only.	"nitro-js"	"nitro-js"

Parameter	Description	Sandbox	Production
data-token *(required for iframe only)*	The access token received from a token request. For web components, the data token is included in the web component tag.	Set to populate the variable for the OAuth token. Replace the parameter with a variable containing the OAuth token at runtime.	Set to populate the variable for the OAuth token. Replace the parameter with a variable containing the OAuth token at runtime.
data-server		"https://api.sandbox.bunchball.com"	Not needed for Production environments
data-version	The version of web component code that you want to run. Bunchball recommends that you bind to a version like "v6.1" to avoid instability.	"latest"	"v6.1"
data-cache-enabled	Allows you turn off caching of session and web component embeds in the browser's localStorage.		
data-debug	Enables verbose logging in the browser's development console.		
data-custom-css	The URL of your custom stylesheet if you want to override the default styles.		

Example Embed Code

Web Component

Sandbox	Production
<script src="https://components.bunchball.com/web/latest/component.min.js"></script> <bunchball-events style="display:block; width:400px; height:450px;"	<script src="https://components.bunchball.com/web/v6.1/component.min.js"></script> <bunchball-events style="display:block; width:400px; height:450px;"

Sandbox	Production

data-token=%oauth_token%
data-server="https://api.sandbox.bunchball.com"
data-user-id="AutoTest"
data-export-navigation='false'
data-return-count='10'
data-title='Recent Achievements'
data-custom-css='[""]'>
</bunchball-events>
	data-token=%oauth_token%
	data-user-id="AutoTest"
	data-export-navigation='false'
	data-return-count='10'
	data-title='Recent Achievements'
	data-custom-css=''>
	</bunchball-events>

iframe

Sandbox	Production

<HTML>
<head><title>My Web Component</title> </head>
<body>
<div name="Events"
class="nitro-widget"
width="400"
height="450"
data-user-id="AutoTest"
data-name='Events'
data-custom-css='[""]'
data-embed-type='iframe'
data-export-navigation='false'
data-return-count='10'
data-title='Recent Achievements'
data-width='400'>
<script id="nitro-js"
data-token=%oauth_token%
data-server="https://api.sandbox.bunchball.com"
data-version="latest"
src="https://components.bunchball.com/nitro/latest/nitro.min.js"
data-custom-css='[""]'></script>
</body></HTML>
	<HTML>
	<head><title>My Web Component</title></head>
	<body>
	<div name="Events"
	class="nitro-widget"
	width="400"
	height="450"
	data-user-id="AutoTest"
	data-name='Events'
	data-custom-css='[""]'
	data-embed-type='iframe'
	data-export-navigation='false'
	data-return-count='10'
	data-title='Recent Achievements'
	data-width='400'>
	<script id="nitro-js"
	data-token=%oauth_token%
	data-version="v6.1"
	src="https://components.bunchball.com/nitro/v6.1/nitro.min.js"
	data-custom-css='[""]'></script>
	</body>
	</HTML>

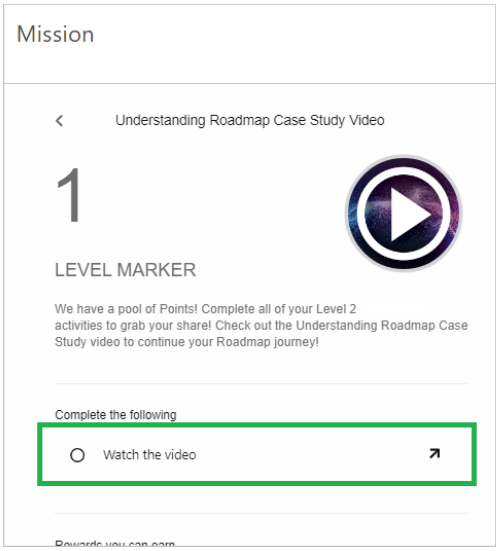
Missions Web Component Click Through - Code Sample

The Missions web component has a built-in interface for action logging on linked mission rules (rules that have URLs for metadata). In Bunchball Go, a HUB_CLICK_THROUGH action is logged when a user clicks on a linked mission task. The action is logged prior to routing the user to the URL's destination.

Using the Missions web component in a non-Bunchball Go program requires you to implement both a client and server side solution. In Impact, a self-reporting rule should be used to log the action when a user marks the task as complete.

Note: Tracking a click through action does not guarantee that the user consumed the content. However, the Quiz web component can be used in parallel to test content proficiency.

The Missions web component displays an arrow in the detail view's task list to indicate the rule is linked to other content.



