



UNREAL ENGINE

PLAYFAB

Login Tutorial



TABLE OF CONTENTS

.....

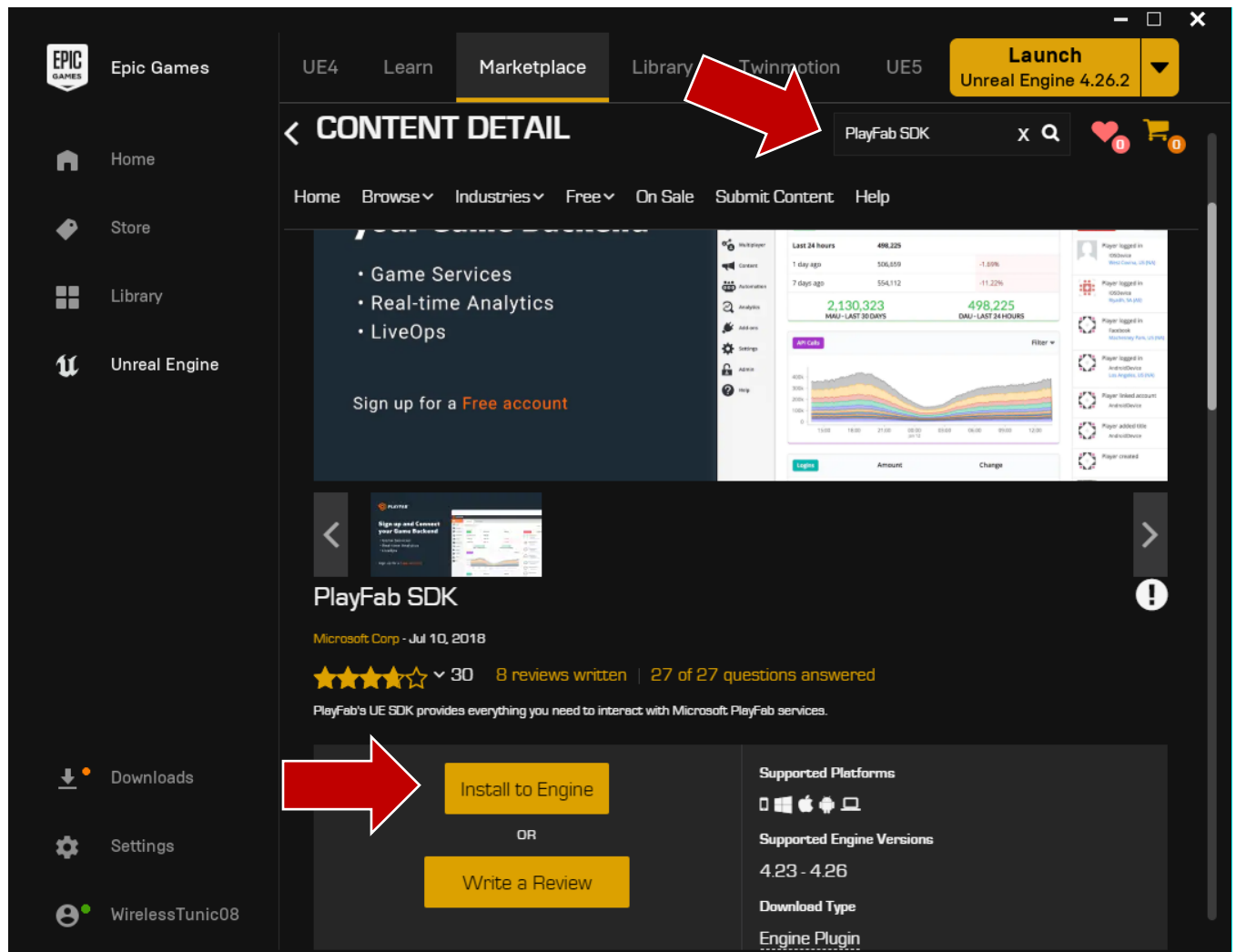
1.	How to install Playfab SKD	
2.	How to link project with playfab	
3.	how to make login/register UI.....	12
4.	Game instance Blueprint.....	
5.	Finishing touches	

This whole tutorial should take around 50 mins. Feel free to use your own ui assets as will not be adding any cosemtic features to my ui. Note that a little bit of unreal knowledge is needed prior to this tutorial. Made by Nathan Aruna

INSTALLING THE SKD

Firstly, head over to the “Epic Launcher” and navigate to the unreal section.

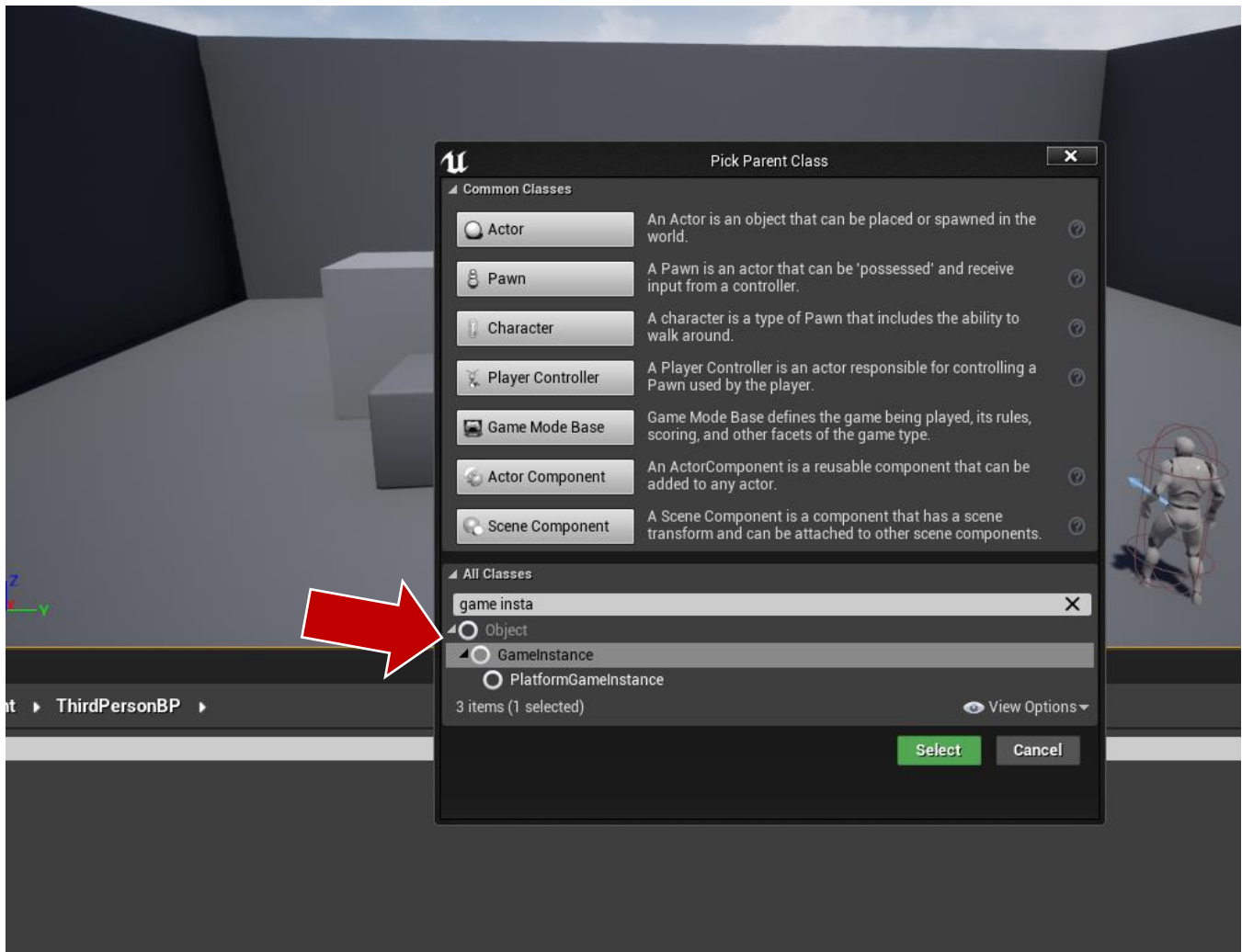
In the marketplace search up “PlayFab” and once you find it click install to engine if you already have Unreal open you will need to restart it for it to take effect.



