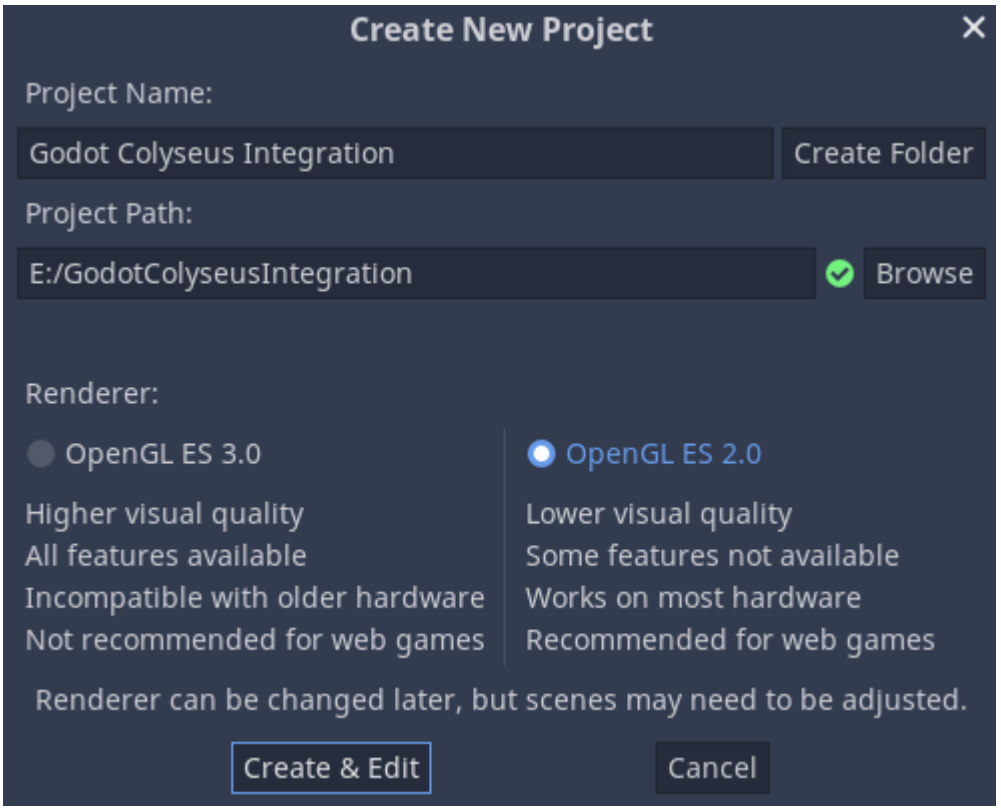
Godot Colyseus Integration

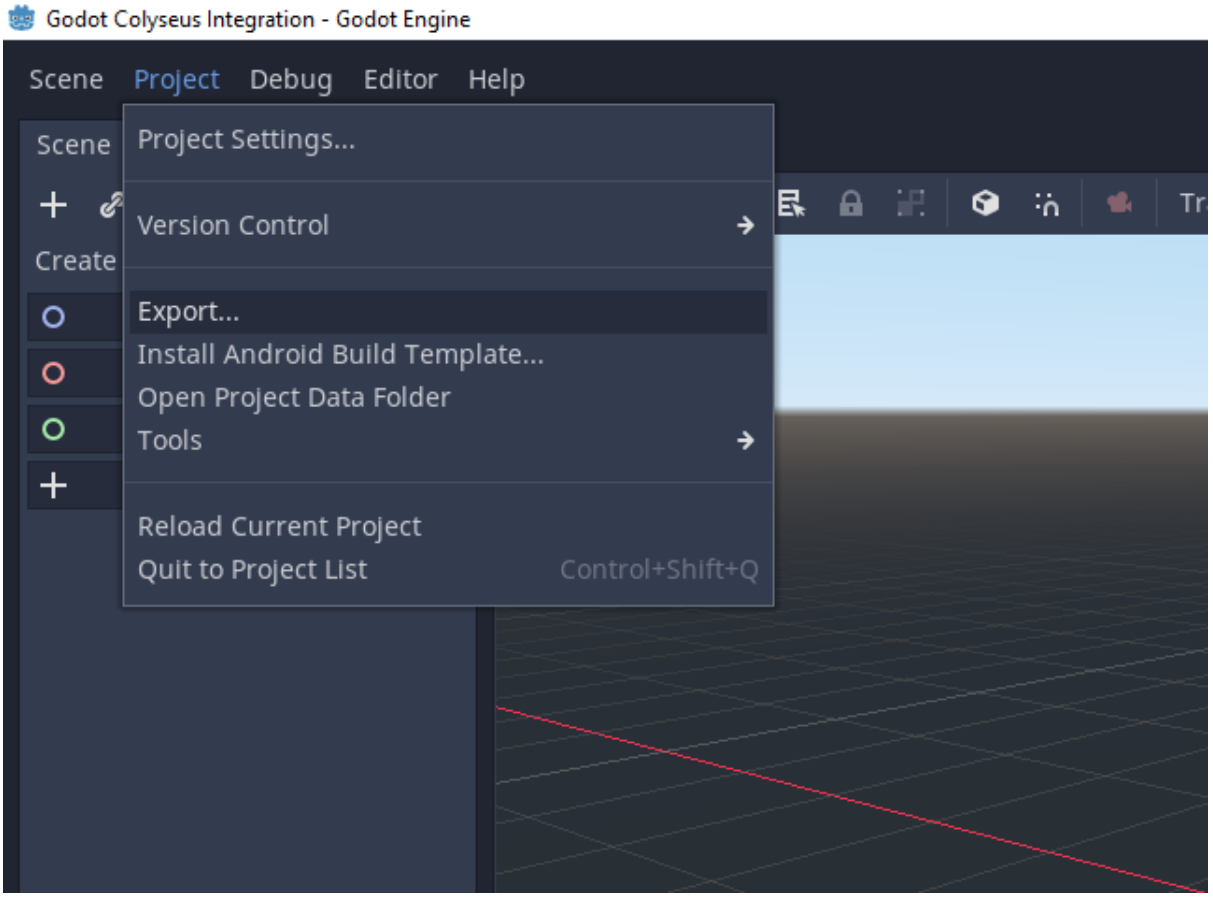
Create a new Godot project

Open up Godot and create a new project with the renderer being set to