Mission

Understanding Roadmap Case Study Video

1

LEVEL MARKER

We have a pool of Points! Complete all of your Level 2 activities to grab your share! Check out the Understanding Roadmap Case Study video to continue your Roadmap journey!

Complete the following

○ Watch the video ↗

Rewards you can earn

Use Case

Adding a linked rule to a mission allows the user to find the content required for the task easily and to be rewarded for accessing it. You may want to track a click through action if:

- your system cannot track or send notification of a view back to Nitro

- your system would require significant custom development to track or send notification of a view back to Nitro

Process Flow

The following steps need to occur to trigger a Nitro action.

1. The user clicks the link in the Missions web component, which triggers an event to the browser.

2. Client-side code detects the event and initiates the call to the server. See the Client Side Implementation section.

3. Server-side code sends the click through action to the Nitro API. See the Server Side Implementation section.

Client Side Implementation

The client side implementation requires the window.logAction function. The Missions web component looks for this function when users click a linked rule. If the function exists, it is invoked and sends the metadata as arguments. This function has one job, to relay the action data to Nitro so that it can log the action on behalf of the user. It does this by performing an XHR or ajax request to the server side endpoint defined in the server side implementation section.

JavaScript Code Example

```
function makeRequest(url, opts, callback){
 var options = opts || {};
 var request = new XMLHttpRequest();
 request.open(opts.method || "GET", url, true);
 request.setRequestHeader('Content-Type', 'application/json; charset=UTF-8');
 request.onload = function(){
  if(request.status >= 200 && request.status < 400) {
   // Success!
   var data = JSON.parse(request.responseText);
   if(callback){
    callback(request, data);
   }
  }
  else if(callback){
   callback(request, null);
  }
 }
 request.onerror = function(err) {
  // There was a connection error of some sort
  if(callback){
   callback(request, null, err);
  }
 }
 if(options.data || (options.method == "POST" || options.method == "PUT")) {
  request.send(JSON.stringify(options.data));
 } else {
  request.send();
 }
}

function logAction(actionTag, data, callback){
 var url = '/actions/' + actionTag;
 makeRequest(url, {
  method: "POST",
  data
 }, callback)
}
```

Server Side Implementation

The server side implementation should be a restful endpoint that accepts and returns JSON. For example, if the defined route is /actions, then actions should receive a POST request. The payload should include the action name, the user ID, and url metadata. When the request is received, a new server side request should be made to get a 2-legged token. That token should be used to log the action for the user passing along the action tag and metadata.

❗The 2-legged credentials should not be exposed to the client. If they are, they can be retrieved and used to compromise the program.

Python Code Example

```
@plugin.route('actions/<action_tag>', methods=["POST"])
def log_action(action_tag):
    action_logger = ActionLogger()
    action_payload = request.get_json()
    success = jsonify({"response": "Action logged", "code": 200}), 200

    if current_user.is_authenticated:

        elif action_tag == "HUB_CLICK_THROUGH":
            action_logger.log_action(Actions.click_through, current_user, raise_exception=False,
payload=action_payload) # noqa
            return success

        else:
            current_app.logger.warning('Action not found {}'.format(action_tag))
            return jsonify({"error": "Action not found", "code": 404}), 404

    else:
        current_app.logger.warning('User is not authenticated')
        return jsonify({"error": "No authentication found", "code": 403}),
403
```

Server Side Action Logger Details

```
from enum import Enum
from urllib.parse import unquote
from src.config import Config
from src.exceptions import ApiGatewayException, SecurityError
from src.plugins.auth.oauth import OauthFactory

class Actions(Enum):
    click_through = 'HUB_CLICK_THROUGH', 'Content clicked through'

    def __new__(cls, name, description):
        obj = object.__new__(cls)
```

```python
        obj._value_ = name
        obj._description = description
        return obj

    def description(self):
        return self._description

    @staticmethod
    def url(gamification_id):
        return '{}/users/{}/actions'.format(Config.SERVICE_API_GATEWAY, gamification_id)

    def payload(self, payload):
        if payload:
            return self.merge_dictionaries({'name': self._value_},payload)

        return {'name': self._value_}

    def merge_dictionaries(self, a, b, path=None):
        if path is None: path = []
        for key in b:
            if key in a:
                if isinstance(a[key], dict) and isinstance(b[key], dict):
                    self.merge_dictionaries(a[key], b[key], path + [str(key)])
                elif a[key] == b[key]:
                    pass # same leaf value
                else:
                    raise Exception('Conflict at %s' % '.'.join(path + [str(key)]))
            else:
                a[key] = b[key]
        return a



class ActionLogger:
    def __init__(self):
        self.backend_oauth = OauthFactory().backend_oauth()
```

```
  def log_action(self, action, user, raise_exception=False, payload={}):
    if user:
      # sanitize gamification_id from url encoded characters
      user.gamification_id = unquote(user.gamification_id)

      try:
        url = action.url(user.gamification_id)
        response = self.backend_oauth.post(user.ics, user.api_key, url, payload=action.payload(payload))
        if response.status_code == 200:
          return self.success(action, user)

        self.failure(action, user, raise_exception=raise_exception)
      except ApiGatewayException as e:
        if raise_exception:
          raise e
        else:
          current_app.logger.warning('failed to log action {}: {}'.format(url, e))
          return
    else:
      if raise_exception:
        raise SecurityError('User context unavailable. Please log in', 401)

  @staticmethod
  def success(action, user):
    current_app.logger.info('logged action ({0}) for
{1.ics}/{1.api_key}/{1.gamification_id}'.format(action.value, user))

  @staticmethod
  def failure(action, user, details={}, raise_exception=False):
    current_app.logger.error('could not log action ({0}) for {1.ics}/{1.api_key}/{1.gamification_id}'
          .format(action.value, user), **details)

    if raise_exception:
      raise ApiGatewayException(user.ics, user.api_key, user.gamification_id,
                   message="Unable to log {} action".format(action.value))
```

