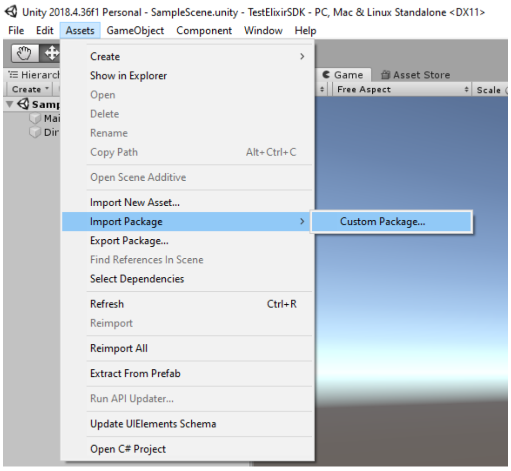
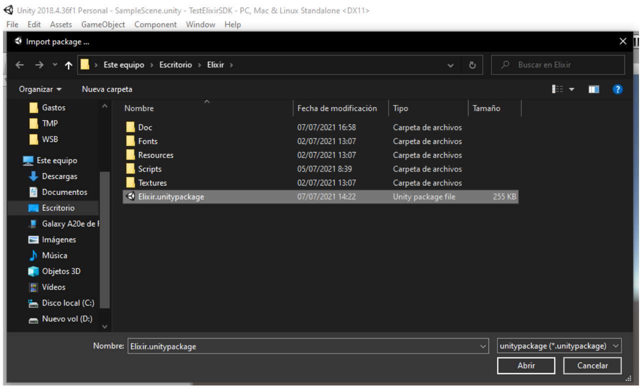
Quick start guide

- Download the package and import it into your project. You can remove the Demo directory if you want.

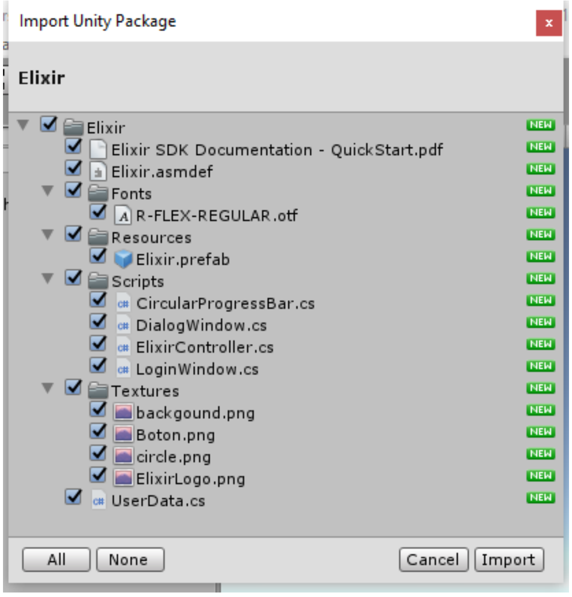




Nombre	Fecha de modificación	Tipo	Tamaño
Doc	07/07/2021 16:58	Carpeta de archivos	
Fonts	02/07/2021 13:07	Carpeta de archivos	
Resources	02/07/2021 13:07	Carpeta de archivos	
Scripts	05/07/2021 8:39	Carpeta de archivos	
Textures	02/07/2021 13:07	Carpeta de archivos	
Elixir.unitypackage	07/07/2021 14:22	Unity package file	255 KB

