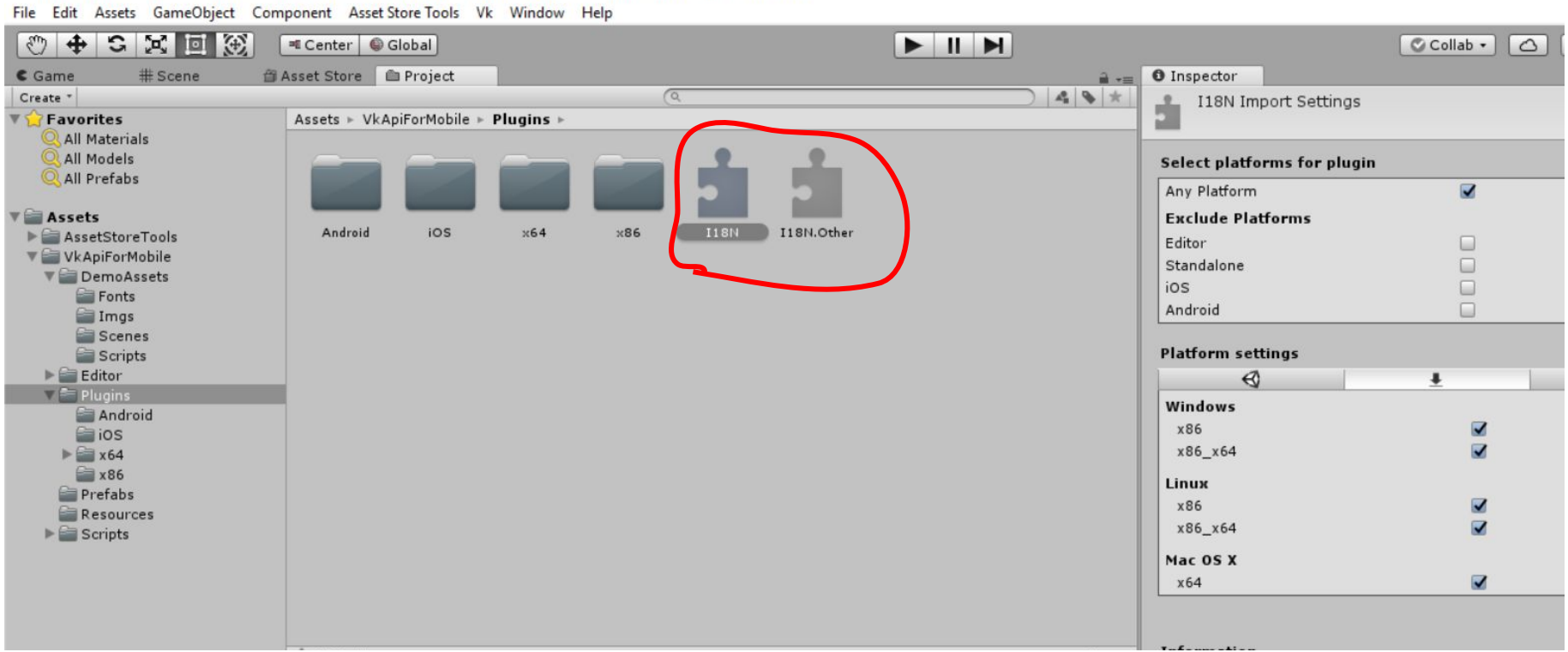
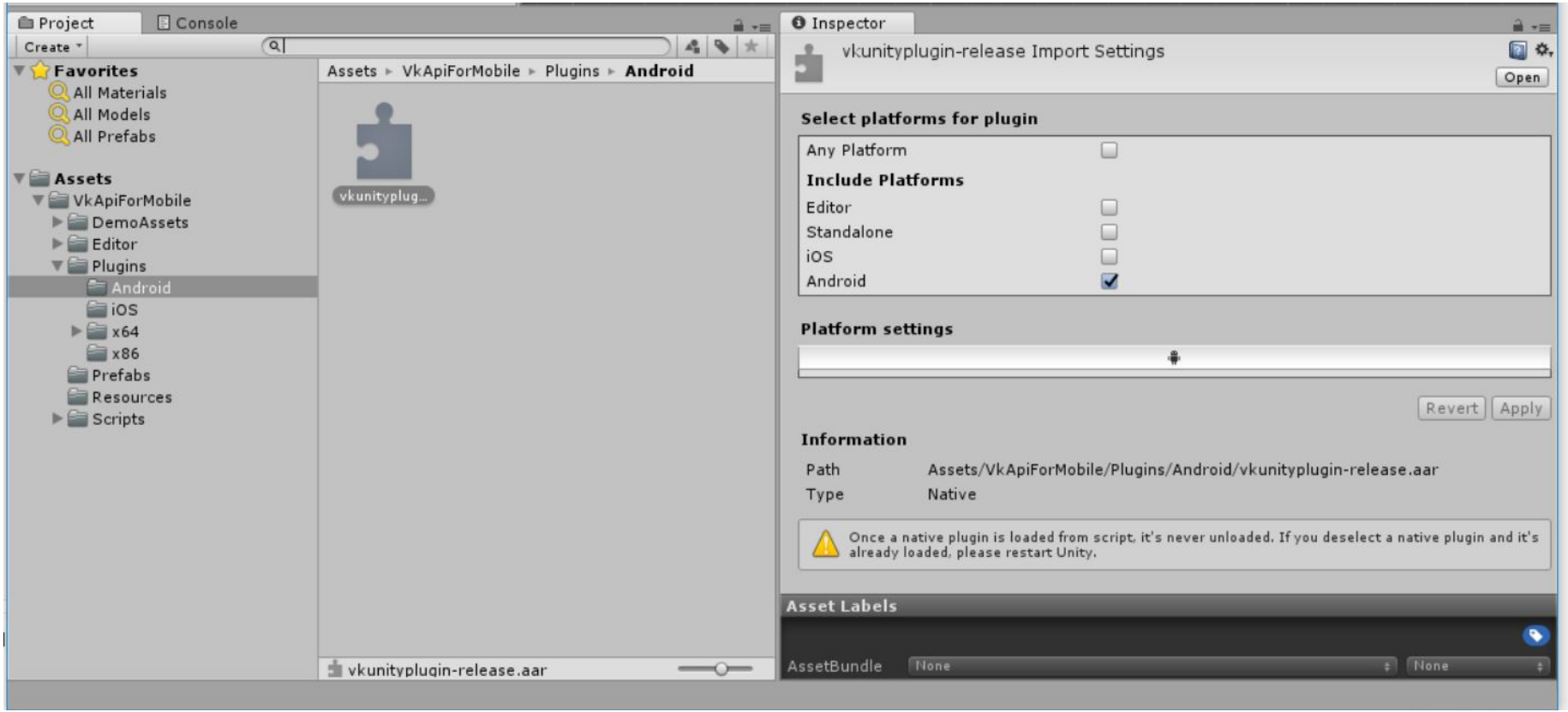
VK Sdk for Unity