OpenGL ES 2.0 as it is recommended for web games.

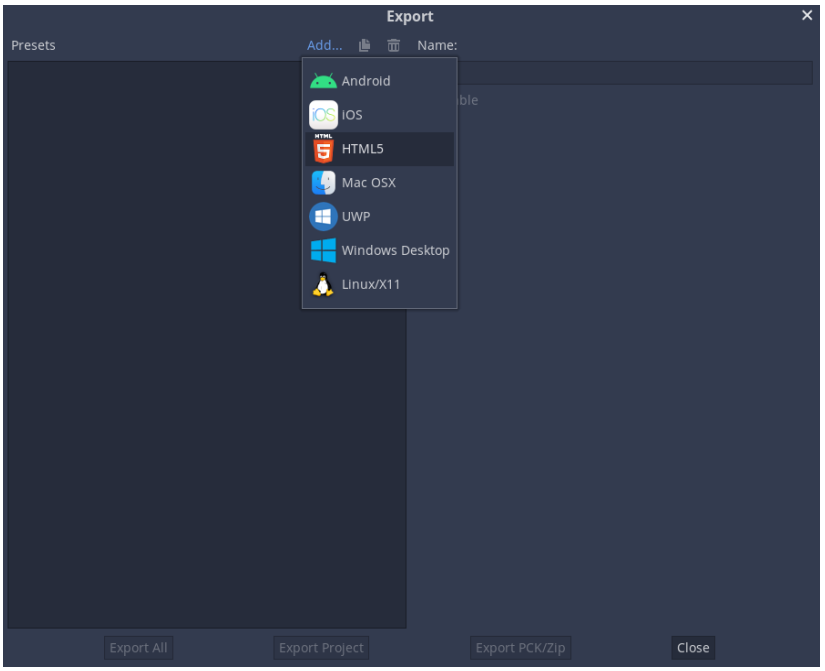


Create an Export Preset

From the toolbar in the upper left side of the editor, go into **Project** > **Export**.