Great the first step is complete if you haven't already open unreal again so we can continue to the next step

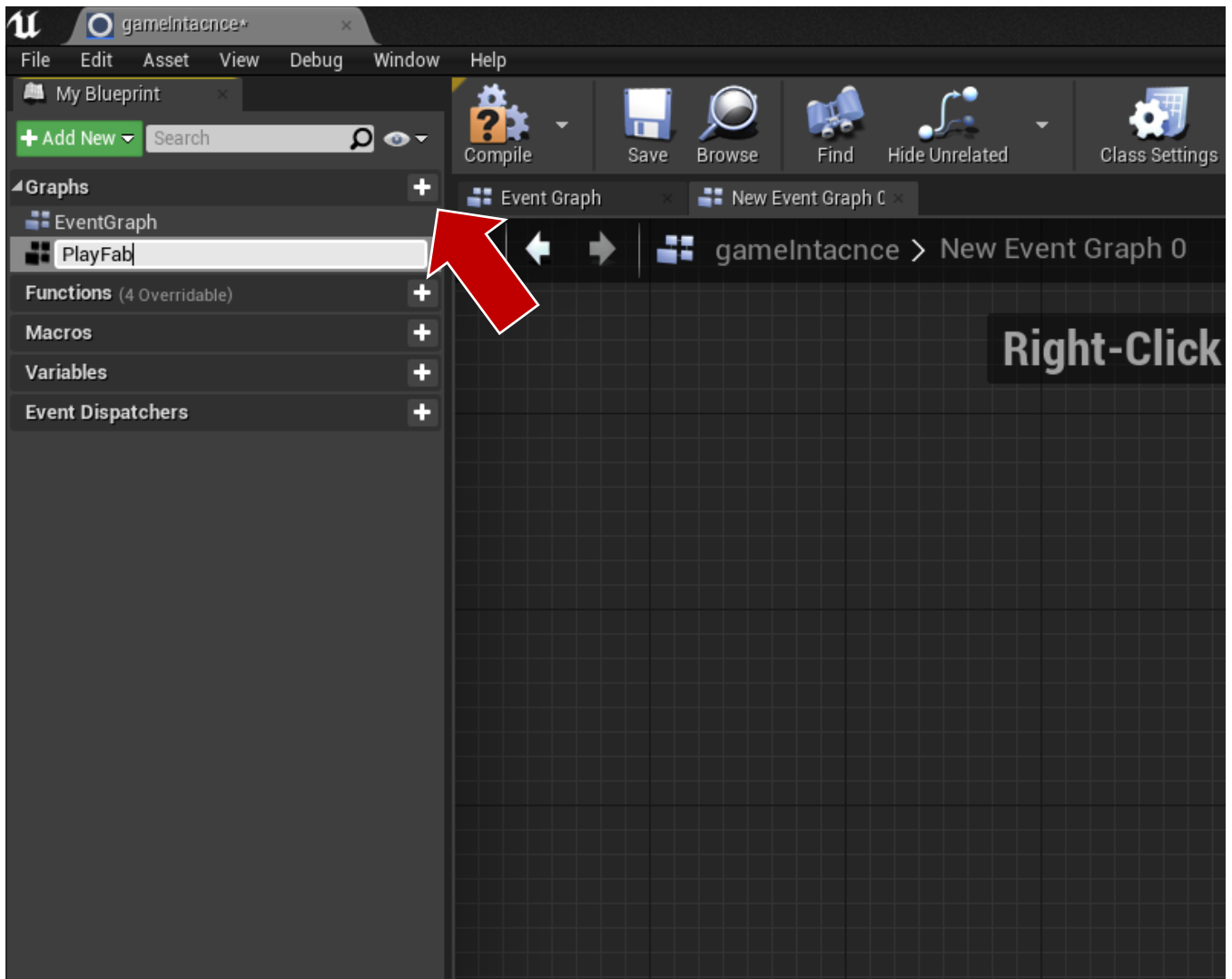
LINKING PLAYFAB WITH YOUR PROJECT

Now that that's out of the way we need to create a game instance where all our blueprints will be written in. Right click in the content browser and create a blueprint class then in "All classes" search "GameInstance" and select it to create a new one. Name it whatever you want, it's your game Afterall. Just remember the name because it will be used a lot.



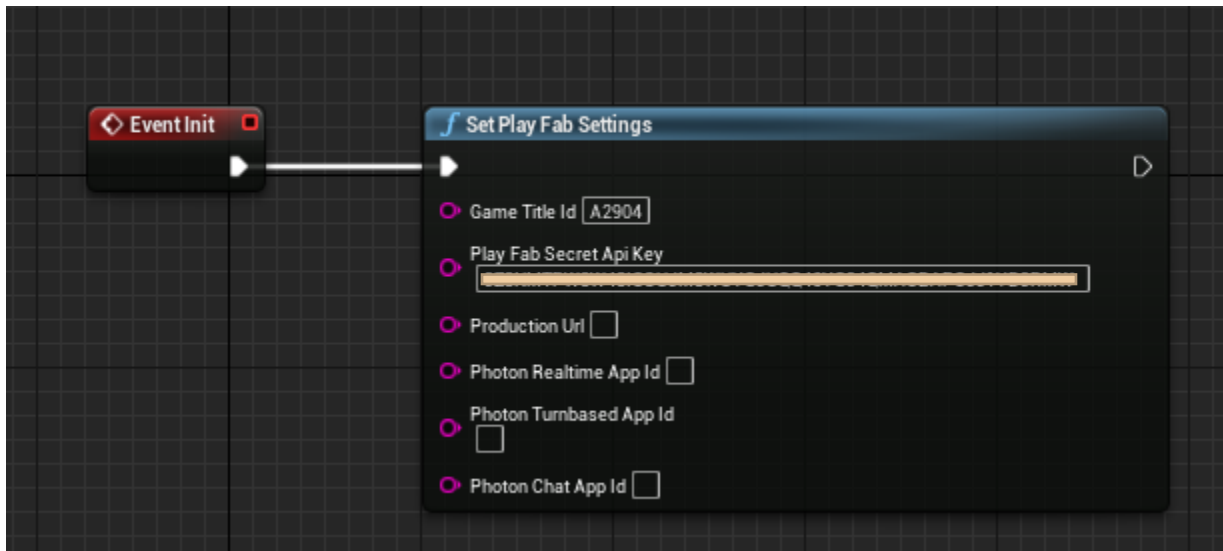
Inside of the of your new game instance look to the right you should see something that says "Graphs" click the little plus to make a new one and name it PlayFab. This is where all the Blueprints for PlayFab related things will be held.

Part 2

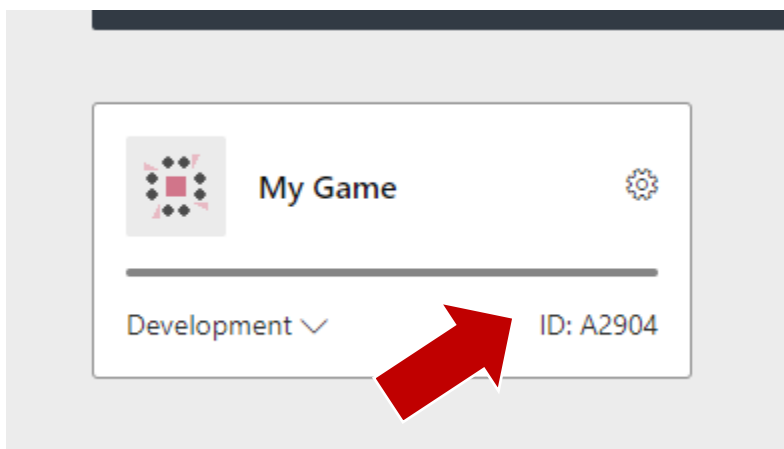


Within our new graph we want to link unreal with PlayFab. To do so right click and create a “Event Init” then off of that link it to “set PlayFab Settings”. If you installed the SKD correctly you should be able to find it. Your Blueprint should look something like this. I’m assuming you have already made a [PlayFab account](#) if not you will need one.