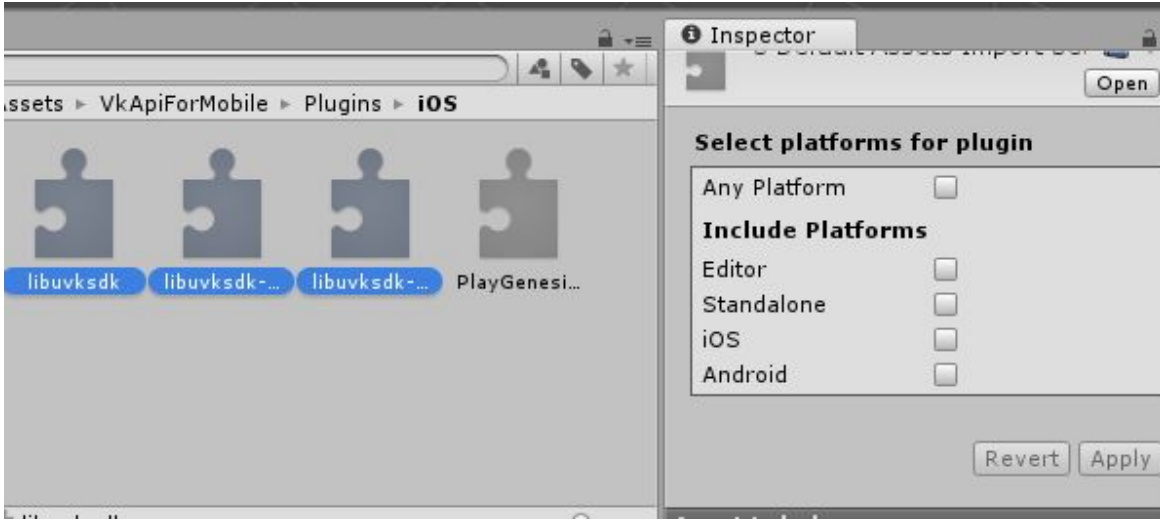
Import Plugin in Unity

Check for plugin dlls are marked correctly for each platform and architecture


File Edit Assets GameObject Component Asset Store Tools Vk Window Help
Center
Global
Collab
Game
Scene
Asset Store
Project
Inspector
Create
Favorites
All Materials
All Models
All Prefabs
Assets
AssetStoreTools
VkApiForMobile
DemoAssets
Fonts
Imgs
Scenes
Scripts
Editor
Plugins
Android
iOS
x64
x86
Prefabs
Resources
Scripts
Assets ▸ VkApiForMobile ▸ Plugins ▸
Android
iOS
x64
x86
I18N
I18N.Other
I18N Import Settings
Select platforms for plugin
Any Platform
Exclude Platforms
Editor
Standalone
iOS
Android
Platform settings
Windows
x86
x86_x64
Linux
x86
x86_x64
Mac OS X
x64


Project
Console
Create
Favorites
All Materials
All Models
All Prefabs
Assets
VkApiForMobile
DemoAssets
Editor
Plugins
Android
iOS
x64
x86
Prefabs
Resources
Scripts
Assets ▸ VkApiForMobile ▸ Plugins ▸ Android
vkunityplug...
vkunityplugin-release.aar
Inspector
vkunityplugin-release Import Settings
Open
Select platforms for plugin
Any Platform
Include Platforms
Editor
Standalone
iOS
Android
Platform settings
Revert
Apply
Information
Path
Assets/VkApiForMobile/Plugins/Android/vkunityplugin-release.aar
Type
Native
Once a native plugin is loaded from script, it's never unloaded. If you deselect a native plugin and it's already loaded, please restart Unity.
Asset Labels
AssetBundle
None
None

- there are 4 version of static library for ios:
  - libuvksdk (release build without including bitcode)
  - libuvksdk-bc (release build including bitcode)
  - libuvksdk-debug (debug build)
- only one of them should be checked for ios, just to be sure select all of them and uncheck all the platforms.



- bitcode version should be used when your main xcode project is also including bitcode (otherwise you won't be able to upload the application to the app store)
- select the version your need and mark it for ios (here I use release with bitcode)


Assets ▸ VkApiForMobile ▸ Plugins ▸ iOS
libuvksdk
libuvksdk-...
libuvksdk-...
PlayGenesi...
PlayGenesisWebViewController.xib
Inspector
PlayGenesisWebViewContr
Open
Select platforms for plugin
Any Platform
Include Platforms
Editor
Standalone
iOS
Android
Platform settings
Asset Labels

Inspector

vkunityplugin.bundle Import Settings

Open

Select platforms for plugin

Any Platform ☐

Include Platforms

Editor ☐

Standalone ☑

iOS ☐

Android ☐

WebGL ☐

Platform settings

Mac OS X

x64 ☑

Plugin load settings

Load on startup ☐

Revert Apply

Information

Path Assets/VkApiForMobile/Plugins/x64/vkunityplu

Type Native

Once a native plugin is loaded from script, it's never unloaded. If you

Project Console

Create

- Favorites
  - All Materials
  - All Models
  - All Prefabs
- Assets
  - Scenes
  - VkApiForMobile
    - DemoAssets
    - Editor
      - NativePluginsSourceCode
    - Plugins
      - Android
      - iOS
      - x64
      - x86
    - Prefabs
    - Resources
    - Scripts
- Packages

Assets > VkApiForMobile > Plugins > x64

PGWebViewW...

vkunityplugin...

Inspector

PGWebViewWin Import Settings

Open

Select platforms for plugin

Any Platform ☐

Include Platforms

Editor ☐
Standalone ☑
iOS ☐
Android ☐
WebGL ☐

Platform settings

Windows

x86 ☐
x86_x64 ☑

Plugin load settings

Load on startup ☐

Revert Apply

Information

Path Assets/VkApiForMobile/Plugins/x64/PGWebVie
Type Native

Once a native plugin is loaded from script, it's never unloaded. If you deselect a native plugin and it's already loaded, please restart Unity.

Project Console

Create

- Favorites
  - All Materials
  - All Models
  - All Prefabs
- Assets
  - Scenes
  - VkApiForMobile
    - DemoAssets
    - Editor
      - NativePluginsSourceCode
    - Plugins
      - Android
      - iOS
      - x64
      - x86
    - Prefabs
    - Resources
    - Scripts
- Packages

Assets > VkApiForMobile > Plugins > x64

PGWebViewW... vkunityplugin...

Inspector

PGWebViewWin Import Settings

Open

Select platforms for plugin

Any Platform ☐

Include Platforms

Editor ☐
Standalone ☑
iOS ☐
Android ☐
WebGL ☐

Platform settings

Windows
x86 ☑
x86_x64 ☐

Plugin load settings

Load on startup ☐

Revert Apply

Information

Path Assets/VkApiForMobile/Plugins/x86/PGWebVie
Type Native

Once a native plugin is loaded from script, it's never unloaded. If you deselect a native plugin and it's already loaded, please restart Unity.