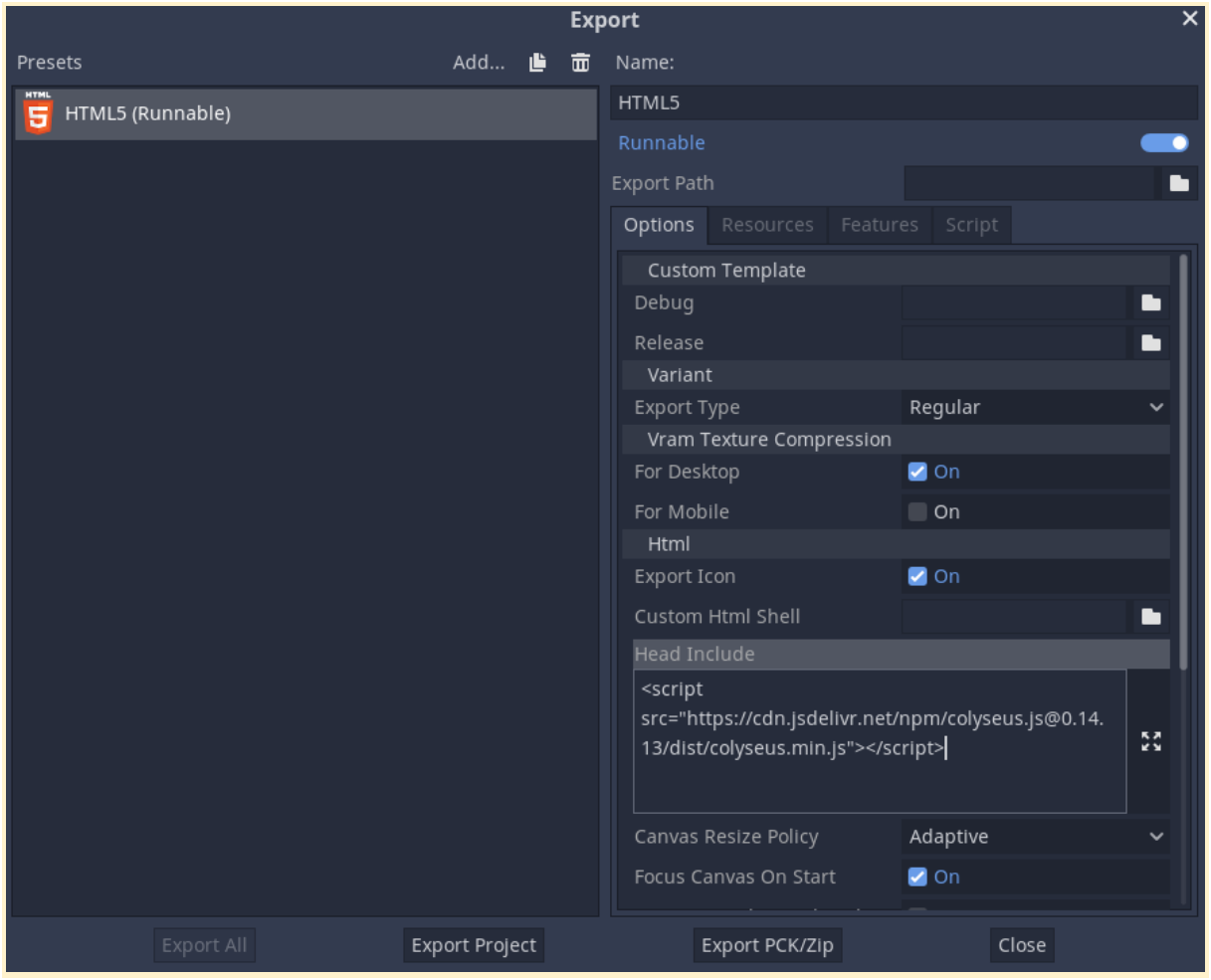


Then, add a new HTML5 export preset.

