SDK Generation

==============

Follow the same procedures as the main SDKGenerator to generate the Unreal Engine PlayFab SDK. Navigate to the output folder that was entered during the SDK generation.

SDK Folder Structure

====================

In the output folder you will see three folders labeled "PlayFabClientSDK", “PlayFabSDK”, and “PlayFabServerSDK”. The Client folder holds only the Client APIs. It also does not have any references to the API Secret Key to keep key safe when building the client. The Server folder holds all other APIs not including the Client API. Finally, PlayFabSDK holds all the API calls.

In these folders are two other folders; "ExampleProject", and "PluginFiles".

The example project folder is a project all ready to go with the plugin already copied into it. Depending on what SDK you use it will have the correct SDK loaded into the plugins folder.

The plugin files folder has another folder "PlayFab". The PlayFab folder is the folder that is dropped into an Unreal Engine project folder. It should be placed in "ProjectName/Plugins/"

Plugin Setup

============

There are a couple of steps to customize the plugin to work with your PlayFab account.

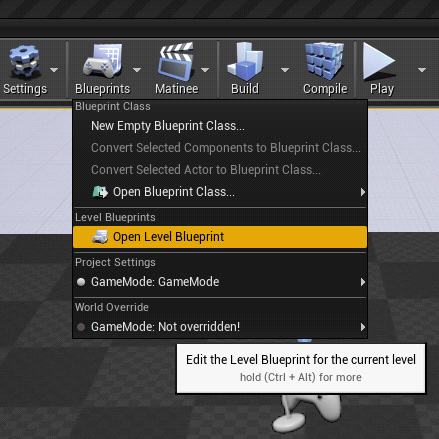
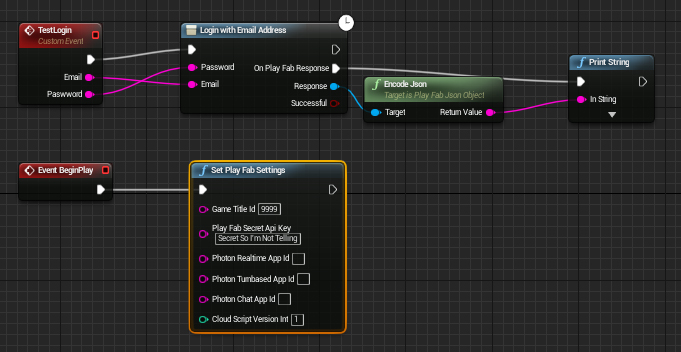
* In the project folder right click on Unreal Engine project file.
* Select "Generate Visual Studio Project Files".
* Open the Visual Studio file associated with the project.
* Now build out the project and continue down to plugin use.

Plugin Use

==========

The plugin is fairly simple to use. The main things you need to learn are; how to use the provided JSON objects, how to wire up the nodes, and realize that it is extremely simple!

Below are the steps to add an api call node, and manipulate the response JSON object. We will make a simple login with email function that will print out the response JSON object to the screen.

* Open the Unreal editor by double clicking on the project file.
* Got to the top bar and select "Blueprints" then "Open Level Blueprint" from the drop down menu.
* 
* To start, add a “Set Play Fab Settings” node found in “PlayFab/Settings”. Add the TitleId and any keys you need.
* Wire this node up to the “Event BeginPlay” node. You can wire the set settings node up to the beginning play wherever you wish, but this needs to be called before anything else.
* Now, right click on an empty spot and add an "Add Custom Event Node".
* Name it something you will remember to be called from the console manager.
* Now add two string variables named "Email" and "Password" to the custom event.
* Right click again and search for the "Login With Email Address" node located in "Play Fab/Client/Authentication".
* You can also type "login" in the search field to find it quicker.
* Now wire up the "Email" on the event to the "Email" on the api node. Do the same for the "Password".
* The response on the node is a JSON object that can be manipulated using the provided JSON functions located in "Play Fab/Json".
* Drag off from the response node and search for "Encode Json"
* Now wire the "Encode Json" return up to a "Print String" node. Below is an image of the complete system.
* 
* Now run the project and in the play in editor window hit the tilde ~ key. This will bring up the console manager.
* Now type "ce TestLogin "email@email.com" password"
* Notice you add quotes to email but not to password.
* Once you hit enter you should get the response printed out on the screen.

The ExampleProject including already has these functions setup in the level blueprint. Just open the project, go to the level blueprint.