Project Console

Create

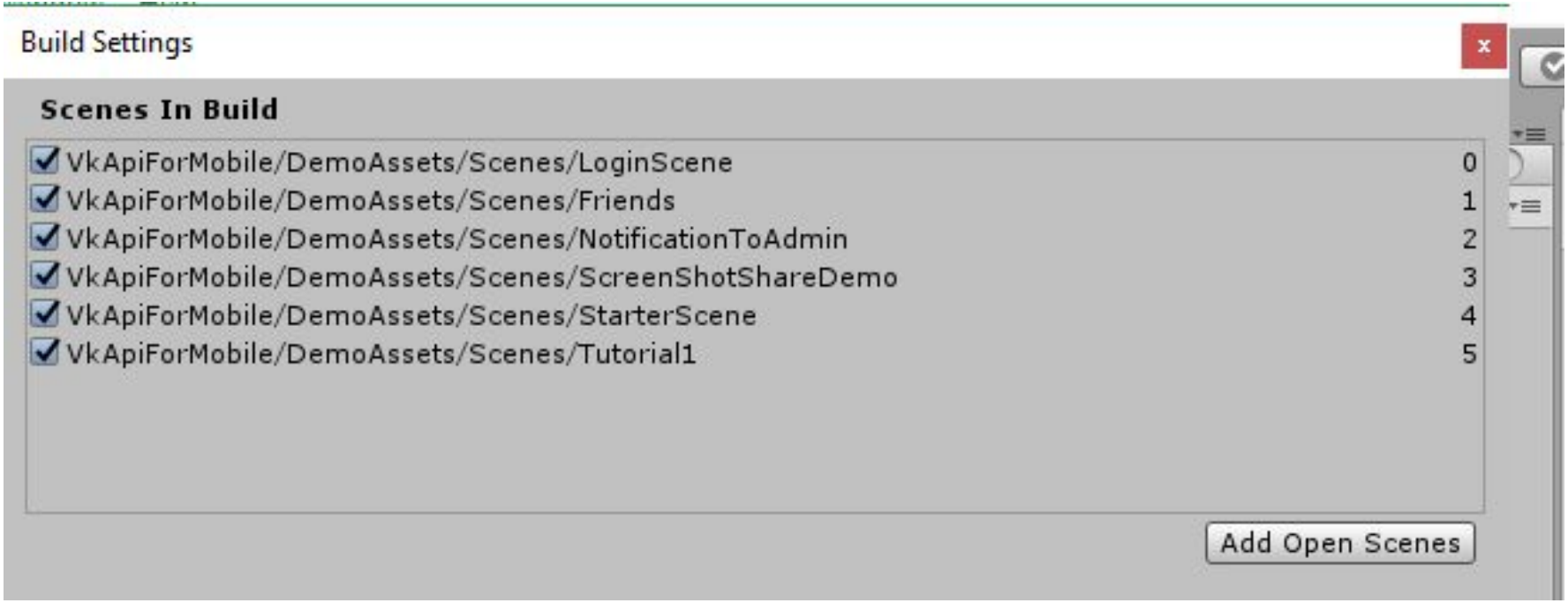
Favorites
All Materials
All Models
All Prefabs

Assets
Scenes
VkApiForMobile
DemoAssets
Editor
NativePluginsSourceCode
Plugins
Android
iOS
x64
x86
Prefabs
Resources
Scripts
Packages

Assets > VkApiForMobile > Plugins > **x86**

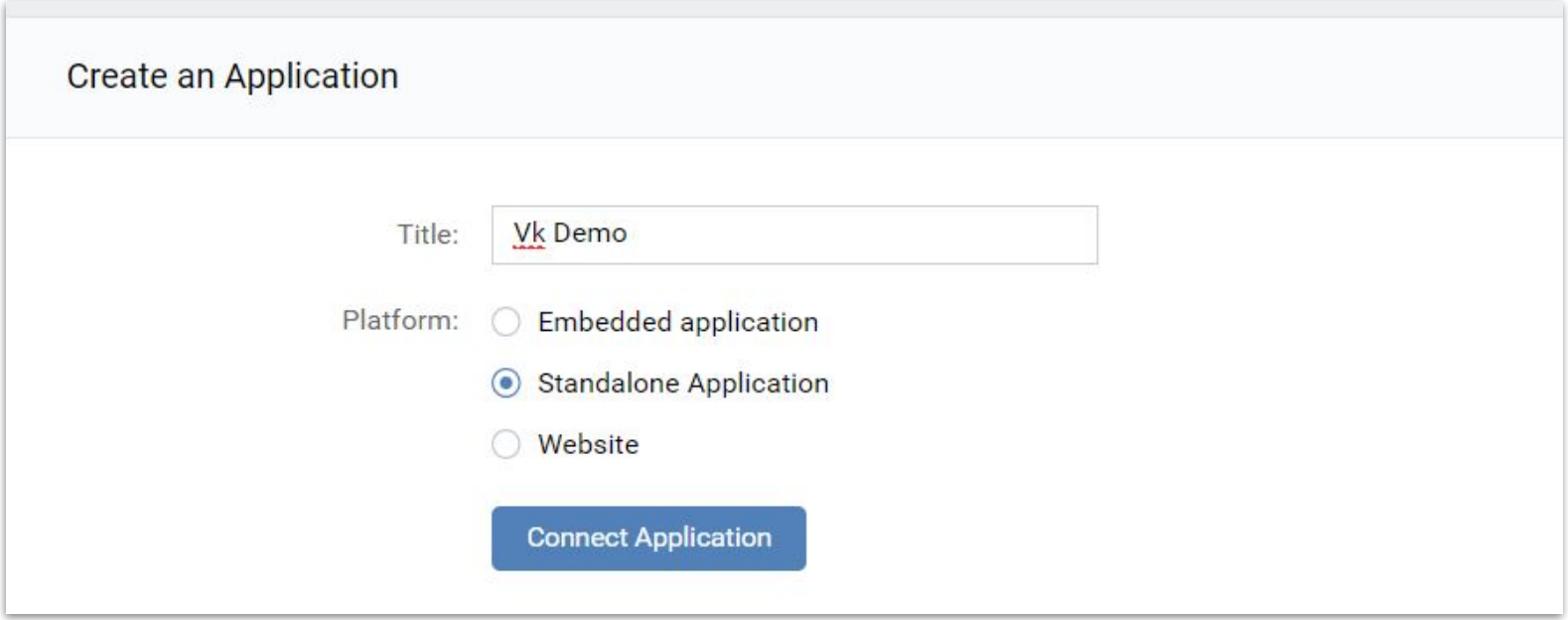
PGWebViewW...

If you want to make a test build with scenes provided in the plugin, assure that the first scene is LoginScene

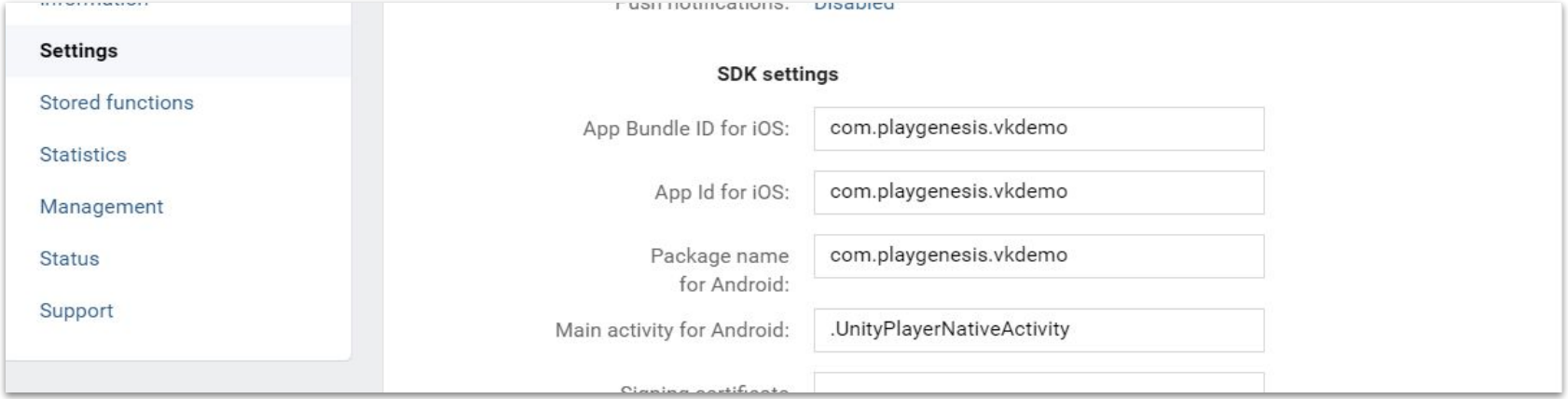


Setup on VK Web portal

- navigate to vk.com/dev select “my application” and “create new app”
- give it a name and select Platform to be a Standalone Application



