

# Web Monetization API for Unity WebGL

This project allows you to register a web monetization `meta` tag to your Unity WebGL game so that you can make money with your game on the web. It also allows you to create an in-game benefit to players who are monetizing your game.

The [Web Monetization API](#) is a proposed [W3C standard](#). There is currently just one Web Monetization provider, [coil.com](#).

This project is provided for free by [SIMMER.io](#), because we want you to be able to make money with your web games whether or not you use our platform. If you have a cool game to share, consider [uploading it](#) to our site!

## Video Tutorial

Probably the easiest way to learn this is to watch [our tutorial](#) on Youtube.

## Getting the Package

If you're viewing this on Github, you can get the latest release Unity Package in `Releases/`. If you're on the Unity Asset Store, simply download and import the asset on your project.

## Quickstart

You might want to first read [Writing a Web Monetized Game](#) for an overview of a simple explanation of monetizing a game.

You can also export `SampleScene/WebMonetizationSampleScene` to WebGL. If you have a Web Monetization browser plugin (Coil - [Chrome](#) | [Firefox](#)) and a paid account, you'll see a coin jar filling up with coins. If you don't have the extension or a paid account, you can click the "Simulate Monetization" button in the game to see what it's supposed to look like.

Add monetization to your game:

1. Add `Prefabs/WebMonetizationBroadcaster` to your scene
2. Get a creator account at [coil.com](#) and set the `Payment Pointer` in the `WebMonetizationBroadcaster` properties window

Recommended: Add player benefits when monetization events occur:

1. Add the sample code from `Scripts/WMReceiver.cs` to a `GameObject` monobehavior script. (copy/paste `OnEnable`, `OnDisable`, `OnMonetizationStart` and `OnMonetizationProgress` to your `GameObject` that will do something special when monetization events occur).
2. Add your player benefits in `OnMonetizationProgress` or `OnMonetizationStart`.
3. [Export to WebGL](#), and test.

Simulate from Unity:

1. You can turn on `Simulate Monetization` in the `WebMonetizationBroadcaster` properties window (Don't forget to turn it off when you deploy your game to the web).

2. You can also call `WebMonetizationBroadcaster.StartSimulation()` from a UI click handler (button).

Simulate from Browser:

1. I recommend getting a paid coil (\$5/mo) account for testing (so that you can support other creators). If that's not an option, I've also put up a tool for simulating monetization events from the browser. Available at [testwebmonetization.com](https://testwebmonetization.com). Search for "bookmarklet".

## Details

Here are details of the events that you can receive from the `WebMonetizationBroadcaster` :

### Register and Unregister Events

```
// register one or both events if you want to use them
void OnEnable()
{
    WebMonetization.OnMonetizationStart += OnMonetizationStart;
    WebMonetization.OnMonetizationProgress += OnMonetizationProgress;
}

// unregister events that you've registered
void OnDisable()
{
    WebMonetization.OnMonetizationStart -= OnMonetizationStart;
    WebMonetization.OnMonetizationProgress -= OnMonetizationProgress;
}
```

### Act upon those events

You might just want to provide a general benefit to the player when you know they're monetizing. In that case you could just set a boolean to true in the first time `onMonetizationProgress` is called. I would discourage use of `onMonetizationStart` because it won't be called if you load another scene. So my recommendation is to always use `onMonetizationProgress`.

Here is the code you'd add to your `GameObject` script for each of these handlers:

### OnMonetizationStart

```
void OnMonetizationStart(Dictionary<string, object> detail)
{
    // these are the parameters that you can read from the detail dictionary.
    // recommended: wrap parsing of each of these values in a try/catch in case the
    spec changes.
    // https://coil.com/docs/#browser-start

    // string requestId = detail["requestId"] as string;
    // string id = detail["id"] as string;
    // string resolvedEndpoint = detail["resolvedEndpoint"] as string;
    // string metaContent = detail["metaContent"] as string;
```

```

    // Debug.Log("MonetizationStart requestId: " + requestId + ", id: " + id + ",
resolvedEndpoint: " + resolvedEndpoint + ", metaContent" + metaContent);

    Debug.Log("MonetizationStart");
}

```

## OnMonetizationProgress

The `detail` object here provides additional data, such as `amount`, `assetCode` (USD, XRP, etc), and `assetScale`. You could provide a certain amount of in-game currency as payments trickle in using these parameters. See [Coil Developer Docs](#) for details.

```

// A monetization progress event should occur roughly every two seconds after the
monetization progress occurs
void OnMonetizationProgress(Dictionary<string, object> detail)
{
    // these are the parameters that you can read from the detail dictionary.
    // recommended: wrap parsing of each of these values in a try/catch in case the
spec changes.
    // https://coil.com/docs/#browser-progress

    // string amount = detail["amount"] as string;
    // long amountAsLong = Convert.ToInt64(amount);
    // string assetCode = detail["assetCode"] as string;
    // long scale = (long) detail["assetScale"];

    // Debug.Log("MonetizationProgress amount " + amountAsLong + ", assetCode: " +
assetCode + ", scale: " + scale);

    Debug.Log("MonetizationProgress");
}

```

The `detail` dictionary is built with [MiniJSON](#). Read the comments on top of that file for more details on parsing the dictionary.

## A note about deployment

Many game portals (itch.io, kongregate, etc) run your game in an `iframe`. Web monetization does not automatically work in `iframe`'s. On [SIMMER.io](#), we bubble up the web monetization `<meta>` tag to the main frame, and bubble down the monetization events so that you can get paid, and can show cool stuff to your user.

If this is something you want, I'd recommend reaching out to the operators of other game portals and request that they add code to support this type of monetization.

## Support

Please fill out a form at <https://simmer.io/support> for help. I'd appreciate you reaching out for help vs leaving a poor review on the asset store. I really do respond to most requests in 24 hours unless I'm traveling.

Thanks

-Rocco

## Contributing

Github pull requests are always appreciated if you find a bug. In the github project there's also a file:

`CONTRIBUTING.md` that covers some things for internal development of this asset.

## Thank You!

If you found this useful, please leave an asset store review or "Star" on Github. Every little bit makes a big difference! Also consider sharing your game at <https://simmer.io/upload>. It's truly drag and drop!

Also, you can [follow us on twitter](#).

## License

The project was developed by SIMMER.io, and you can use it in your projects for free. If downloaded from Github it is released under the MIT License. If downloaded from the Unity Asset Store it is licenced under the [Unity Asset Store EULA](#). Credit is always nice, but not required!