Part 2

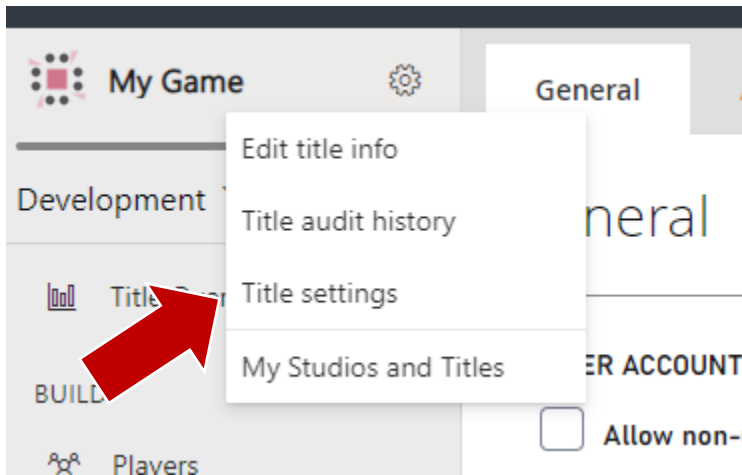


You will need two codes the “Game ID” and the “PlayFab Secret API key” Make sure you do not share this key with anyone otherwise they will have access to your game. Now you may ask where do I find these keys and id’s? No worries to find the go to the home screen of play fab and it should be there under the title of your game

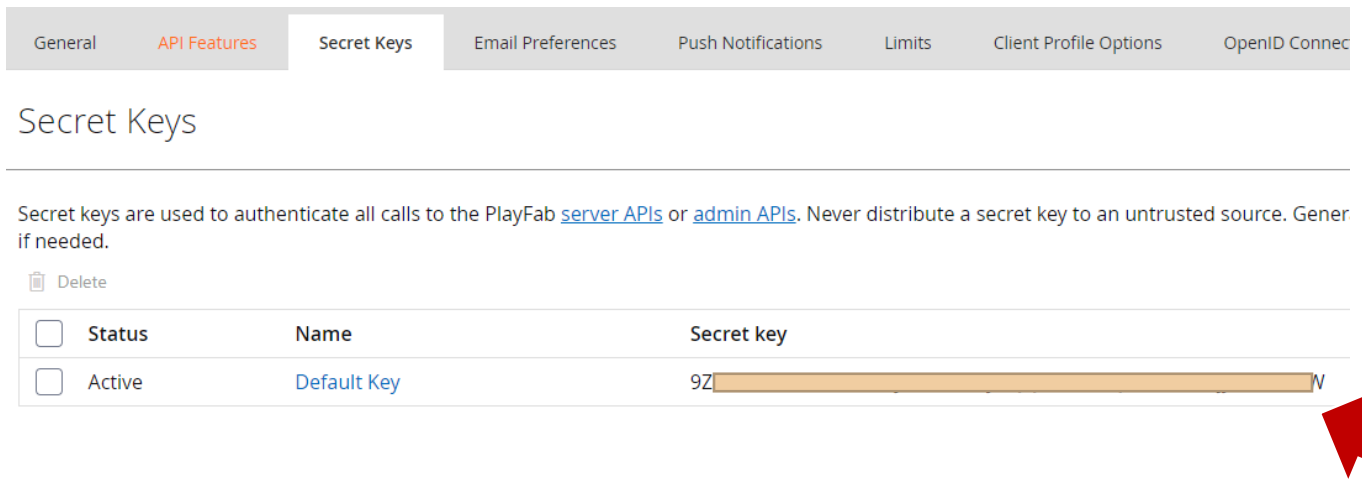


Now to find the Secret Key you need to click on your game then on the top left there should be a gear click the gear then select title settings

Part 2



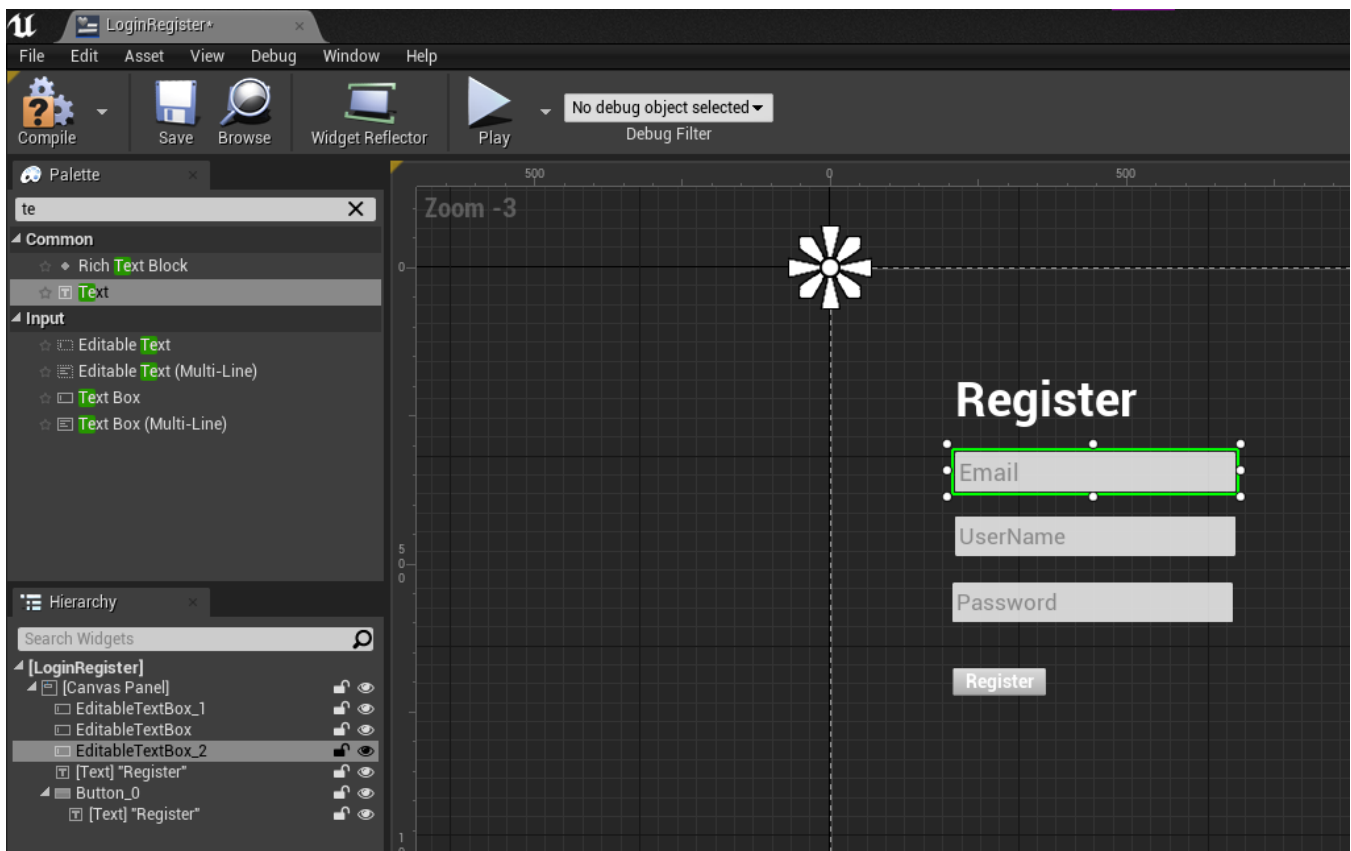
It should say “General, API Features, Secret Keys, Email Preferences “and so on and so forth. The magic item we need it Secret keys