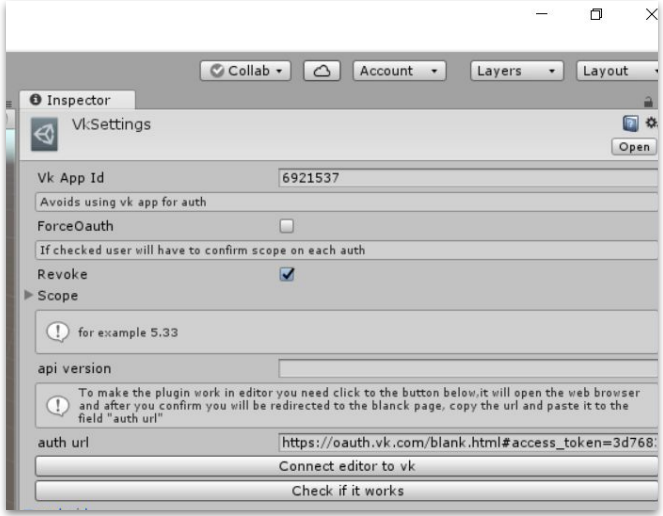
- go to setting of your new created app
- fill App Bundle ID for iOS, App Id for iOS, Package name for Android. The same ids you will need to input in Unity Player Setting
- fill Main activity for Android with **.UnityPlayerNativeActivity**
  In the end of this step you should have something similar to this:



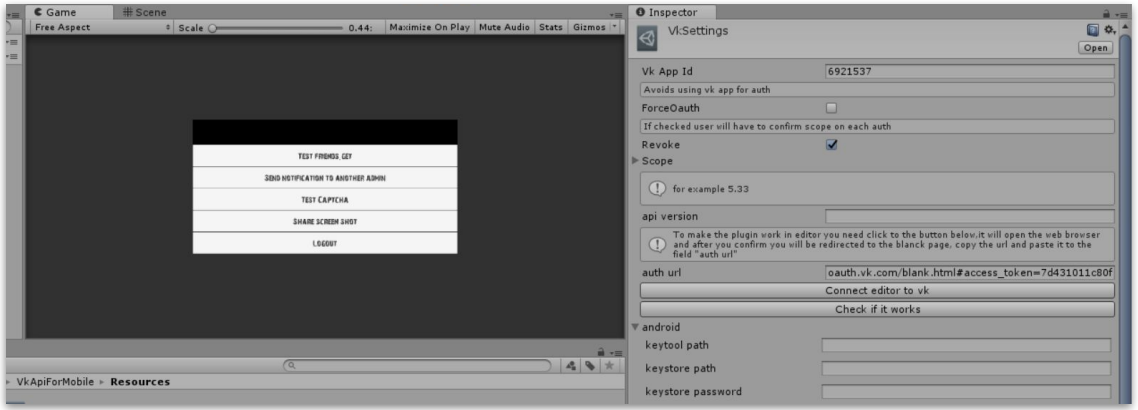
- copy somewhere Application ID, we'll need it later and save settings

Unity Editor Setup

- import Vk Sdk package to your unity project
- on top bar your should see new menu item “**VK**”, to to **VK->Edit Vk Setting**
- fill **Vk App Id** with **Application Id** from vk web portal
- open scope section and check scope items you need. For this particular example we’ll need access to “Friends”, “Offline”, “Wall” and “Email”
- click on “connect editor to vk”, this will open your system web browser. Login and allow access for the application, copy url your will be redirected to and paste it back to unity editor “auth url” field.

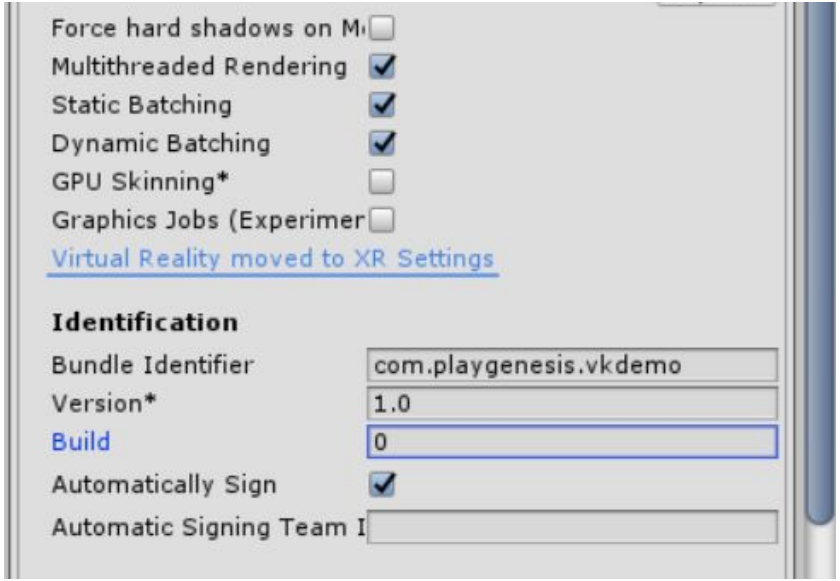


- if you've done everything correctly, clicking on "check if it works" button should open the system web browser and you'll see the information about your vk account.
- now editor is connected. You may open DemoAssets->Scenes->LoginScene and run the demo in Unity Editor.