Quiz Web Component - Code Sample

This example shows how you can display the Quiz web component via php code.

Nitro.php

```
<?php
class Nitro {
  /**
  * Generate an access token
  *
  * @param $clientId : OAuth client ID (string | required)
  * @param $clientSecret : OAuth client secret (string | required)
  * @param $threeLegToken : Generate 3 leg token when true (bool | optional)
  * @param $userId : User ID to pass in (string | optional)
  */
  public function makeOAuthToken($clientId, $clientSecret, $threeLegToken = false, $userId = ''){
    $endpoint = "https://api.sandbox.bunchball.com/oauth/token/";
    $headers = ['Content-Type: application/x-www-form-urlencoded'];

    $data = "grant_type=client_credentials&client_id=".$clientId."&client_secret=".$clientSecret;
    if($threeLegToken){
    $data .= "&xoauth_end_user_id:" . $userId;
    }
    $response = $this->executePost($endpoint, $data, $headers);
    return $response;
  }
  /**
  * Execute URL (POST)
  *
  * @param $url : URL to be passed in (String | Required)>
  * @param $data  : Post data to send (Array  | Required)
  * @param $headers : Headers to send with request (Array | Optional)
  */
  public function executePost($url, $data, $headers = []){
    $ch = curl_init($url);
    curl_setopt($ch, CURLOPT_POST, true);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    $response = curl_exec($ch);
    $status = curl_getinfo($ch, CURLINFO_HTTP_CODE);

    curl_close($ch);
    return $this->response = $response;
  }
}
?>
```

quiz.php

```
<?php

require_once('nitro.php');
$clientId = 'gg0eXj6Kdo';
$clientSecret = 'MCgaON91T1wmIfUrhlVle';
$userLoginId = 'salesdemo';
$nitro = new Nitro();
//Generate three legged token
$threeLeggedToken = $nitro->makeOAuthToken($clientId, $clientSecret, true, $userLoginId);/p>
if(!empty($threeLeggedToken)) {
  //do something like store it in a db or something
  //print_r($threeLeggedToken);
} else {
  print "Failed fetching access token, response was: " . $nitro->getLastResponse();
}
$authObj = json_decode($threeLeggedToken);
if (isset($authObj->access_token)){
  global $accessToken;
  $accessToken = $authObj->access_token;
}
```

```
//Generate two legged token
//$twoLeggedToken = $nitro->makeOAuthToken($clientId, $clientSecret);
?>
<!DOCTYPE html>
<html xmlns:nitro="http://www.bunchball.com/nitro">
<head>
  <title></title>
  <meta charset="utf-8">
</head>
<body>
<div name="My Quiz"
  class="nitro-widget"
  width="400px"
  height="400px"
  data-user-id="salesdemo"></div>
  <script id="nitro-js"
  data-token="<?php echo $accessToken ?>"
  data-server="https://api.sandbox.bunchball.com"
  data-version="v6.1"
  src="https://components.bunchball.com/nitro/v6.1/nitro.min.js"
  ></script>
</body>
</html>
```

Localization

Nitro's built-in language support allows blocks, web components, and APIs to work in collaboration with user-facing content translations to achieve a fully translated program. Show me a diagram of the localization process

To translate your program, you will need to:

1. Enable localization. Show me supported locales

2. Translate your program content such as missions, levels, and recognitions.

3. Translate your front-end UI. This front-end may be blocks, web components, or a custom program created with APIs.

   Note: Bunchball Go can be used globally but does not support localization in the front-end UI.