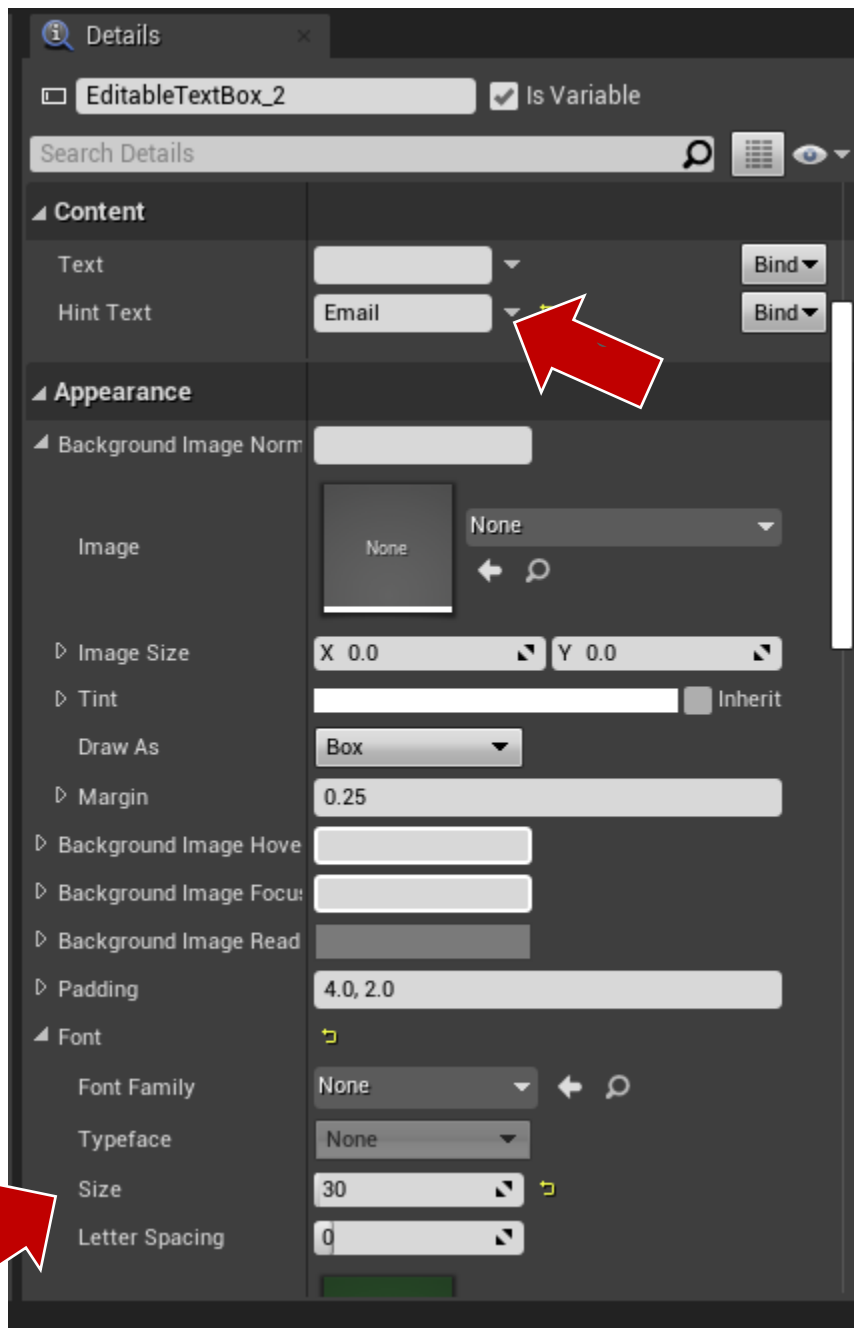
Now Copy the key and paste it in unreal. Great now the easiest step is complete DON'T FORGET TO SAVE. I learnt this the hard way.

1. LOGIN UI

The design and look of your login/register UI is completely up to you but a few key elements need to be the same. You can make the login section and the register section separately but for simplicity in this demonstration it will all be done in one widget. For the Register Ui Three “TextBoxes”. One for the user’s email, username and password. Finally, one button to accept the registration and to make it look better add some text, so the user knows that they are in the registration section. To make the text boxes say email, Username, Password. You need to select the textbox and change the hint text and scroll down until you see “font” and that will change the size of the hint text. Don’t forget to anchor your UI elements so they don’t move in game



Part 2



Now do the same thing for the login but only include the email field and the password field it should look something like this.

The image shows a wireframe of a web form with two sections: **Register** and **Login**. The background is a dark grid. A green dashed line outlines the form area.

Register Section:

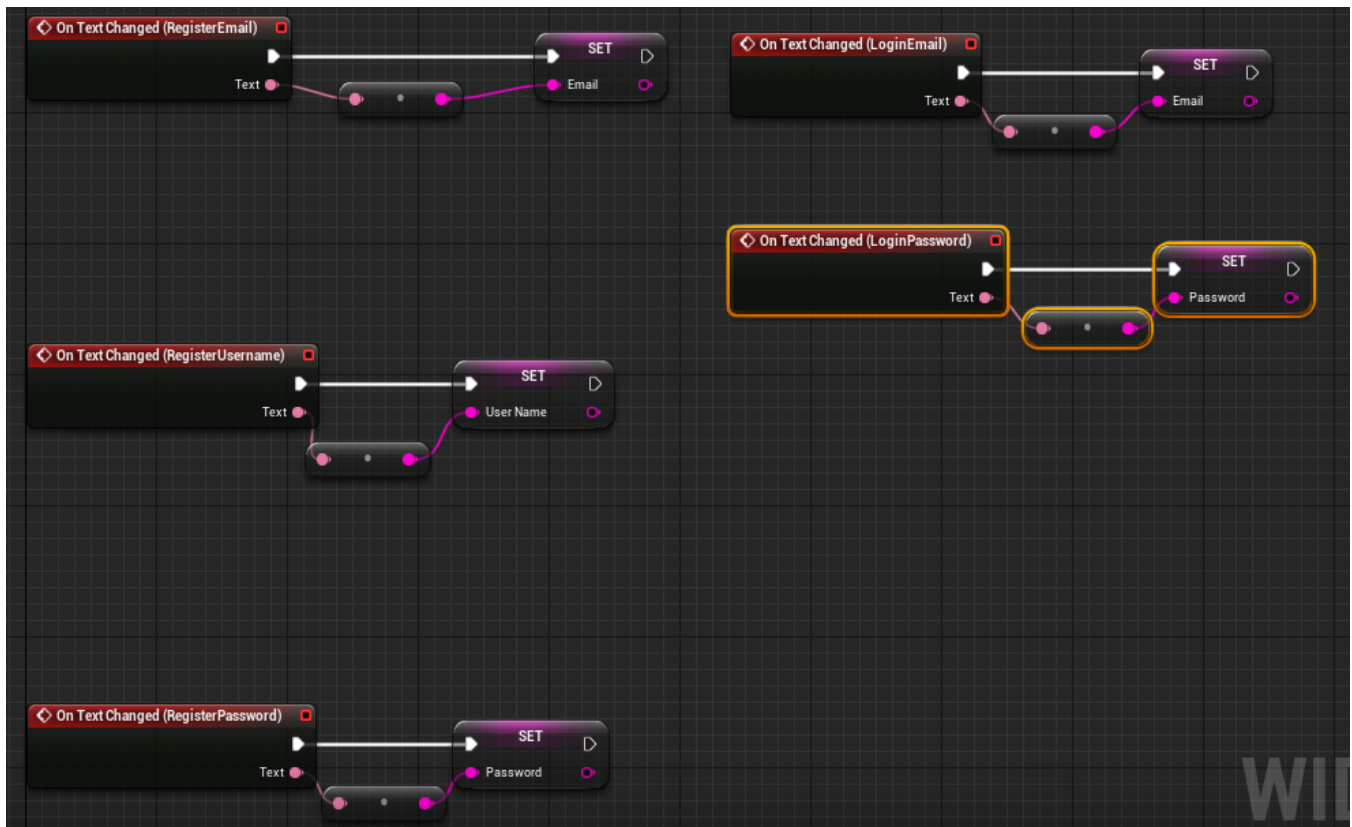
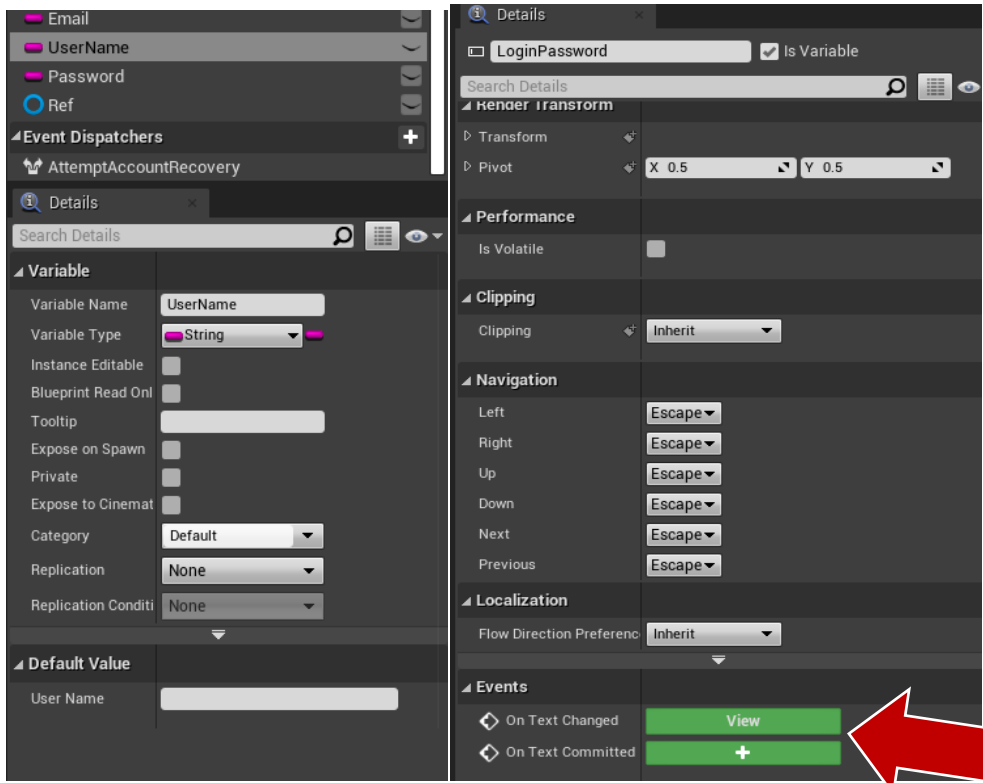
- Text input field: Email
- Text input field: Username
- Text input field: Password
- Button: Register