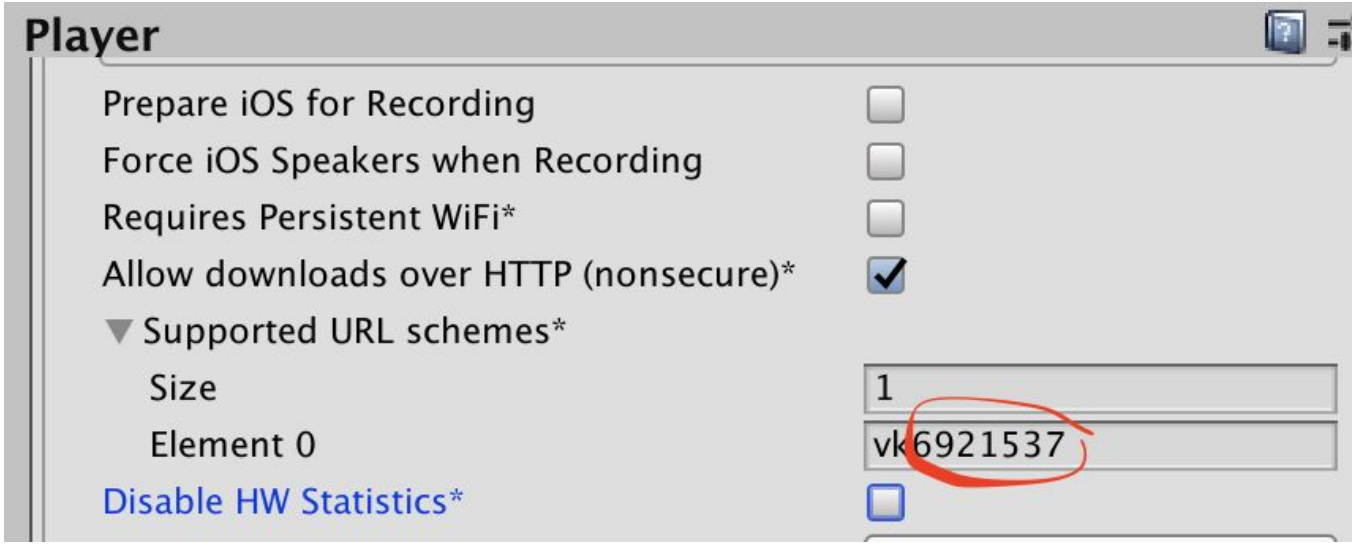


iOS Setup

- copy **App Bundle ID for iOS** from app setting section on vk web portal and paste it to File->Build Settings->Player Settings->Bundle Identifier



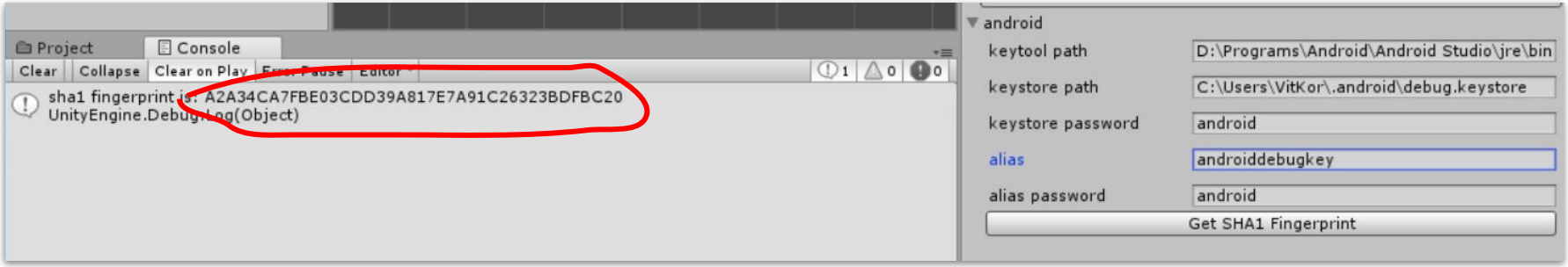
File->Build Settings->Player Settings->**Supported Url Schemes** add vk+app id in the case it will become **vk6921537**



Android Setup

- open Vk->Edit Vk Setting in Unity Editor and scroll to bottom
- fill full path to keytool.exe, search in your java/jre/bin folder. In my case I have android studio installed and I'll find it there. **D:\Programs\Android\Android Studio\jre\bin\keytool.exe** for windows.
- fill path to keystore file. In this case I'll use an android debug keystore. But the same process is applicable to any keystore file.
- fill the keystore password, alias and alias password(for android debug key keystore password and alias password should be **android**, alias should be **androiddebugkey**

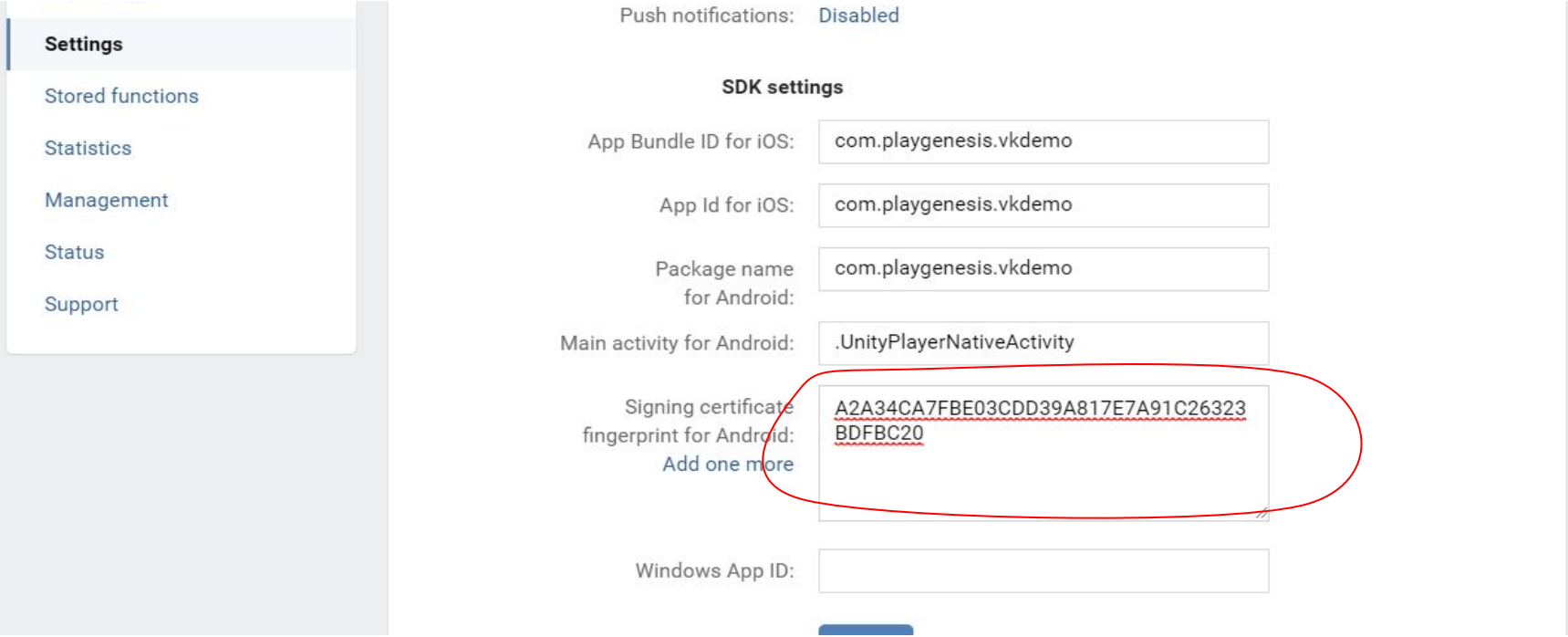
By clicking on Get SHA1 Fingerprint your should see a unity debug message with sha1 for your key