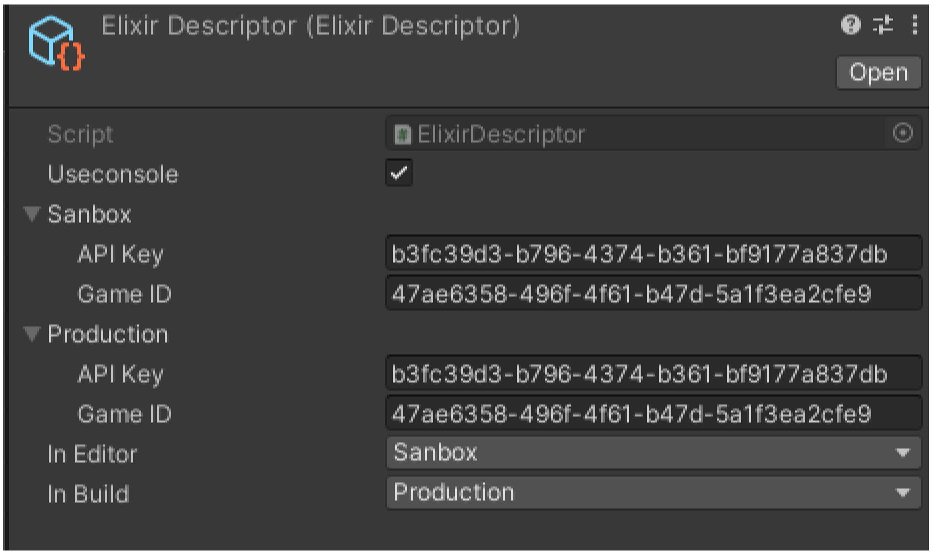
Nombre: Elixir.unitypackage

unitypackage (*.unitypackage)



```
☑ Elixir                                              NEW
    ☑ Elixir SDK Documentation - QuickStart.pdf       NEW
    ☑ Elixir.asmdef                                   NEW
    ☑ Fonts                                           NEW
        ☑ R-FLEX-REGULAR.otf                          NEW
    ☑ Resources                                       NEW
        ☑ Elixir.prefab                               NEW
    ☑ Scripts                                         NEW
        ☑ CircularProgressBar.cs                      NEW
        ☑ DialogWindow.cs                             NEW
        ☑ ElixirController.cs                         NEW
        ☑ LoginWindow.cs                              NEW
    ☑ Textures                                        NEW
        ☑ backgound.png                               NEW
        ☑ Boton.png                                   NEW
        ☑ circle.png                                  NEW
        ☑ ElixirLogo.png                              NEW
    ☑ UserData.cs                                     NEW
```

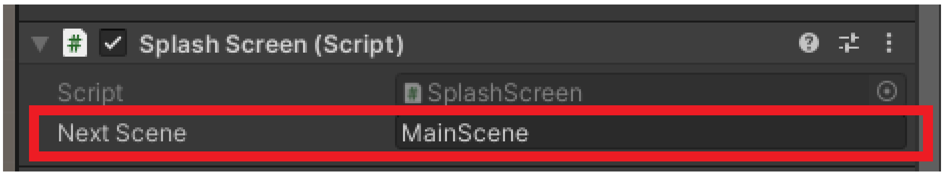
- Look for **Assets/Elixir/Resources/ElixirDescriptor** and set your **APIKeys** and **GameIDs** for Sandbox and Production.



Elixir Descriptor (Elixir Descriptor)

Script ElixirDescriptor
Useconsole ✓
Sanbox
API Key b3fc39d3-b796-4374-b361-bf9177a837db
Game ID 47ae6358-496f-4f61-b47d-5a1f3ea2cfe9
Production
API Key b3fc39d3-b796-4374-b361-bf9177a837db
Game ID 47ae6358-496f-4f61-b47d-5a1f3ea2cfe9
In Editor Sanbox
In Build Production

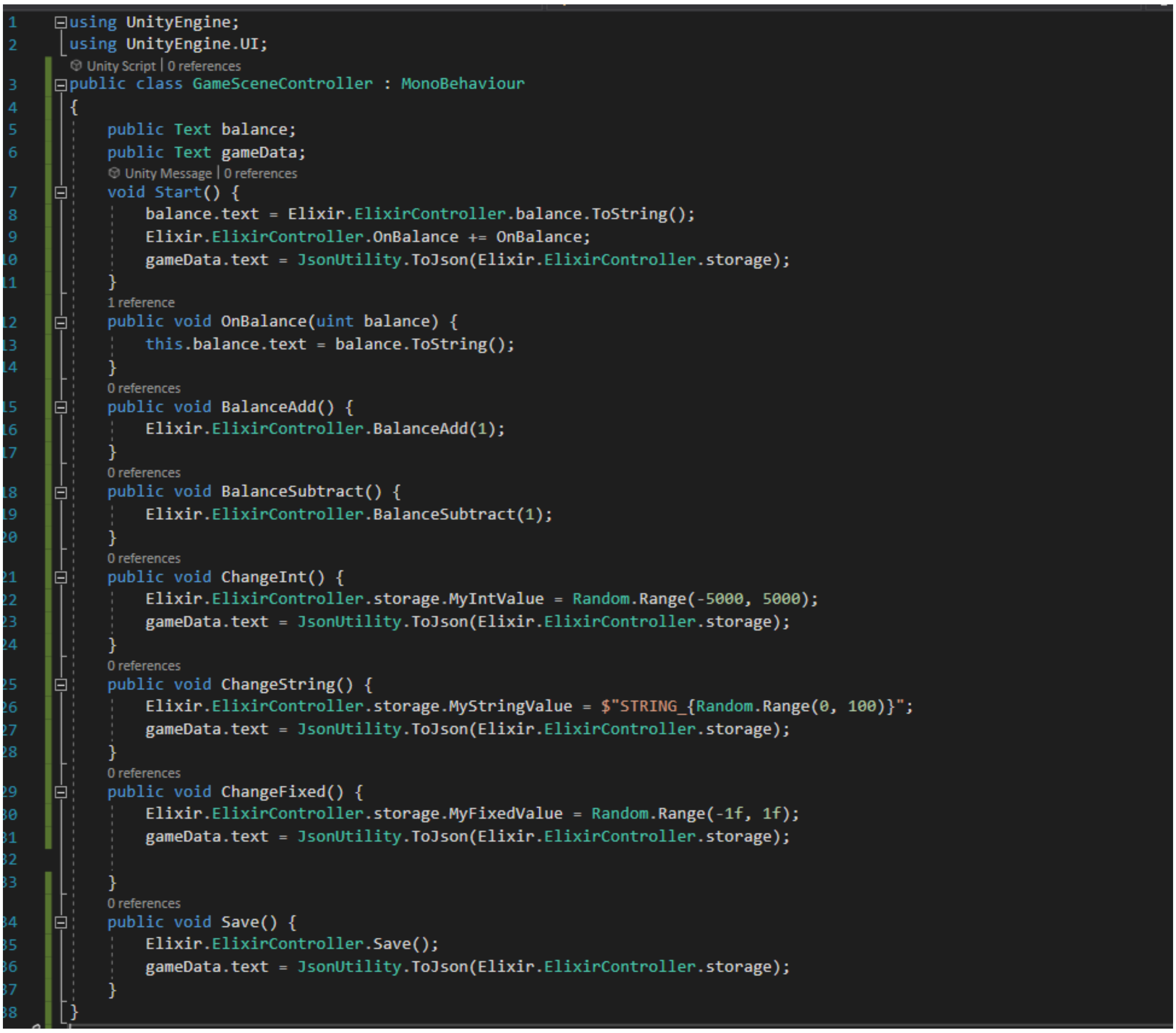
- You can use the dropdowns “In Editor” and “In Build” to target the two environments.
- The “Useconsole” check adds logs to the Unity console.
- Set **Assets/Elixir/Scenes/Splash** as your first scene on **Build Settings.**
- Open this scene and change on **Hierarchy/Splash/Canvas/SplashScreen, Splash Screen Script** the name of the very first scene after Elixir will be initiated.
-