Login Section:

- Text input field: Email
- Text input field: Password
- Button: Login

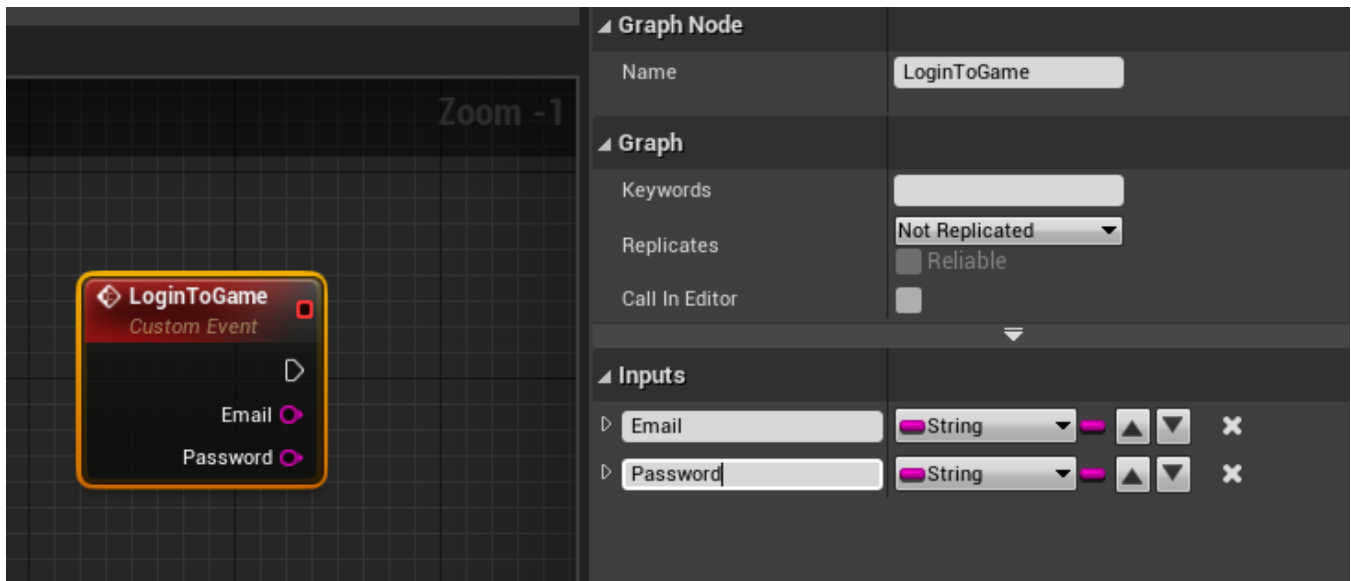
Next make 3 variables with their type being “strings”. Name them as Email, Username, and Password. Once that is done select the editable text box and, in the details, panel scroll down to the events and click “on text changed”. This should bring up the event graph. Do this for all 5 of your editable text boxes. Then attach the variables to the appropriate editable text boxes in the graph section make sure the email variable goes with the email box and the same for the password editable text box and the username. The username variable should only be used once because we only need that data for registration

Part 2

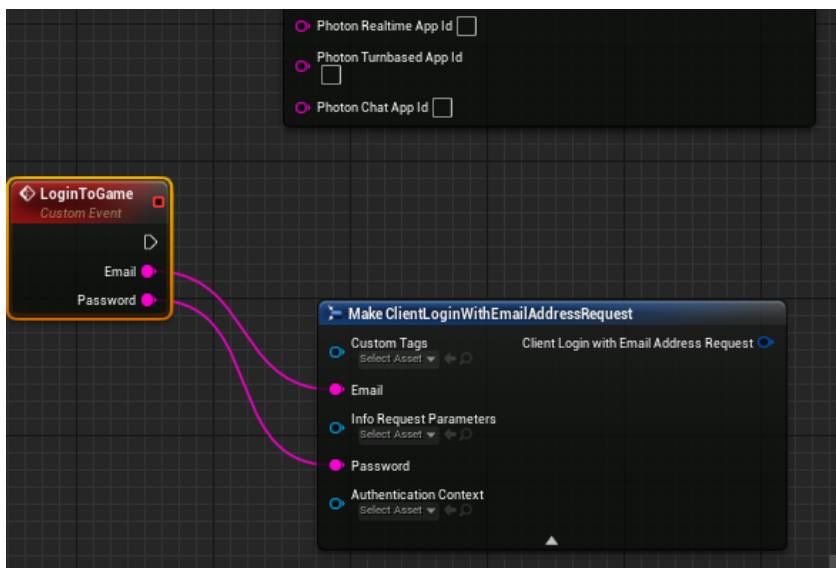


GAME INSTANCE BLUEPRINT

Reopen the game instance we made earlier. Create a custom event and name it “LoginToGame”. Make two inputs and name them Email and Password once again make sure the variable type is string.

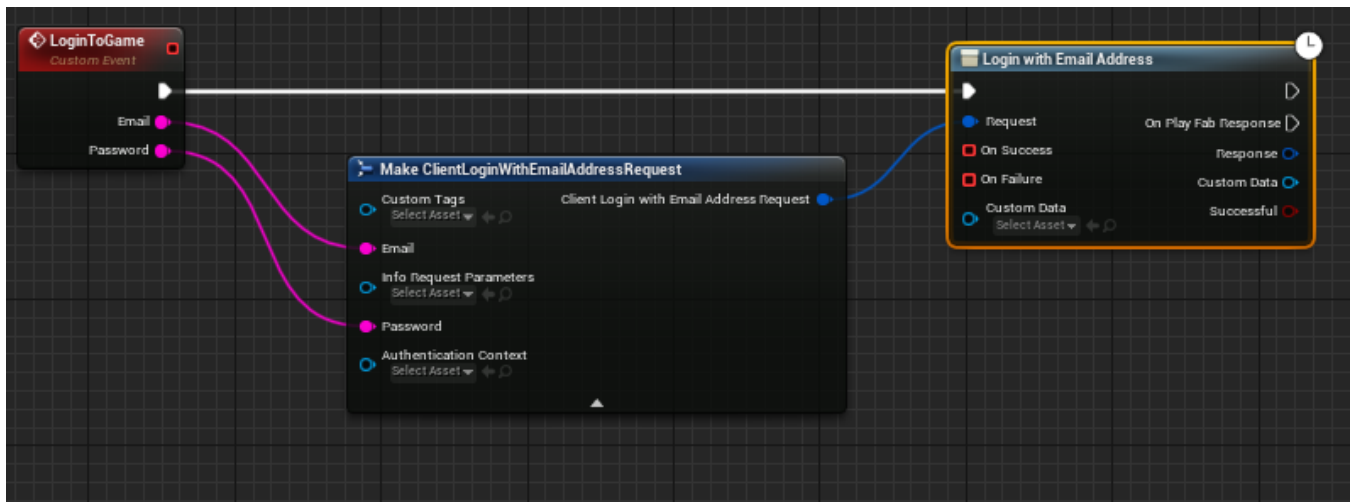


Off the pink dot saying email drag that out and type “Make ClientLoginWithEmail”. Click the arrow to expand the box to and link the password dot on our custom event and connect it to where it says password