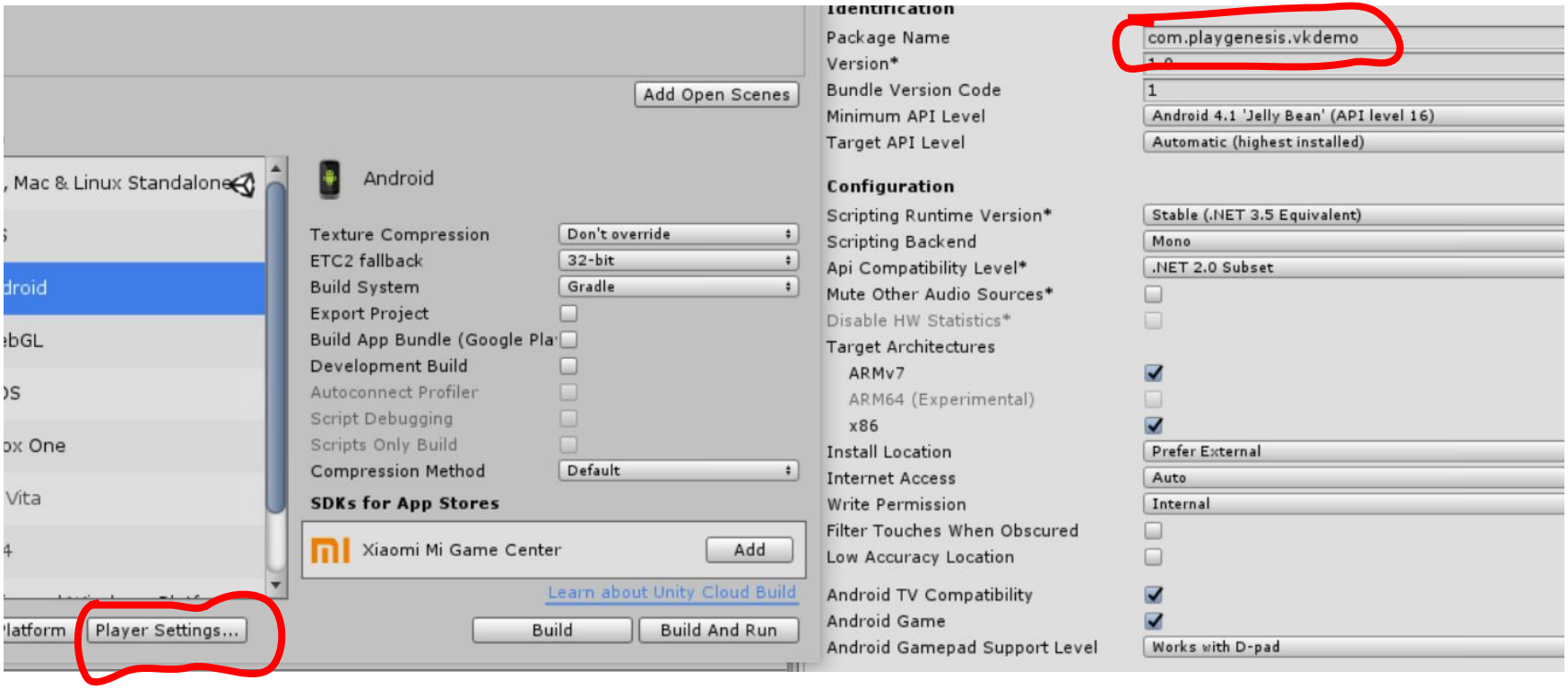
vk web portal supports multiple fingerprints so I definitely recommend to add android debug key fingerprint as one of them. In case you forget to sign your application, unity will use debug key

In case you are using OS X, your path to keytool and keystore files **should not** have backslashes (ex. **/My\ Folder** should be **\My Folder)**

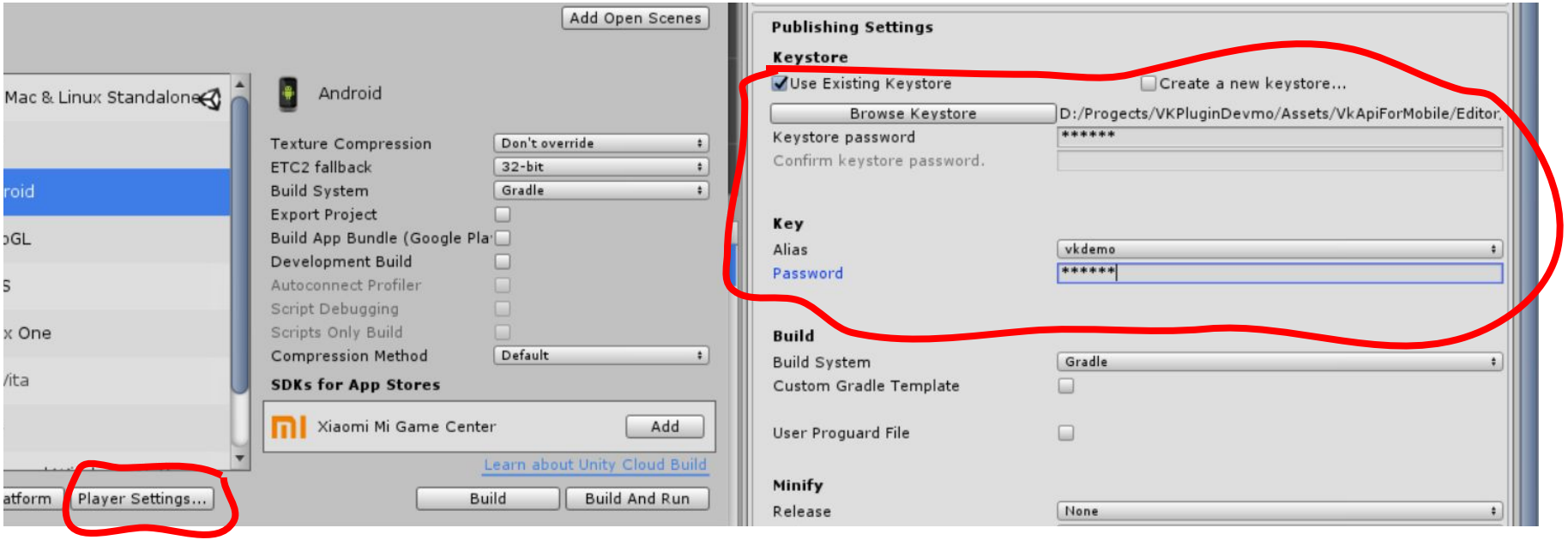
- copy sha1 value and paste it to the field **Signing certificate fingerprint for Android** on vk web portal in app setting section



- copy **Package name for Android** from app setting section on vk web portal and paste it to File->Build Settings->Player Settings->Package Name



- if your registered different than debug key fingerprint on vk web portal you also need to configure unity to use that keystore.