✓ Splash Screen (Script)

Script SplashScreen
Next Scene MainScene

- To ask for the balance of the user, use **Elixir.ElixirController.balance.** You can set a callback using **Elixir.ElixirController.OnBalance += (balance)=>{};** it will be called every time **balance** is modified. Or, you can add the **SatoshiBalanceController** script to a **UnityEngine.UI.Text** control.
- To add Satoshis to the user, call to **Elixir.ElixirController.BalanceAdd(amount);**
- To subtract Satoshis to the user, call to **Elixir.ElixirController.BalanceSubtract(amount);**



```
using UnityEngine;
using UnityEngine.UI;
Unity Script | 0 references
public class GameSceneController : MonoBehaviour
{
    public Text balance;
    public Text gameData;
    Unity Message | 0 references
    void Start() {
        balance.text = Elixir.ElixirController.balance.ToString();
        Elixir.ElixirController.OnBalance += OnBalance;
        gameData.text = JsonUtility.ToJson(Elixir.ElixirController.storage);
    }
    1 reference
    public void OnBalance(uint balance) {
        this.balance.text = balance.ToString();
    }
    0 references
    public void BalanceAdd() {
        Elixir.ElixirController.BalanceAdd(1);
    }
    0 references
    public void BalanceSubtract() {
        Elixir.ElixirController.BalanceSubtract(1);
    }
    0 references
    public void ChangeInt() {
        Elixir.ElixirController.storage.MyIntValue = Random.Range(-5000, 5000);
        gameData.text = JsonUtility.ToJson(Elixir.ElixirController.storage);
    }
    0 references
    public void ChangeString() {
        Elixir.ElixirController.storage.MyStringValue = $"STRING_{Random.Range(0, 100)}";
        gameData.text = JsonUtility.ToJson(Elixir.ElixirController.storage);
    }
    0 references
    public void ChangeFixed() {
        Elixir.ElixirController.storage.MyFixedValue = Random.Range(-1f, 1f);
        gameData.text = JsonUtility.ToJson(Elixir.ElixirController.storage);

    }
    0 references
    public void Save() {
        Elixir.ElixirController.Save();
        gameData.text = JsonUtility.ToJson(Elixir.ElixirController.storage);
    }
}
```

Keeping data on the Elixir cloud.

- Once **Elixir.ElixirController.isReady** is true, you can read all your stored data from **Elixir.ElixirController.storage**. In order to personalize your data, modify the **class UserData** (Assets/Elixir/UserData.cs) at your convenience.
- You can modify the instance of the UserData class as many times as you want, but the data will only be saved when calling the **Elixir.ElixirController.Save()** function.
- The next time the user enters the game, their data will be available
- Mind to keep your UserData class modifications if you update the SDK.