Enable Localization and Translate Program Content

Step 1: Enable localization in Nitro Studio. Show details

1. Open Nitro Studio > **Configuration** > **Localization**.

2. In the **Enable Locales** list, select the locales your site is using.

   Localization must be enabled before it's available in your environment. Contact Bunchball Support for assistance.

Step 2: Translate your mission information, levels, exchanges, and other program content. Show details

An Excel or XML translation file is used to localize program content. Each locale you choose to support has its own translation file which is managed in Nitro Studio. Translation text within the Excel or XML file must be in Ascii format.

1. Open Nitro Studio > **Configuration** > **Localization**.

2. In the **Enable Locales** list, select all the locales in your program.

3. Under **Translate Content**:

   a. Select **Excel** or **XML** as your file type.

   b. In the **Download Translation File** list, select the language file you want to download.

   c. Click **Download**.

      ❗When downloading translatable data, you must download from the instance you want to localize. You cannot download a template from your Sandbox instance and try to upload it to Production. Although the content may be the same between instances, the object IDs are different.

4. Update the value field for every value you want to localize and save the file. Leaderboard translations only apply to Impact programs. You must translate leaderboard names separately when using web components.

5. When you've finished translating your content, open the **Localization** page and click **Upload Translation File**.

   Note: If you receive an error when uploading a translation file, make sure the file size is less than 1MB. If you have a large number of missions, you may need to split your data into multiple files for upload.

6. Repeat steps 3-5 for each language your program supports.

7. As your program grows, you will need to repeat this process to translate the new content.

Note: If you upload a translation file and then download via the same browser, the file may not contain all translations until you refresh the cache.

Translate User Facing Elements

Web Components and Custom Programs

Nitro supports seven default languages for web components: English, French, Spanish, Italian, German, Mandarin (Simple), and Hindi. A translation schema in each default language is maintained by Nitro. Depending on your program configuration, you can reference the default language translation schemas to translate your program. However, any text overrides, quizzes, leaderboard names, and blocks will require additional translation. When translating into a non-default language, you must create your own translation schema(s).

Complete the following steps as needed to translate your program. Skip any steps that do not apply to your program configuration.

Step 1: Translate web components into a default language (English, French, Spanish, Italian, German, and Mandarin (Simple), and Hindi). Show details

When translating your program into a default language, a Nitro translation schema exists for the text strings used in web components. You don't need to create your own translation schema or modify the default schema unless you have single text overrides or quizzes.

1. Use the **Localization** view in Nitro Studio to find your language code. Use a 2-digit code to translate into the base language and a 4-digit code to translate into the locale.

2. If your program uses the Missions web component, check the embed code for the "eligible-missions-tab-name" and "completed-missions-tab-name" parameters and remove them.

3. Add the "data-language" attribute and the language code to the web component embed code.

   If you have multiple languages, use a token that can be replaced at runtime with the user's locale.

Show me an example embed code

```
<bunchball-events
  data-token="aa41ff8a8e2e52xxxxxxxc096d0ff78f"
  data-server="https://api.bunchball.com"
  data-user-id=""
  data-export-navigation='false'
  data-return-count='10'
  data-language='fr'
  data-custom-css=''
</bunchball-events>
```

Step 2: Translate web components into a non-default language (any language other than English, French, Spanish, Italian, German, and Mandarin (Simple), and Hindi). Show details

When translating your program into a non-default language, you must create a translation schema for all default text strings that are used in web components. You will need to translate every string in the translation key.

1. Create a translation schema.

2. Add the translation schema to the page containing your web component embed code.

3. Use the **Localization** view in Nitro Studio to find your language code. Use a 2-digit code to translate into the base language and a 4-digit code to translate into the locale.

4. Add the "data-translation-reference" attribute and the language code to the web component embed code to reference your translation schema.

   If you have multiple languages, use a token that can be replaced at runtime with the user's locale.

Show me an example script

```
<script>
 window.BunchballTranslations = {
  ar: {
   translation: {
    “profile”: {
      "title": “ملفك الشخصي”
    }
   }
  }
</script>

 <bunchball-profile
  data-token="<token>"
  data-user-id="<userId>"
  data-language="<locale code>"
  data-translation-reference="BunchballTranslations">
 </bunchball-profile>
```

Step 3: Translate web component single text string overrides. Show details

If you customized text within a web component, such as the title, subtitle, or button name, you must translate that custom text into all your supported languages. You can use a single text override or translation schema to change the default text to your custom value.

