Foundry Content Embed Implementation Guide

This guide includes the technical and operational logistics required to implement Foundry content Embed.

Introduction

Everfi courses, modules, and sequences can be included in partner websites using simple JavaScript and HTML. After including the JavaScript on your site, Everfi injects an iFrame window with your content into the page. Visit the Everfi Embed demo site to see an example of how this might look and work.

Everfi content will appear in the language the learner's browser is set to, unless not supported, in which case the content will appear in English.

Getting Started

To get started Everfi's implementation team will need to set up your account with your programs, playlists, and content, and enable the embed capability for your account. They will provide you with an admin account to allow you to log into the customer portal and set up your domains on which you will embed content. Lastly, once your account is set up with the appropriate content, Everfi will provide you with a list of variables that you will need to pass through your javascript.

Adding Domains to the Allowlist

Foundry will permit its content to be embedded only after you add your domains to the allowlist in Foundry.

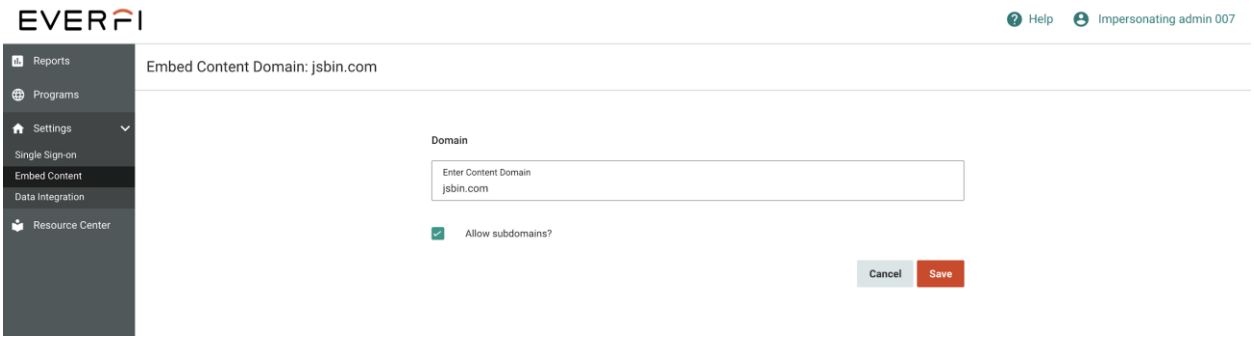
To add your domains to the allowlist, log in to the customer portal as an admin user. Navigate to **Settings** → **Embed Content**, then add the domain(s) on which you will embed Everfi content.

The domain input value format must either be a root name (no http/s protocol, URI paths, parameters, etc.) with a top-level domain (TLD) or a subdomain with the name and TLD. This format will validate upon saving. The "allow subdomains?" checkbox permits content embedding for wildcard subdomains.

Some Examples:

- A domain value of "fidev.net" would permit content embedding at "www.fidev.net" and "fidev.net" (www is automatically included by the backend because it is a common configuration for most domains)

- A domain value of “only-here.fidev.net” with “allow subdomains?” unchecked would permit embedding at “only-here.fidev.net”
- A domain value of “fidev.net” with “allow subdomains?” checked **true** would permit embedding at “foo.fidev.net”, “bar.fidev.net”, “anything.fidev.net”, etc.
- When an allowlist record is set up successfully, any URI path will work for embedding. E.G. “fidev.net/foo”, “fidev.net/bar”



JavaScript Implementation

On the web pages where you want to embed content, include https://everfi-next.net/embed_script.js in a script tag with:

- A unique id such as “everfi-program-content-iframer”
- And the src set to https://everfi-next.net/embed_script.js

```
<script id="everfi-program-content-iframer"
    src="https://everfi-next.net/embed_script.js">
```

If you have multiple embeds on the same page, then you should give each one a different and unique id. More examples that demonstrate how the script should be configured in HTML pages to successfully embed content are included below.

Parameters

A set of required parameters must be passed to the script so that the correct content is loaded in the iFrame. In addition to these required parameters, there are a set of optional parameters that can be used to set functions for certain click events as well as to modify the appearance of the iFrame. The required parameters will be provided to you by the Everfi implementation team.