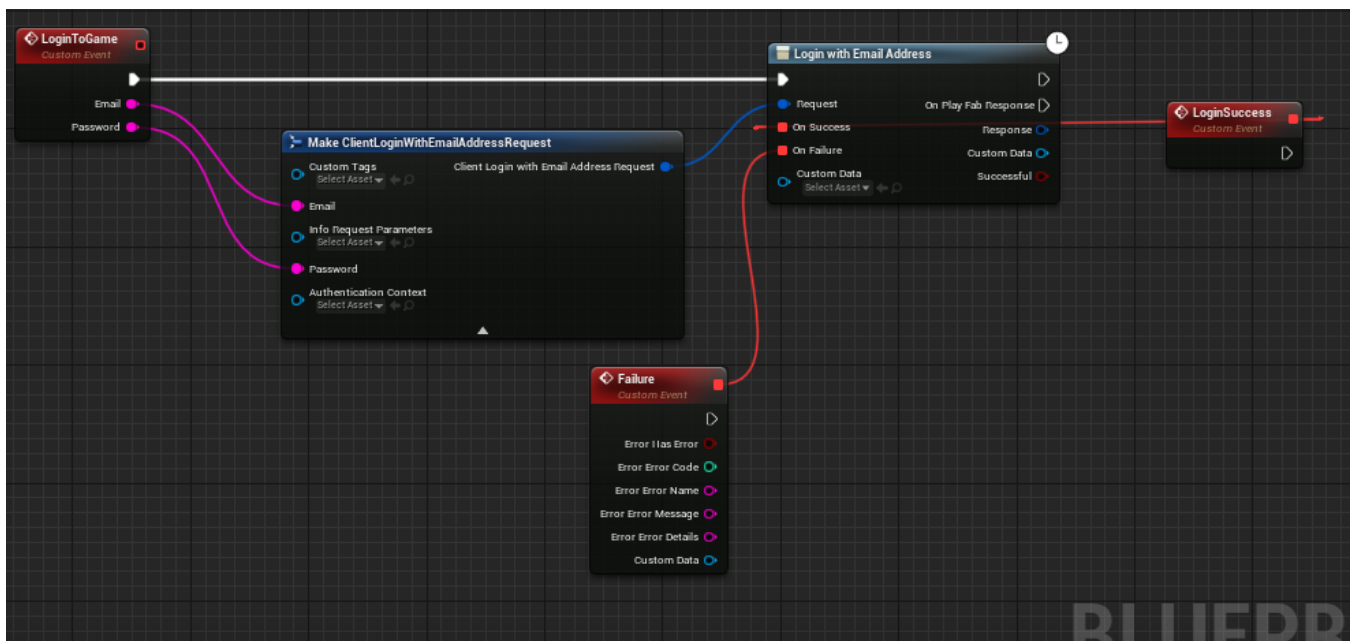


Part 2

Off “clientLoginWithEmailAddressRequest” make a new node called “Login with Email Address” this should go into the request dot and connect the main dot to our “LoginToGame” event.

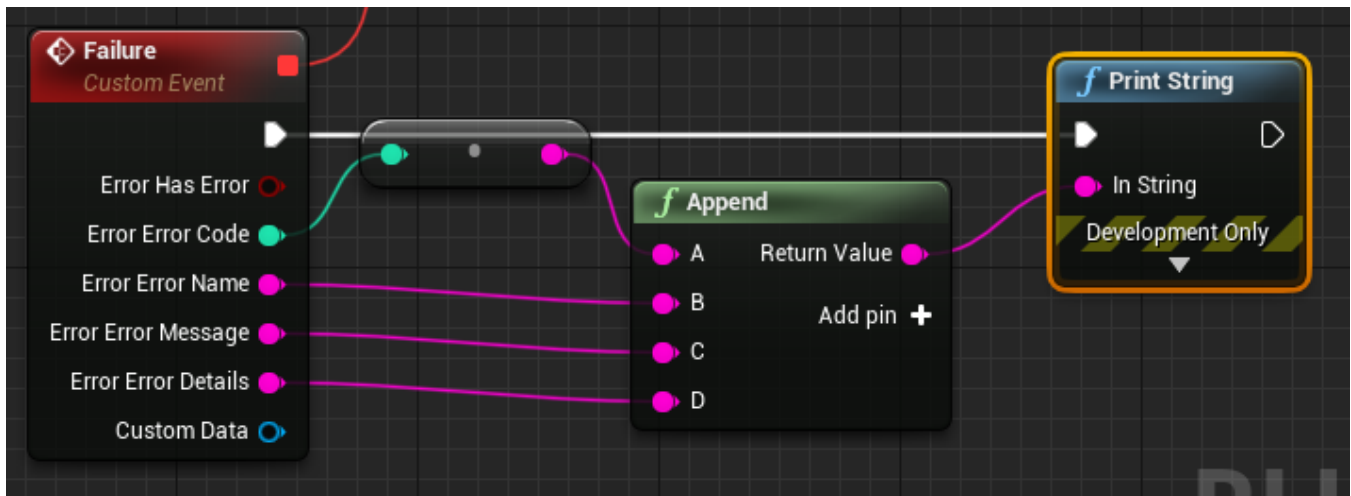


Make a new custom event and name it “LoginSuccess” and connect that to “on Success”. Then Make another custom event and call it “Failure”. Where it says “Error” right click on it and split the structure pins

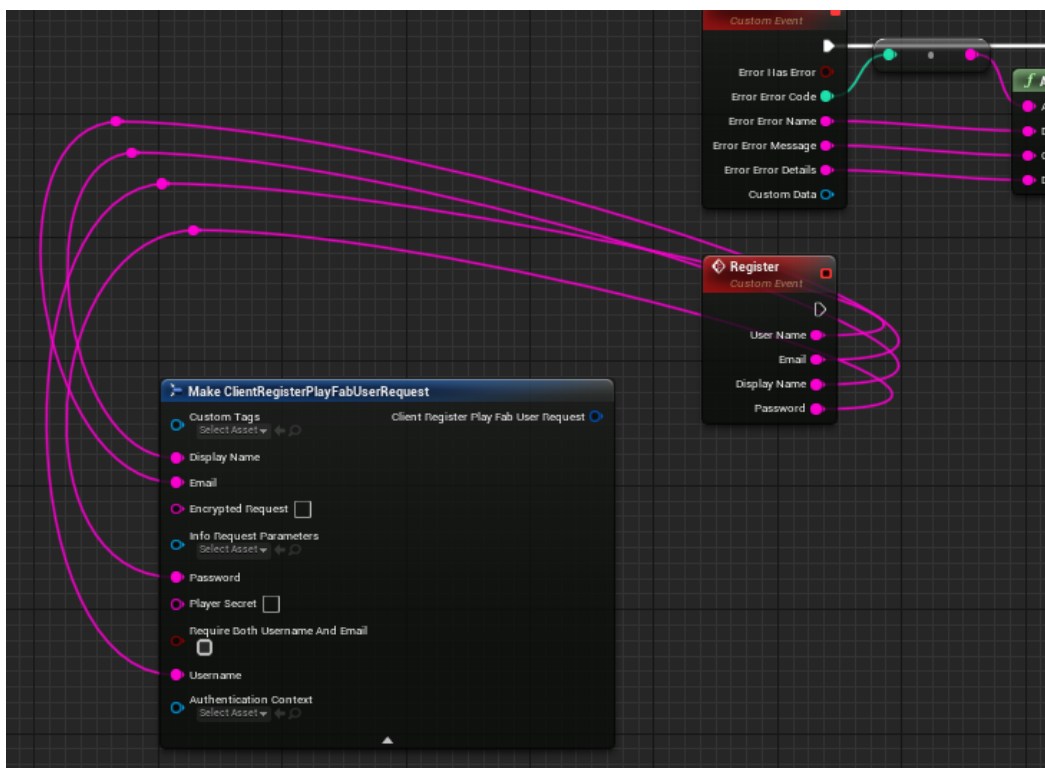


Create a “Append” node and add two more pins. Then connect “Error Error Code” – A “Error Error Name” – B “Error Error Message” – C “Error Error Details” – D. Off of our “Failure” event print string and connect the return value to “in String”. We will reuse that event, so we know what goes wrong with our login process, registration etc...