1. In your web component embed code, remove the value for parameters you want to override with a custom value. For example, you changed the title of the Events web component from Recent Achievements to Community Achievements. The title parameter for the Events web component is "title": "Recent Achievements". You would change that parameter to "title": "".

2. If you're only translating into one language, add your translated text directly to the parameter in the embed code.

3. If you're translating into multiple languages, add your custom values to your translation schema. Add a custom override for each language your program supports.

4. Add the "data-translation-reference" attribute and the language code to the web component embed code to reference your translation schema.

   If you have multiple languages, use a token that can be replaced at runtime with the user's locale.

Show me an example script This example shows how you can use a translation schema to override the default (Spanish) translation within a Profile web component to use a different title. Rather than including a reference in the language file, you can specify a single key value pair that will be merged in to override the default.

```
<script>
 window.BunchballTranslations = {
  es: {
   translation: {
    "profile": {
      "title": “Votre profil”
    }
   }
  }
</script>

 <bunchball-profile
  data-token="<token>"
  data-user-id="<userId>"
  data-language="<locale code>"
  data-translation-reference="BunchballTranslations">
 </bunchball-profile>
```

Step 4: Translate custom web components created using APIs. Show details

1. Use the **Localization** view in Nitro Studio to find your language code. Use a 2-digit code to translate into the base language and a 4-digit code to translate into the locale.

2. Add the "Accept-Language" header and language code to all your API calls.

❗The header is required on every API call as each is a single transaction and there is no session data.

Step 5: Translate quizzes. Show details

Quiz content (quiz name, questions, and answers) is translated as flat text.

❗Translation overrides are EXACT matches on the quiz content including the quiz title and description, question text, and answer text. Matches are case-sensitive.

1. Create a translation schema containing your quizzes, their questions, and their answers (or add quizzes to your existing translation schema). No quiz, question, or answer IDs are needed.

2. If you're translating into multiple languages, add a custom override for each language your program supports.

3. Add the "data-translation-reference" attribute to the web component embed code to reference your translation schema.

Show me an example script

```
<script>
 window.BunchballTranslations = {
  es: {
   translation: {
    // quiz 1
    "Space": "Espacio",
    "The space quiz": "la prueba del espacio",

    "Is Pluto a planet?": "¿Plutón es un planeta?",
     "no": "no",
     "yes": "sí",

    "Why do the moon and sun look bigger on the horizon?": "¿La atmósfera de la Tierra es una lupa?",
        "Earth's atmosphere is a magnifying glass": "La atmósfera de la Tierra es una lupa",
        "We have tears in our eyes watching them": "Tenemos lágrimas en los ojos mirándolos",
        "Earth's gravity bends light so we 'see' them before seeing them": "La gravedad de la Tierra dobla la luz para que los 'veamos' antes de verlos",

      "What is Ceres?": "¿Qué es Ceres?",
        "Dwarf planet": "Planeta enano",
        "Asteroid": "Asteroide",
        "Constellation": "Constelación"
    }
   }
 }
</script>

<bunchball-quiz class="shadow"
 data-token="12345678"
 data-user-id="TestUser"
 data-translation-reference="BunchballTranslations">
</bunchball-quiz>
```

Step 6: Translate leaderboard names. Show details

If your program does not use the Leaders web component to show multiple leaderboards, you do not need to translate leaderboard names.

1. In your Leaders web component embed code, get the value in the "data-leaderboard-configs" parameter.
2. Translate the "name" parameters within the string into the desired language(s).
3. In your translation schema, add the full string, with translations, within the "leaderboardConfigs" text key.

Show me an example script

```
<script>
 windows.BunchballTranslations = {
 es-MX : {
  translation: {
     leaders: {
       leaderboardConfigs: “[{"name":"todos los puntos de
tiempo","leaderboardId":"215","groupClassIds":"","showProfile":"true","criteria":"lifetimeBalance"}]”
     }
   }
 }
</script>
```

Translation Schema

A translation schema is used to extend language support to additional languages, custom text strings, and quiz content. The translation schema is added to your web page using the following format. Web component's embed code should reference this schema.

```
{
  <language code> : {
   translation: {
     "<component name>" : {
         "<text key>": "<translation of text key in the target language>"
     }
   }
 }
```

Show me an example translation schema

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Localization Testing</title>
   <link href="https:///npm/dist/css/min.css" rel="stylesheet" integrity="sha3" crossorigin="anonymous">
</head>
<body>
```

```
 <script src="https://components.bunchball.com/web/latest/component.min.js"></script>