- **account** [required] - text value used to identify the partner account. It is the subdomain in the partners Achieve Organization URL. For example: **account-slug**.everfi-next.net/student/dashboard/program-slug/playlist-slug/content-slug
- **program** [required] - text value used to identify the program. It is usually the third URI fragment in the partner's Achieve program URL, for example: account-slug.everfi-next-dev.net/student/dashboard/**program-slug**/playlist-slug/content-slug
- **playlist** [required] - text value used to identify the playlist. It is usually the fourth URI fragment in the partner's Achieve program URL, for example: account-slug.everfi-next-dev.net/student/dashboard/program-slug**/playlist-slug**/content-slug
- **module** [required] - text value used to identify the content. It is usually the fifth URI fragment in the partner's Achieve URL, for example: account-slug.everfi-next-dev.net/student/dashboard/program-slug/playlist-slug/**content-slug**
- **script_id** [optional] - The html ID value set on the script tag. This should be used when multiple embed scripts are in a single page.
- **locale** [optional] - to force the language the module will appear in, use this parameter with the value equal to 'es' for Spanish, or 'zh-CN' for Chinese. Not all modules support all languages. Otherwise, the module will appear in the language of the learner's browser, unless not supported, in which case the content will appear in English.
- **height** [optional] - integer value to control the embed height in pixels. Be aware that separate from this parameter property, you might want to set a height property for the on your website that will contain the embedded content. See the Tips section for more.
- **width** [optional] - integer value to control the embed width in pixels
    - If height / width are omitted, the containing element the script tag is inside of should be given some sizing to ensure that the iFrame is visible
- **expand** [optional] - You can use the *expand* parameter to specify whether the iFrame should expand to fit and if a scroll bar should appear. The options are the following:
    - *false* - If it's not specified, by default 'expand = false'. The embed script will determine a fixed height based on its container and add a scrollbar if the container is not tall enough to fit the module.
    - *true* - This option is for backwards compatibility. The embed script will expand the container to the full height of the module.
    - *no-scroll* - The embed script will expand the container to the full height of the module. This option ensures that the iFrame will not have its own scrollbar.
- **onExit** [optional] - function or name of a function that will get called when a user clicks the **exit** hyperlink in a module.

- **onPageLoad** [optional] - function or name of a function that will get called when a user navigates to a new page within the module.

Customer Specific Parameters

It is possible to pass along additional parameters that are specific to your program, to be recorded at the time the learner first clicks on a CTA within the embedded content. To accomplish this, your Everfi Implementation Specialist will need to create a custom question with a slug, which they will then provide to you. Include the following show parameter set to 'registration' and the slug with the value set to anything of your choosing to start collecting data.

These parameters can be used for anything. Some examples include sending along a user ID, information about the referral site, or user type. The answers are recorded in the *User Data Report* accessible to the Customer Admin.

- **show** - The value of this parameter should always be 'registration'.
- **cq_slug** - a dynamic key used to identify a program question. The value will be a free text answer to save as the response.

Examples

These examples have a custom slug to send along and record a user ID.

- Data Parameter Method
- Object Parameter Method
- Query Parameter Method

Methods to pass parameters

The Everfi script also needs to have parameters (params) passed to it to display the correct content with the right configuration. Multiple passing procedures can be used in conjunction with one another, but there is a priority so that duplicate parameter values are overwritten in an expected way.

There are 3 different ways to pass the params:

- Configuration Object
    - The object must be assigned to a global variable called everFiProgramContentIframeOptions.
    - Highest Priority. It will overwrite parameters specified any other way.

- Data Attributes
    - Second Highest Priority. It will be overwritten by parameters in the configuration object and only overwrite parameters specified via query string.
- URL Query String
    - Lowest Priority

Object Params Method

The parameters are added to an object named everFiProgramContentIframeOptions. This option allows for adding inline functions for *onExit* and *onPageLoad*.
Note: The following is an example to show how to pass parameters. It is not definitive way to implement this code.

Example: https://embed.s3.amazonaws.com/sandbox/ObjectParams_sandbox.html

```
        <script>
        function onPageLoadFunction(container) {
            container.scrollIntoView();
            console.log("Module Page has been changed");
        }
        function exitMessage() {
            alert('done!!');
            console.log('Module exited');
        }
        </script>

        <script>
            everFiProgramContentIframeOptions = {
                account: 'embed-code-demo',
                program: 'embed-code-demo-program',
                playlist: 'embed-code-demo-my-playlist',
                module: '88',
                script_id: 'everfi-program-content-iframer-obj',
                onExit: 'exitMessage',
                onPageLoad: 'onPageLoadFunction',
                expand: 'no-scroll',
            }
       </script>
        <script id="everfi-program-content-iframer-obj"
            src="https://everfi-next.net/embed_script.js?script_id=everfi-
program-content-iframer-obj">
        </script>
```

Data Params Method

The parameters are passed as data attributes within the script.
Note: The following is an example to show how to pass parameters. It is not definitive way to implement this code.