Part 2

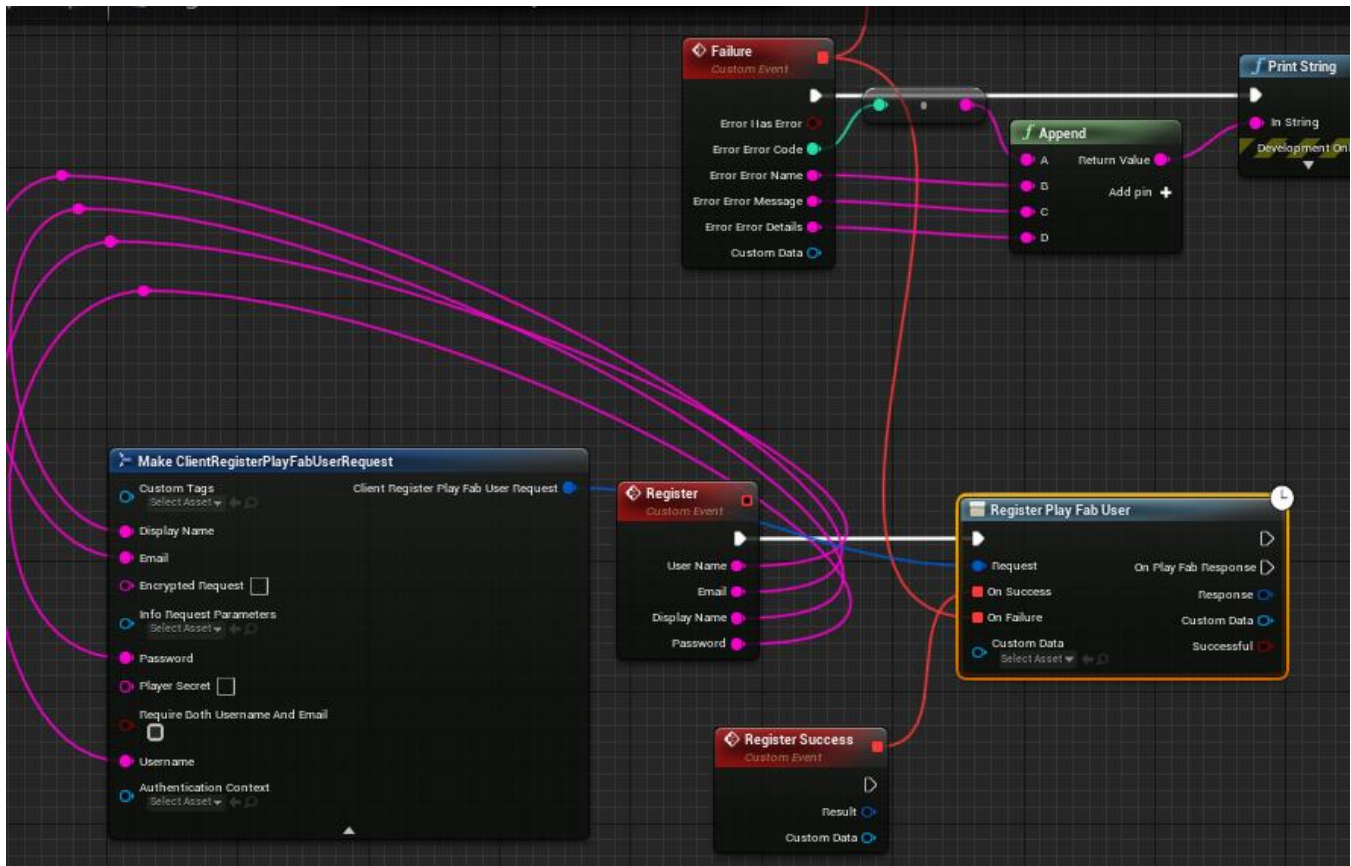


Same thing for the login we need to make a custom event for Registration. Make 4 inputs Username, Email, Display Name and Password. Once again make sure the variable type is a string. Drag off of “Username” and search “Make ClientRegisterPlayFabUserRequest” expand the node with the arrow and connect the rest to their appropriate dot.



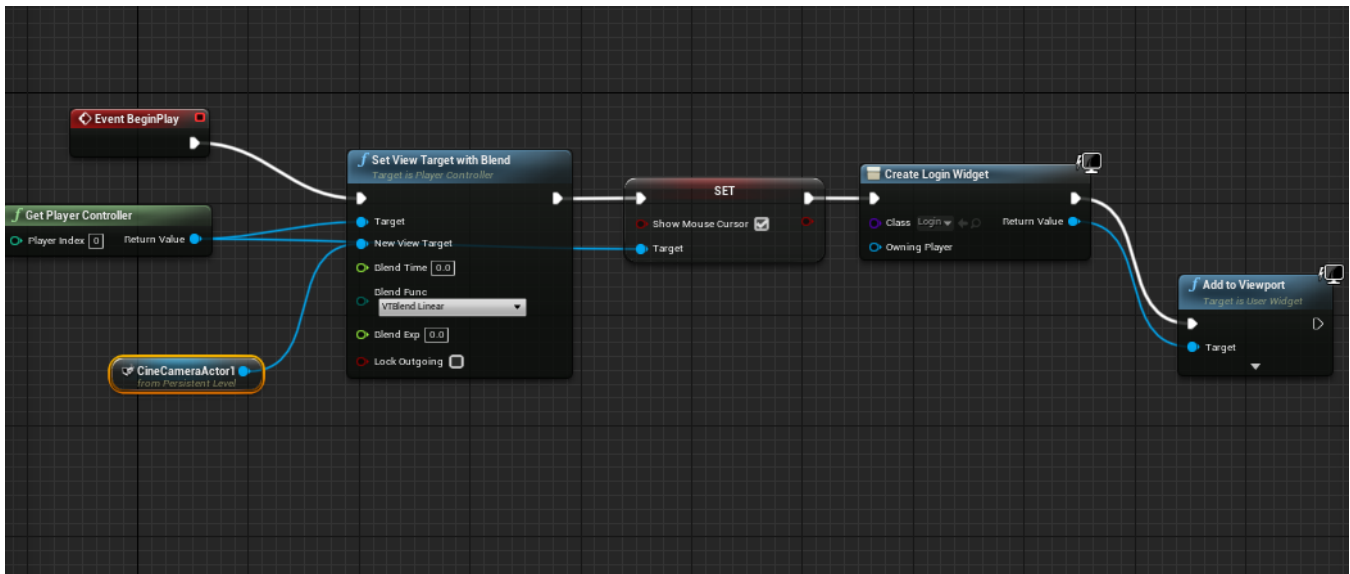
Part 2

Then from “Client Register Play Fab User Request” drag out and search “Register Play Fab User” Then connect the white arrow with the register event. From “On Failure” connect that to the failure event we made earlier and “On Success” make a new event named “RegisterSuccess”

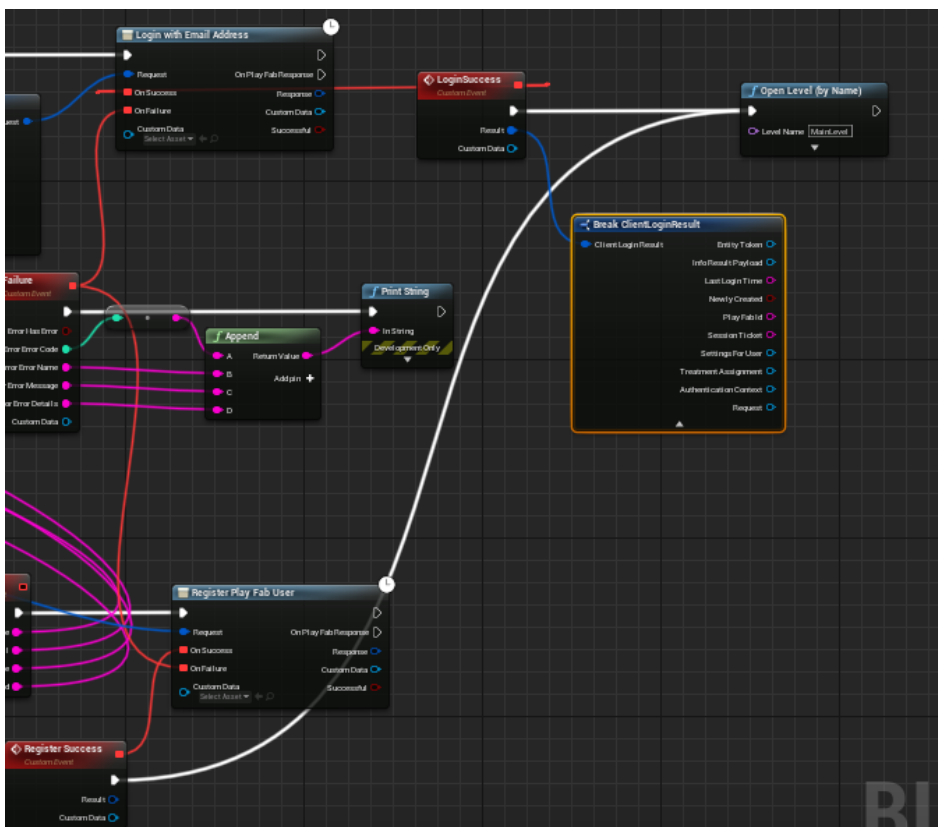


Congratulations this step is done. Just kidding we have a few more things to do later. Make a new Level and name it “menuMap” you can use any map, but I will be using the empty level. If you don’t know how to make a new level, just navigate to the top level where is say file and click new level. Add a “Cine Camera Actor” to the Map. Then open the level Blueprint from the top bar that says “Blueprints”. Write the following code in the level Blueprint. To get the “CineCameraActor” into the graph just drag and drop it from the world outline. You can now press play to make sure your login/register UI appears.