 <script>
   window.ExampleTranslations = {
     en: {
       translation: {
         "leaders": {
           "title": "Leaders",
           "subtitle": "Complete missions to earn your spot on the leaderboard!",
           "leaderboardConfigs": “[{"name":"Weekly Coin Leaders","leaderboardId":
           "3929","groupClassIds":"9112,9279","showProfile":"true","criteria":"lifetimeBalance"},
           {"name":"Monthly Coin Leaders","leaderboardId":"3929","groupClassIds":"9112,9279",
           "showProfile":"true","criteria":"lifetimeBalance"},{"name":"All-time Coin Leaders",

"leaderboardId":"3932","groupClassIds":"9112,9279","showProfile":"true","criteria":"lifetimeBalance"}]'
         }
       }
     },
     de: {
       translation: {
         "leaders": {
           "title": "Führer",
           "subtitle": "Schließe Missionen ab, um dir einen Platz auf der Bestenliste zu verdienen!",
           "leaderboardConfigs": “[{"name":"Wöchentliche Münzführer","leaderboardId":"3929",
           "groupClassIds":"9112,9279","showProfile":"true","criteria":"lifetimeBalance"},
           {"name":"Monatliche Münzführer","leaderboardId":"3929","groupClassIds":"9112,9279",
           "showProfile":"true","criteria":"lifetimeBalance"},{"name":"Allzeit-Coin-Leader",

"leaderboardId":"3932","groupClassIds":"9112,9279","showProfile":"true","criteria":"lifetimeBalance"}]'
         }
       }
     },
     es: {
       translation: {
         "leaders": {
           "title": "Dirigentes",
           "subtitle": "¡Completa misiones para ganar tu lugar en la tabla de clasificación!",
           "leaderboardConfigs": “[{"name":"Líderes de monedas semanales","leaderboardId":"3929",
           "groupClassIds":"9112,9279","showProfile":"true","criteria":"lifetimeBalance"},
           {"name":"Líderes mensuales de monedas","leaderboardId":"3929","groupClassIds":"9112,9279",
           "showProfile":"true","criteria":"lifetimeBalance"},{"name":"Líderes de monedas de todos los
tiempos",

"leaderboardId":"3932","groupClassIds":"9112,9279","showProfile":"true","criteria":"lifetimeBalance"}]'
         }
       }
     },
     zh: {
       translation: {
         "leaders": {
           "title": "领导人",
           "subtitle": "完成任务，在排行榜上赢得一席之地!",
           "leaderboardConfigs": “[{"name":"每周硬币领袖
","leaderboardId":"3929","groupClassIds":"9112,9279",
           "showProfile":"true","criteria":"lifetimeBalance"},{"name":"每月硬币领袖
","leaderboardId":"3929",
           "groupClassIds":"9112,9279","showProfile":"true","criteria":"lifetimeBalance"},{"name":"历史上
的硬币领袖",