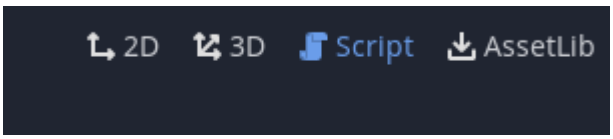


In the export preset settings, set the **Head Include** to:

```
<script src="https://cdn.jsdelivr.net/npm/colyseus.js@0.14.13/dist/colyseus.min.js"></script>
```



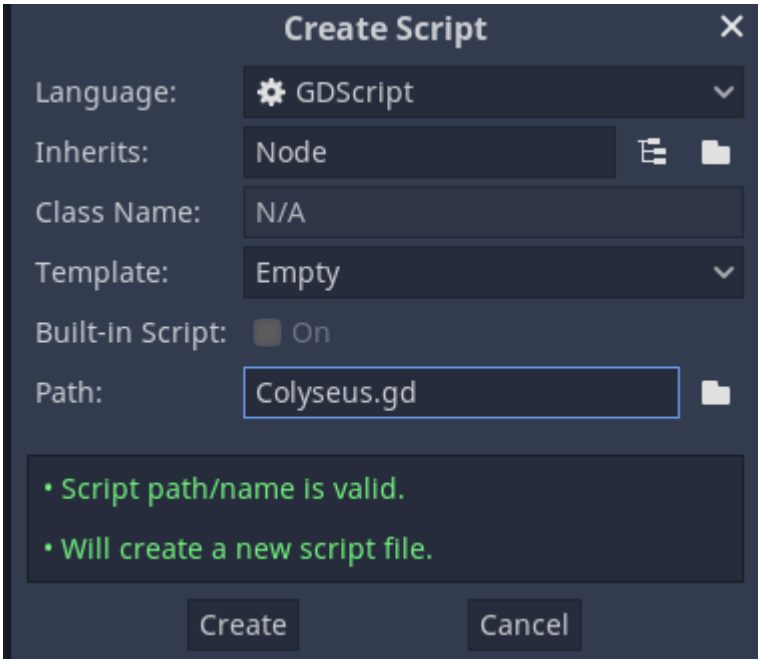
Create a Colyseus AutoLoad



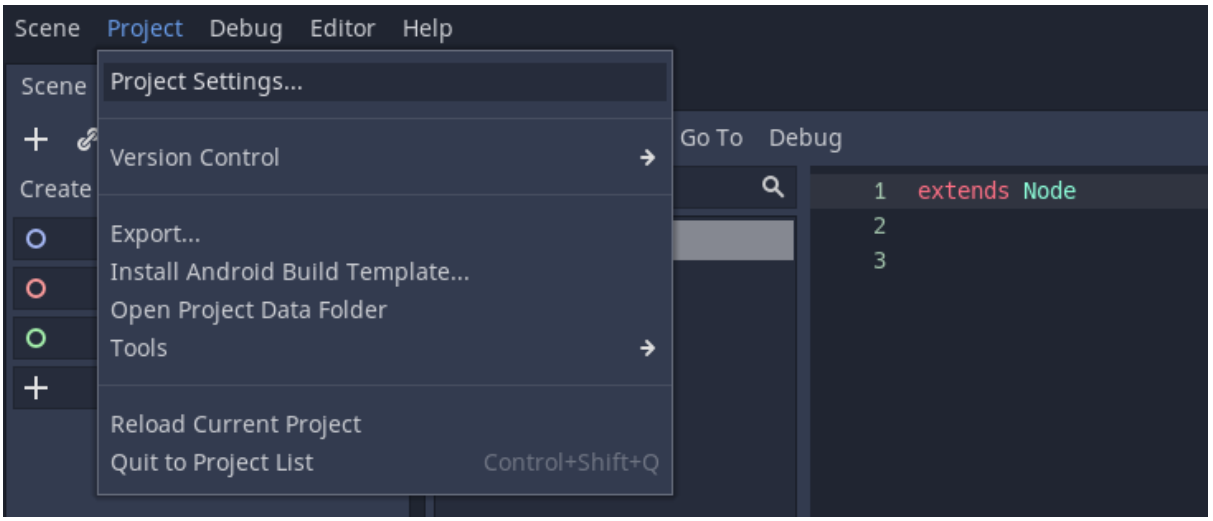
From the toolbar at the top of the editor, go into the script editor.