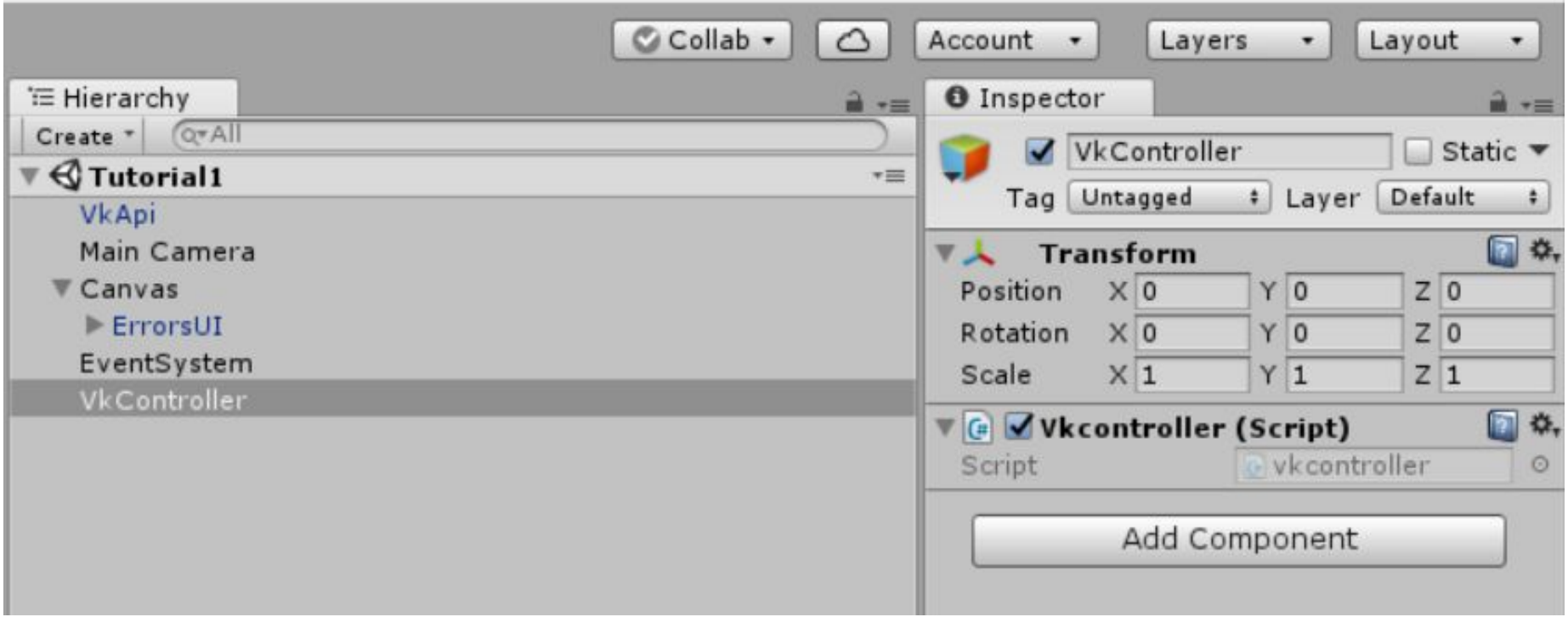


Windows and Mac Setup

After you've done Unity Editor Setup, Windows and Mac Standalone do not require any additional setup

Tutorial

- create new scene, as first element in the hierarchy add VkApiForMobile/Prefabs/VkApi
- add Canvas to your scene and add to it VkApiForMobile/Prefabs/ErrorsUI this element will handle errors where user interaction is required. (For example some calls require you to display an yes/no dialog to user). See more on https://vk.com/dev/errors
- add new empty object to your scene and rename it to VKController
- create new script VKController and add it to VKController game object.


Collab
Account
Layers
Layout
Hierarchy
Create
All
Tutorial1
VkApi
Main Camera
Canvas
ErrorsUI
EventSystem
VkController
Inspector
VkController
Static
Tag
Untagged
Layer
Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Vkcontroller (Script)
Script
vkcontroller
Add Component

- first check if user is already logged in, if yes we can call vk methods if not call login method.

Controller.cs | VKToken.cs | VKController.cs × | VkApi.cs

```
using UnityEngine.SceneManagement;
using com.playGenesis.VkUnityPlugin;
using com.playGenesis.VkUnityPlugin.MiniJSON;


0 references
public class VKController : MonoBehaviour {

    // Use this for initialization
    0 references
    void Start () {
        if (VkApi.VkApiInstance.IsUserLoggedIn) {
            GetFriendInfo();
        } else {
            VkApi.VkApiInstance.LoggedIn+=onLoggedIn;
            VkApi.VkApiInstance.Login ();
        }
    }
    2 references
    public void onLoggedIn(){
        VkApi.VkApiInstance.LoggedIn-=onLoggedIn;
        GetFriendInfo();
    }

```

- now lets create a VKRequest instance
- set url, which mimics http request(**method_name?parameter1=value1&parameter2=value2**) and callback function
- execute request
  more on https://vk.com/dev/methods

```
2 references
public void GetFriendInfo(){
    VKRequest r = new VKRequest
    {
        url="users.get?user_ids=205387401&photo_50",
        CallBackFunction=OnGotUserInfo
    };
    VkApi.VkApiInstance.Call (r);
}
1 reference
```

let’s look at the callback function

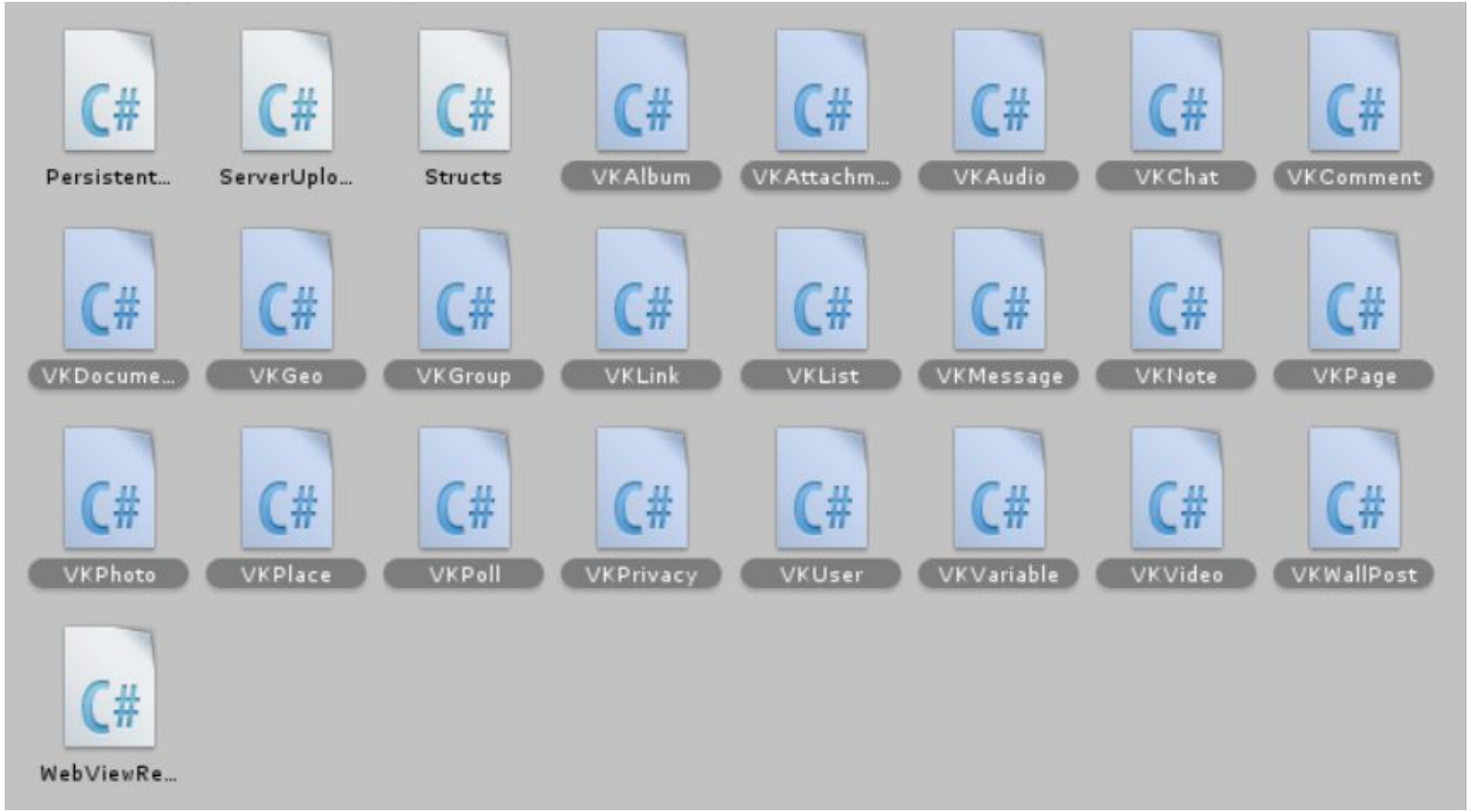
```
public void OnGotUserInfo (VKRequest r)
{
    if(r.error!=null)
    {
        if(r.error.error_code == "5"){
            SceneManager.LoadScene ("LoginScene");
        }else
            FindObjectOfType<GlobalErrorHandler>().Notification.Notify(r);
        //hande error here
        Debug.Log(r.error.error_msg);
        return;
    }

    //now we need to deserialize response in json from vk server
    var dict=Json.Deserialize(r.response) as Dictionary<string,object>;
    var users=(List<object>)dict["response"];
    for (int i = 0; i<users.Count;i++){
        var user = VKUser.Deserialize (users [0]);

        Debug.Log ("user id is " + user.id);
        Debug.Log ("user name is " + user.first_name);
        Debug.Log ("user last name is " + user.last_name);
    }
}
```