Example: https://embed.s3.us-east-1.amazonaws.com/prod/DataParams_prod.html

```
<script>
function onPageLoadFunction(container) {
    container.scrollIntoView();
    console.log("This fires when I change pages in the module");
}
function exitMessage() {
    alert('done!!');
    console.log('We have hit exit');
}
</script>


<script src="https://everfi-next.net/embed_script.js"
    id="everfi-program-content-iframer"
    data-account="agiledesignlabs"
    data-program="strength"
    data-playlist="achieve"
    data-module="554"
    data-script_id="everfi-program-content-iframer-params"
    data-on-exit="exitMessage"
    data-on-page-load="onPageLoadFunction"
    data-expand="no-scroll">
</script>
```

Query String Params Method

The params are passed as query string parameters to the *src* URL. The URL is followed by a '?' and each set of parameters is separated by an '&'.
Note: The following is an example to show how to pass parameters. It is not definitive way to implement this code.

Ex. https://embed.s3.amazonaws.com/prod/QueryParams_prod.html

```
<script>
function onPageLoadFunction(container) {
    container.scrollIntoView();
    console.log("This fires when I change pages in the module");
}
function exitMessage() {
    alert('done!!');
    console.log('We have hit exit');
```

```
        }
        </script>
        <script id="everfi-program-content-iframer"
            src="https://everfi-next.net/embed_script.js?account=embed-code-
demo&program=embed-code-demo-program&playlist=embed-code-demo-my-
playlist&module=556&onExit=exitMessage&expand=no-
scroll&onPageLoad=onPageLoadFunction&script_id=everfi-program-content-
iframer">
        </script>
```

A Deeper Look at Optional Parameters

The optional parameters are available to help you control the display and functionality of the embedded content.

- **locale** - set the language
- **height** - sets the height of the embedded iFrame
- **width** - sets the width of the embedded iFrame
- **expand** - specifies whether or not the iFrame should expand to fit and if a scroll bar should appear
- **onExit** - sets the function to be used when the 'Exit' button is clicked
- **onPageLoad** - sets the function to be used when a new 'page' within the module is loaded

Using Parameters to Adjust the Look of the iFrame

The *height*, *width*, and *expand* parameters may all be used to help achieve the desired size of the iFrame and the scroll bar. Even when using these parameters, it is a best practice to set a height on the enclosing div (See Tips for more) to allow the iFrame to load more smoothly.

Using Parameters to Set the Language - Locale

If a learner has their browser set to a particular language, and the learning content is available in that language, the content should automatically appear in the preferred language. If that language is not available, the content will appear in US English.

To force the module to appear in a particular language, use the 'locale' parameter. Your implementation specialist can provide you with a list of modules and the locales they are available in.

Examples

- Data Parameter Method
- Object Parameter Method
- Query Parameter Method

Using Parameters to Set the onExit Function

Much of our learning content includes an 'Exit' button/link in the navigation or on the last page of the module. By using the 'onExit' parameter you can set a function that will fire whenever the 'onExit' event occurs. The function could do any number of things such as redirect the user to the main page, open a survey, close the modal, etc.

Using Parameters to Set the onPageLoad Function

The 'onPageLoad' event fires any time a "page" in the content is loaded. A page here does not necessarily mean a new URL within the iFrame but is when the learner navigates to a what appears to be a new page via the navigation menu or navigation buttons within the content. You could set the 'onPageLoad' function to do any number of things. One example use case is if the iFrame has been embedded directly onto the web page to show at full height. In this scenario, if a user has scrolled down to the module, a function could be set to 'onPageLoad' that would bring the Embed's container back into view. Otherwise, the learner would have to manually scroll to the top.

Analytics & Tag Managers

There are a couple of options for collecting analytics on your Embed program. One option is to integrate your Google Analytics account into your Embed program. You will provide your Implementation Specialist with your Google Analytics ID and they will add it to your program. In addition, you can also choose to implement Google Analytics Linker to provide better cross-domain measurement. If you choose to use a conversion linker you must include the program param in the script URL. Therefore, it may be easiest to use the Query param method of implementation.

Another option is to add your organization's tag management system to your Foundry account. We currently support Google Tag Manager, TealiumIQ, and Adobe Launch. Your Implementation Specialist can provide you with more information.

When a tag management system is added to your account, depending on how the tag management system is configured, cookies may be used to collect information about the learner. With the embed experience, there is no banner asking the learner for consent to gathering information.

Tips

For a better iOS experience, add this style to the element the embed is inside of: -webkit-overflow-scrolling:touch as explained in IFRAMEs and the Safari on the iPad, how can the user scroll the content? | Stack Overflow

When the embed script is included on an unauthorized domain, Everfi's JavaScript may continue to poll our servers to check for newly allowlisted domains or subdomains. In certain browsers, this causes the parent div to grow with each poll response. To avoid this scenario, set an explicit height on the parent div. Alternatively, you could set an optional parameter in the javascript expand=false. A best practice would be to make the height be a percentage of the viewport. Without constraining the height, the screen could grow too tall.