In the script editor, select **File** > **New Script**

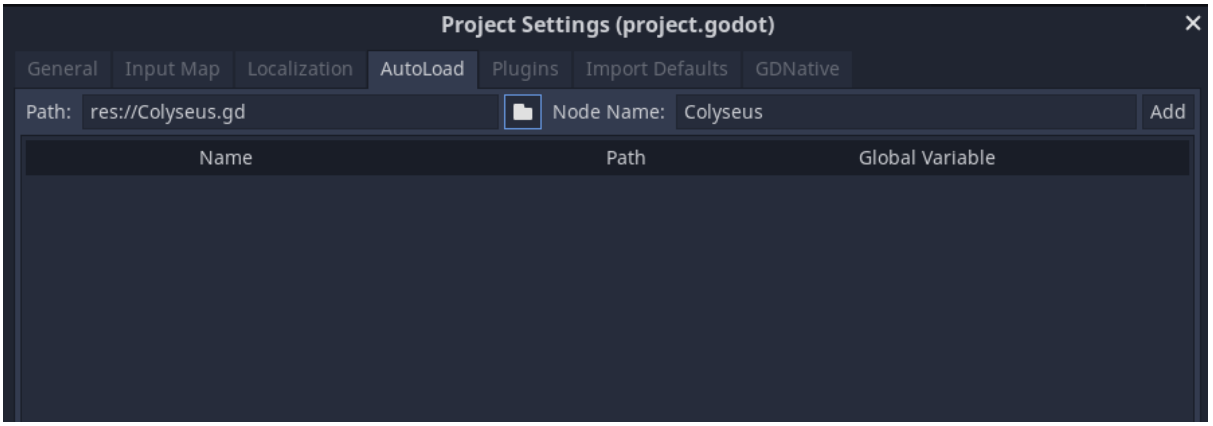
Name the script however you want. For the sake of this tutorial, we'll just name it Colyseus.



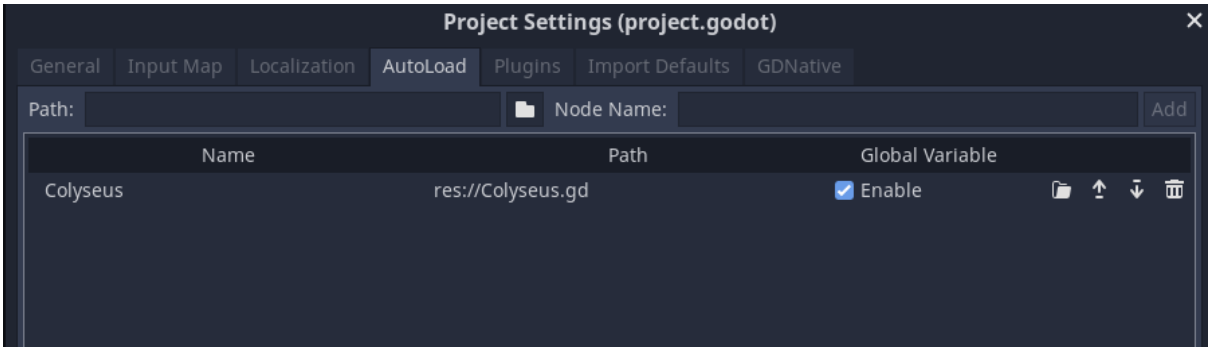
Go into the **Project** > **Project Setting**s using the upper left side toolbar.