"leaderboardId":"3932","groupClassIds":"9112,9279","showProfile":"true","criteria":"lifetimeBalance"}]'
         }
       }
     },
     it: {
       translation: {
         "profile": {
           "title": "Mi perfil",
           "subtitle": "",
           "level": "Nivel",
           "next": "próximo",
           "maxLevelAchievedMessage": "",
           "notApplicable": "N / A",
           "redeemButton": "Redimir",
           "firstName": "Primer nombre:",
           "firstNamePlaceholder": "Escriba el nombre aquí ...",
           "lastName": "Apellido:",
           "lastNamePlaceholder": "Escriba el apellido aquí ...",
           "awardBalance": "Saldo de premios",
           "uploadImage": "Cargar imagen",
           "removeImage": "Quita la imagen",
           "availableBalance": "Saldo disponible:",
           "accountNotVerified": "Cuenta no verificada",
           "currentBalance": "Saldo actual:",
           "balanceNotAvailable": "No disponible",
           "balanceTitle": "Cuenta De Premio"
         },
         "missions": {
           "title": "Misiones",
           "subtitle": "",
           "missions": "Misiones",
           "userMissions": "Mis Misiones",
           "missionType": "Seleccione el Tipo de Misión",
           "pointsLeft": "Solo quedan {{points}} puntos",
           "completionsLeft": "Solo quedan {{completions}} terminaciones",
           "timeRemaining": "Solo queda {{time}}",
           "hour": "{{count}} hora",
           "hour_plural": "{{count}} horas",
           "day": "{{count}} día",
           "day_plural": "{{count}} días",
           "startsInLessThan": "La misión comienza en menos de",
           "endsInLessThan": "Termina en menos de {{time}}",
           "expires": "La missione scade",
           "rewardExpired": "Premio scaduto!",
           "rewardExpires": "Il premio scade",
           "rightNow": "proprio adesso",
           "on": "su",
           "completionCount": " (tiempo {{count}} completado)",
           "completionCount_plural": " (completado {{count}} veces)",
		"completionCountOnMissionCard": "Completado {{count}} completado",
		"completionCountOnMissionCardPlural": "Completado {{count}} veces",
		"repeatable": "Repetible",
		"group": "Misión de Grupo",
		"groupMembers": " como un grupo",
		"completeAny": "Complete lo siguiente",
		"completeAll": "Complete <b>TODO</b> lo siguiente",
		"completeOne": "Complete <b>UNO</b> de los siguientes",
		"completeAllInSequence": "Complete todo lo siguiente en el <b>ORDEN dado</b>",
		"rewards": "Recompensas que puede ganar",
		"badge": "INSIGNIA",
		"noRewards": "No se dan recompensas",
		"expiringSoon": "Expirará Pronto",
		"inProgress": "En Progreso",
		"filters": "Filtros",
		"locked": "Bloqueado",
		"lockedByChallengeDescription": "Debe completar ",
		"lockedByLevelDescription": "Debe ser ",
		"lockedSectionSimple": "La misión está bloqueada",
		"ineligibleMission": "No eres elegible para esta misión",
		"lockedSectionMeetOne": "Cumplir con el siguiente requisito para desbloquear esta misión",
		"lockedSectionMeetAll": "Conoce <b>TODOS</b> los siguientes para desbloquear esta misión",
		"lockedSectionMeetAny": "Conoce a <b>UNO</b> de los siguientes para desbloquear esta misión",
		"emptyMessage": "No Se Encontraron Misiones",
		"emptyMessageDetail": "Misión Seleccionada No Encontrada",
		"remaining": "({{current}} de {{goal}} completado)",
		"close": "Cierre",
		"challengeUrlText": "Hacer la misión",
		"isExpired": "La misión ha expirado",
		"lockedSectionTitle": "¿Por qué está bloqueada esta misión?",
		"selfReportingConfirmationText": "Al marcar como completado, certifico que he completado esta tarea.",
		"complete": "Completo",
		"cancel": "Cancelar",
		"todo": "Que hacer",
          "completed": "Terminado",
          "rewardInLimitedSupply": "Recursos limitados",
          "outOfStock": "Agotado",
          "limitedSupply": "Recursos limitados",
          "startsIn": "Comienza la misión ",
          "completeOneOfTheseMissions": "Completa una de estas misiones."
        },
        "events": {
          "title": "Raggiunto",
          "subtitle": "",
          "emptyMessage": "No hay actividad reciente para mostrar"
        },
        "leaders": {
          "title": "Líderes",
          "subtitle": "Completa le missioni per guadagnare il tuo posto in classifica!",
          "yourValue": "Tu valor:",
          "notIncluded": "Parece que no estás en esta tabla de clasificación",
          "emptyMessage": "No hay líderes para mostrar"
        },
        "trophy": {
          "title": "Estuche de trofeo",
          "subtitle": "",
          "emptyMessage": "Los trofeos que ganes aparecerán aquí.",
          "networkError": "No se pudieron recuperar los datos para mostrar.",
          "back": "atrás",
          "description": "Descripción",
          "lastDateEarned": "Última fecha de obtención",
          "timesEarned": "Tiempos ganados",
          "earned": "Ganado:"
        }
      }
    }
  }
</script>

<div class="container text-center mt-2">
```

```
    <div class="row">
      <div class="col">
        <bunchball-missions style="display:block; width:400px; height:450px;"
          data-token="<token>"
          data-server="https://api.bunchball.com"
          data-user-id="<userId>"
          data-hide-filter-expiring='false'
          data-hide-filter-inprogress='false'
          data-hide-filter-labels='false'
          data-log-click-through='false'
          data-navigation='inline'
          data-return-count='5'
          data-show-completed-missions='true'
          data-show-date-as-duration='true'
          data-show-eligible-missions='true'
          data-show-forced-label-filter='false'
          data-show-locked='all'
          data-language='<locale code>'
          data-translation-reference="ExampleTranslations"
          data-custom-css=''>
        </bunchball-missions>
      </div>
      <div class="col">
        <bunchball-leaders style="display:block; width:400px; height:450px;"
          data-token="<token>"
          data-server="https://api.bunchball.com"
          data-user-id="<userId>"
          data-leaderboard-configs='[{"name":"Weekly Coin
Leaders","leaderboardId":"3929","groupClassIds":"9112,9279",
          "showProfile":"true","criteria":"lifetimeBalance"},{"name":"Monthly Coin
Leaders","leaderboardId":"3929",
          "groupClassIds":"9112,9279","showProfile":"true","criteria":"lifetimeBalance"},{"name":"All-time
Coin Leaders",