Part 2

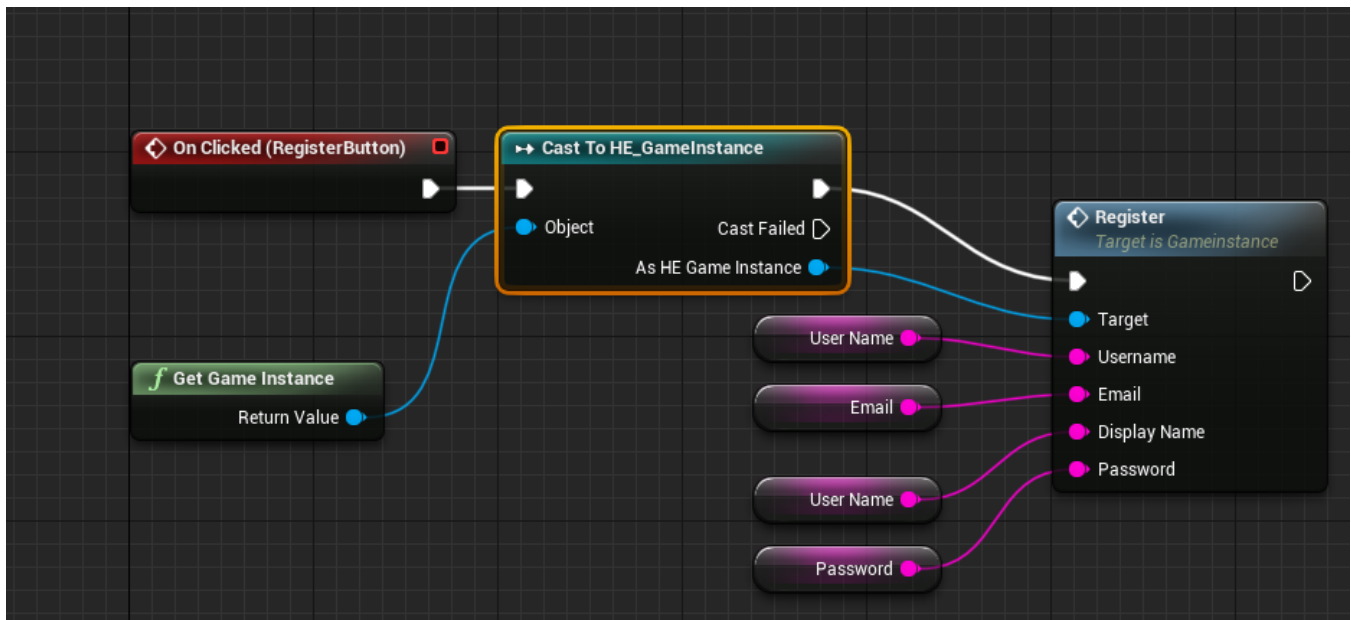


If you already have a level to open after you login or register you can use that one but if not make one. Go back to our game instance and off “LoginSuccess” and “RegisterSuccess” events make a “Open Level (by name)” and put the name of the level you want to open after the registration or login is successful.

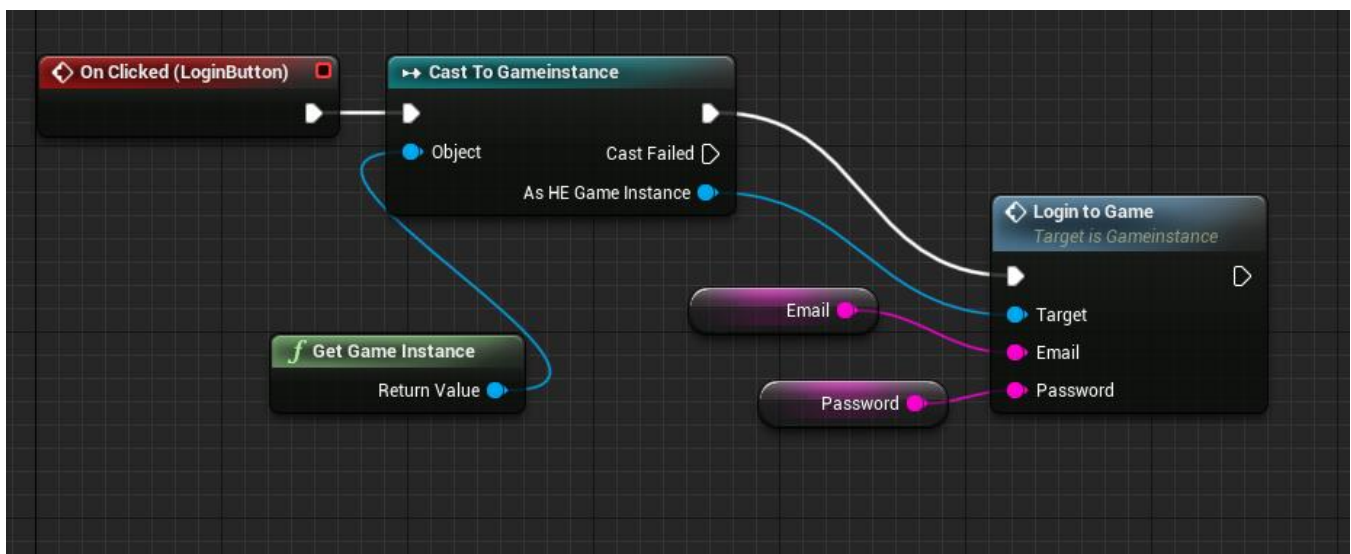


Part 2

That's all the registration and login stuff done in our game instance. To get the game to register or login we need to do a few things in our login UI. Select the button you want to use to confirm registration after all the info is filled out and scroll down in the details panel and click the green onclicked button. Off the button node cast it to your game instance make sure it's the game instance we made. The from object "get Game instance". Then get our "Register" event that we made and connect the 3 variables by dragging them from the left bar.



Same thing for the login Button just cast it to the game instance and get our Login event and connect the Email variable and the Password variable



Part 2

Now the users Should be able to login and register to the game any error would appear so if you did anything wrong you know what to fix. If you go to your game from the playfab web app you will be able to see who when and where people are using, you game.

<div>My Game</div> <div>Development 10 / 100K</div> <div>Title Overview</div> <div>BUILD</div> <div>Players</div> <div>Multiplayer</div> <div>Groups</div> <div>Automation</div> <div>Add-ons</div> <div>ENGAGE</div> <div>Economy</div> <div>Leaderboards</div> <div>Content</div> <div>ANALYZE</div> <div>Dashboards</div> <div>Data</div> <div>Experiments</div>	Players	Segments	Shared Group Data
	Players		
	10 total players		
	Query		
	Most recent logins	Search players	
	ID	Last login	Created
	Country		
	38A252C969D1FC0 nikodmius	Aug 23, 2021 2:36 AM	3 days ago
	9E946BCD928CD38 test	Aug 20, 2021 10:06 PM	5 days ago
	35D3E731724FF00 trulyNathan	Aug 16, 2021 5:31 AM	15 days ago

Thank you for following along with my tutorial. This was made by Nathan Aruna from Montreal QC. I am 14 and love game developing. If you have any questions or something does not work, you can message me on discord at [trulyNátháñ#1001](#). Please keep in mind I have school and other tasks so if I don't answer in a day I'm not ignoring you I'm just probably not able to answer.