- first, check for errors: error with code 5 means invalid token, so in this case we go back to Login scene. Some vk methods return an error when user interaction is needed, for example, user needs to input captcha, this types of errors are already handled by the plugin, when you add ErrorUI prefab to your scene.
- for other types of errors here we just show user a popup with error description
- if no errors found we can deserialize json a use data

Example of usage of miniJSON

```
4   using com.playGenesis.VkUnityPlugin.MiniJSON;
5
    0 references
6   public class MiniJsonExample : MonoBehaviour {
7
8       // Use this for initialization
        0 references
9       void Start () {
10
11          var jsonString = "{ \"array\": [1.44,2,3], " +
12                              "\"object\": {\"key1\":\"value1\", \"key2\":256}, " +
13                              "\"string\": \"The quick brown fox \\\"jumps\\\" over the lazy dog \", " +
14                              "\"unicode\": \"\\u3041 Men\\u00fa sesi\\u00f3n\", " +
15                              "\"int\": 65536, " +
16                              "\"float\": 3.1415926, " +
17                              "\"bool\": true, " +
18                              "\"null\": null }";
19
20          var dict = Json.Deserialize(jsonString) as Dictionary<string, object>;
21
22          Debug.Log("deserialized: " + dict.GetType());
23          Debug.Log("dict['array'][0]: " + ((dict["array"]) as List<object>)[0]);
24          Debug.Log("dict['string']: " + dict["string"] as string);
25          Debug.Log("dict['float']: " + dict["float"]); // floats come out as doubles
26          Debug.Log("dict['int']: " + dict["int"]); // ints come out as longs
27          Debug.Log("dict['unicode']: " + dict["unicode"] as string);
28
29          var str = Json.Serialize(dict);
30
31          Debug.Log("serialized: " + str);
32      }
```

To simplify json deserialization process there are some predefined classes



So in your callback function you don’t need to drill down to each single value:

just get top level dictionary:

```
var dict=Json.Deserialize(r.response) as Dictionary<string,object>;
```

let’s take the value of “response” key and cast it to a List<object>:

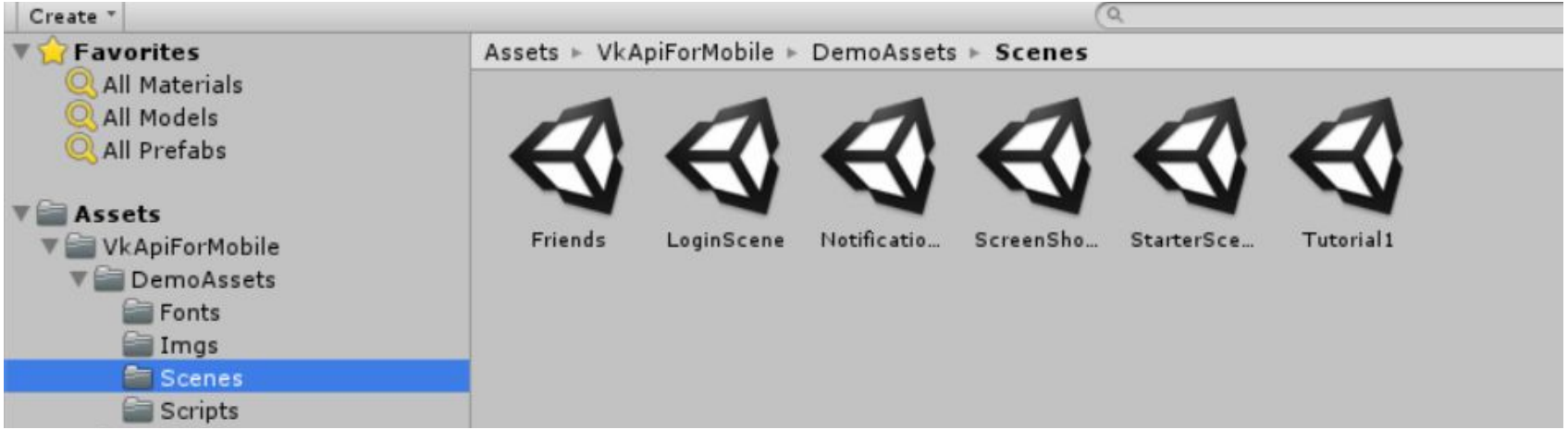
```
var users=(List<object>)dict["response"];
```

now we can deserialize vk user objects with VKUser class like this:

```
var vk_users = VKUser.Deserialize(users.ToArray());
```

You have to keep in mind that this deserialization helper classes are written for a specific version of vk api(5.21), so you may want to make some little changes to feet your needs

You’ll find this and other examples in VKApiForMobile/DemoAssets/Scenes



Api Cheat Sheet

```
get vkapi instance object
var vkapi = VkApi.VkApiInstance;

check if user is logged in
var isLoggedIn = vkapi.IsUserLoggedIn;

starts login
vkapi.Login();
```

```
get vk setting the same you can edit
selecting menu VK->Edit Vk Setting
var settings = VkApi.VkSetts;

forces usage of webview during login
settings.forceOAuth = true;

forces showing to user required
permissions during login
if false shows them only the first time
you login
settings.revoke = true;
```

```
register for authorization access denied
event
vkapi.AccessDenied += (sender, error)=>{
    Debug.Log(error.error_msg);
};

register for logged in event
vkapi.LoggedIn += ()=>{
    Debug.Log("loged in!!");
};

resiter for logged out event
vkapi.LoggedIn += ()=>{
    Debug.Log("loged out!!");
};
```

```
register for recieved new token event
vkapi.ReceivedNewToken += (s, token)=>{
    Debug.Log(token.access_token);
};

get current token data
var tokenData = VkApi.CurrentToken;

get seconds to token expiration
var tokenValidForSeconds =
tokenData.TokenValidFor();

check if token has expired
var isValid =
VKToken.IsTokenNotExpired(tokenData);
```

```
new request
VKRequest r = new VKRequest
{
     url = "users.get?user_ids=205387401",
     CallBackFunction = (request) =>{
            Debug.Log(request.response);
     }
};

execute request
vkapi.Call (r);
```

Support

For any questions you may write me an email:

vitaly.korobchuk@gmail.com

Native libraries source code:

https://drive.google.com/drive/folders/1P6fe4n3BanMbdSZvV510R4vr9ZoKFZ8D?usp=sharing