"leaderboardId":"3932","groupClassIds":"9112,9279","showProfile":"true","criteria":"lifetimeBalance"}]'
          data-page-size='25'
      data-language='<locale code>'
      data-translation-reference="ExampleTranslations"
      data-custom-css=''>
    </bunchball-leaders>
  </div>
  <div class="col">
    <bunchball-trophy style="display:block; width:400px; height:450px;"
      data-token="<token>"
      data-server="https://api.bunchball.com"
      data-user-id="<userId>"
      data-latest-only='false'
      data-language='<locale code>'
      data-translation-reference="ExampleTranslations"
      data-custom-css=''>
    </bunchball-trophy>
  </div>
</div>
<div class="row mt-2" >
  <div class="col">
    <bunchball-profile style="display:block; width:400px; height:450px;"
      data-token="<token>"
      data-server="https://api.bunchball.com"
      data-user-id="<userId>"
      data-balance-title='Award Account'
      data-editable-items='image'
      data-hide-level-progress='false'
      data-navigation='new tab'
      data-point-category='Points'
      data-language='<locale code>'
      data-translation-reference="ExampleTranslations"
      data-custom-css=''>
    </bunchball-profile>
  </div>
  <div class="col">
    <bunchball-events style="display:block; width:400px; height:450px;"
      data-token="<token>"
```

```
          data-server="https://api.bunchball.com"
          data-user-id="<userId>"
          data-export-navigation='false'
          data-return-count='5'
          data-language='<locale code>'
          data-translation-reference="ExampleTranslations"
          data-custom-css=''>
        </bunchball-events>
      </div>
      <div class="col">
        <bunchball-events style="display:block; width:400px; height:450px;"
          data-token="<token>"
          data-server="https://api.bunchball.com"
          data-user-id="<userId>"
          data-export-navigation='false'
          data-group-class='Brand'
          data-return-count='5'
          data-language='<locale code>'
          data-translation-reference="ExampleTranslations"
          data-custom-css=''>
        </bunchball-events>
      </div>
    </div>
  </div>
</body>
</html>
```

Translation Keys

Web Components

The translation key includes every string that's needed for a full translation of web components into a target language. This information will change as web components are updated and new text is introduced. Note that tokens like “{{count}}” should not be translated.

Work with your account team to ensure you have the current translation key.

Link to a Mission From an External Source

The following steps can be used when you want to link to a specific mission from an external source such as an email or web banner. When a user clicks the external link, they will open that mission directly in the block or web component without having to search for it.

Missions Web Component

1. Get the URL string to the page hosting your Missions web component including the specific mission ID. For example, https://.../mymissionpage.mydomain.com?missionid=123.

2. In the web app hosting the page where your Missions web component is embedded, set it to parse the data-mission-id parameter to a variable.

3. In the Missions web component embed code, set the data-mission-id parameter to use the variable from step 2. For example, data-mission-id={mission_var}.

4. From your external source (web banner, email, etc.), link to the mission using the URL in step 1.

Nitro Tokens

Tokens are placeholders that are replaced by the identifier they represent at runtime. Nitro supports the following tokens:

Availability	Token	Replaced at runtime with...
Within mission action phrases	**{username}**	the user's first and last name (if set with user preferences). Replaced with first name + space + last name.
	{challengeName}	the mission's display name. If no display name is set, uses the mission name.
	{firstName}	the user's first name
	{groupName}	the group name
	{groupShortName}	the group short name

Availability	Token	Replaced at runtime with...
	{lastInitial}	the first letter of the user's last name. Use with {firstName} to display the user's name as first name last initial. For example, Jane D.
	{lastName}	the user's last name
	{points}	the points awarded with the mission. If there is more than one point reward for a mission, the token will use the points for the award account point category. If there is no award account point reward, it will use the default point category.
	{userId}	the user's ID
	{userlinkUrl}	the userlink URL (if set in site settings)
	[ ]	If a URL is specified for the mission, it's used as the hyperlink for the part of the action phrase enclosed between square brackets. Show me an example
Within level action phrases	**{username}**	the user's first and last name (if set with user preferences). Replaced with first name + space + last name.
	{firstName}	the user's first name
	{lastName}	the user's last name
	{lastInitial}	the first letter of the user's last name.
	{levelName}	the new level name
	{userId}	the user's ID
Within recognition action phrases	**{recipientName}**	the recipient user's first and last name (if set with user preferences)
	{giverName}	the giver user's first and last name (if set with user preferences)
	{categoryName}	the recognition category name
	{points}	the number of points awarded for the recognition
	{comment}	the free-form comment added with a recognition submission

Disclaimer

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.

1/13/2025

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