In the **AutoLoad** section, select a path to your script and press **Add.**



Make sure your new AutoLoad is enabled.



Coding the AutoLoad

Inside the AutoLoad, create variables for the Colyseus interface, Colyseus client, Colyseus room and for the address of the Colyseus server.

```
var colyseus_interface = JavaScript.get_interface("Colyseus")
var colyseus_client: JavaScriptObject = null
var colyseus_room: JavaScriptObject = null
var colyseus_address: String = ""
```

```
3  var colyseus_interface = JavaScript.get_interface("Colyseus")
4  var colyseus_client: JavaScriptObject = null
5  var colyseus_room: JavaScriptObject = null
6
7  var colyseus_address: String = ""
```

Now, let's write a function that will create a Colyseus client for us.

```
func create_colyseus_client() -> void:
    if OS.has_feature("JavaScript"):
        # Initialize the Colyseus Client

        JavaScript.eval("window.colyseus_client = new Colyseus.Client('" +
colyseus_address + "');", true)
        colyseus_client = JavaScript.get_interface("colyseus_client")
```

```
11 # Creates a colyseus client @ colyseus_address
12 func create_colyseus_client() -> void:
13     if OS.has_feature("JavaScript"):
14         # Initialize the Colyseus Client
15
16         JavaScript.eval("window.colyseus_client = new Colyseus.Client('" + colyseus_address + "');", true)
17         colyseus_client = JavaScript.get_interface("colyseus_client")
18
```

The function creates a Colyseus client at the given colyseus_address, however there is no room created yet! To create a room, we will add another variable at the top of our script.

```
var colyseus_room_name: String = "myroom"
```

```
8  var colyseus_room_name: String = "myroom"
```

And now the function:

```
func create_colyseus_room() -> void:
    if OS.has_feature("JavaScript"):
        var options = JavaScript.create_object("Object")
        options.player_name = "ColyseusPlayer"
        colyseus_client.joinOrCreate(colyseus_room_name, options)
```

```
func create_colyseus_room() -> void:
    if OS.has_feature("JavaScript"):
        var options = JavaScript.create_object("Object")
        options.player_name = "ColyseusPlayer"
        colyseus_client.joinOrCreate(colyseus_room_name, options)
```

This will create a Colyseus room with the given colyseus_room_name and provided options, for example a player name, that will be received by the server. However, we don't actually know in our code if the room creation was successful, so we have to create a callback for that.

Back to the top of our script, and add a line:

`var _callback_room_join = JavaScript.create_callback(self, "_on_room_joined")`

```
var _callback_room_join = JavaScript.create_callback(self, "_on_room_joined")
```

Now in the create_colyseus_room() function, add `.then(_callback_room_join)` to our joinOrCreate call.

```
# Creates a colyseus room @ colyseus_room_name
func create_colyseus_room() -> void:
    if OS.has_feature("JavaScript"):
        var options = JavaScript.create_object("Object")
        options.player_name = "ColyseusPlayer"
        colyseus_client.joinOrCreate(colyseus_room_name, options).then(_callback_room_join)

```

Colyseus will now call the _on_room_joined function in our script if you have joined a room successfully, so we need to create that function.

```
func _on_room_joined(args) -> void:
    colyseus_room = args[0]
```

```
# Room joined callback
func _on_room_joined(args) -> void:
    colyseus_room = args[0]
```

The first element of the array that is returned with the callback is the Colyseus room that we have joined, so we set it to our colyseus_room variable.

The last thing we need is a way to send messages to our Colyseus room. Let's create a function for that.

```
func send_to_colyseus(message_type: String, message) -> void:
    if colyseus_room:
        colyseus_room.send(message_type, message)
```

```
30 func send_to_colyseus(message_type: String, message) -> void:
31     if colyseus_room:
32         colyseus_room.send(message_type, message)
33
```

Now, for example, if we wanted to submit a player's score variable to our server that has a registered message type "submit-score", we would just call

```
Colyseus.send_to_colyseus("submit-score", score)
```

from any script.