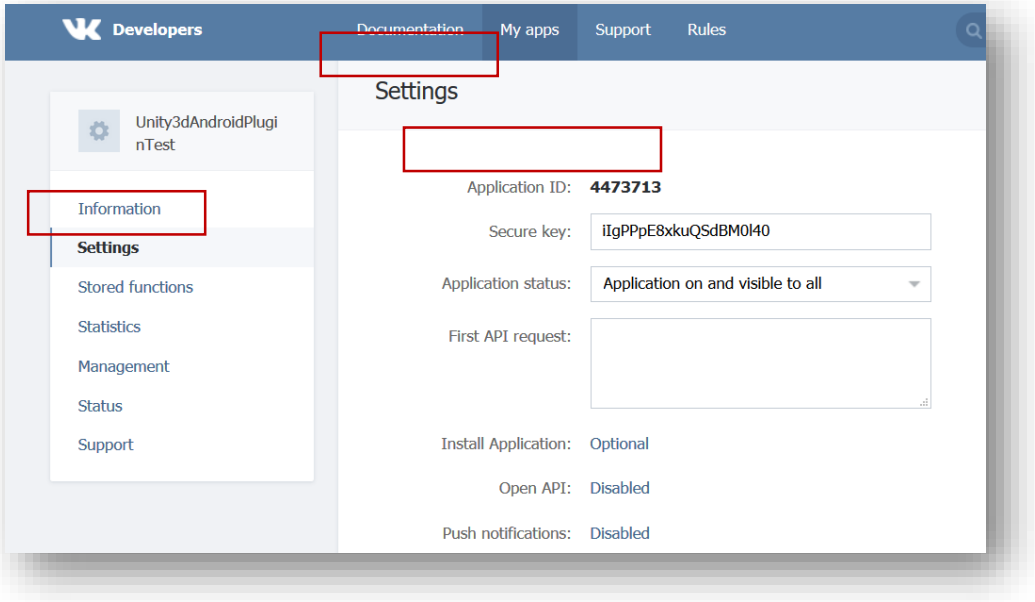
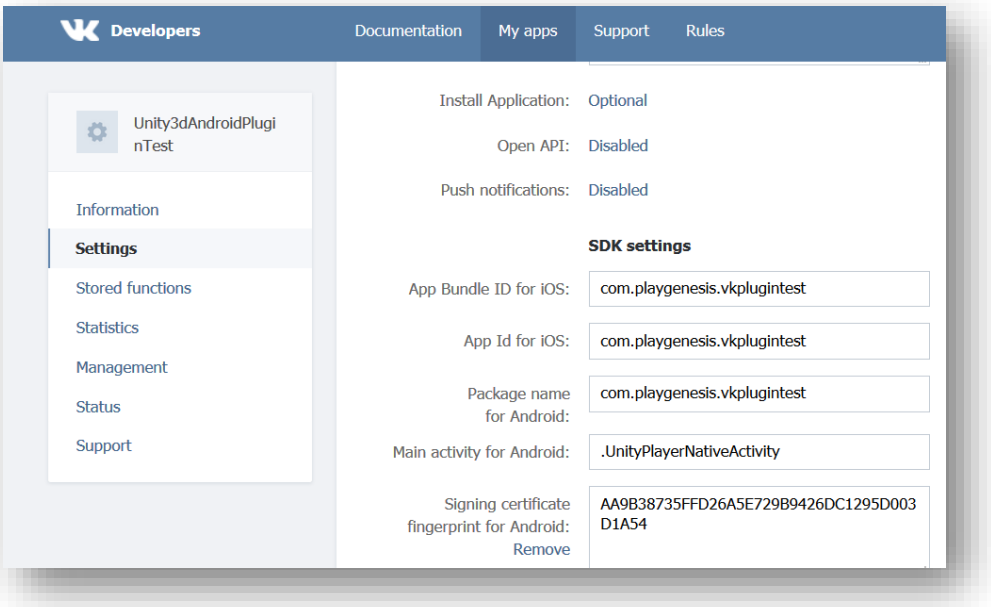
Vk Api For Mobile 2.1

Getting started

1) Setup your application on vk website at vk.com/dev. Create new application of stand-alone type. Navigate to your application setting page it and copy somewhere application id,should look like this

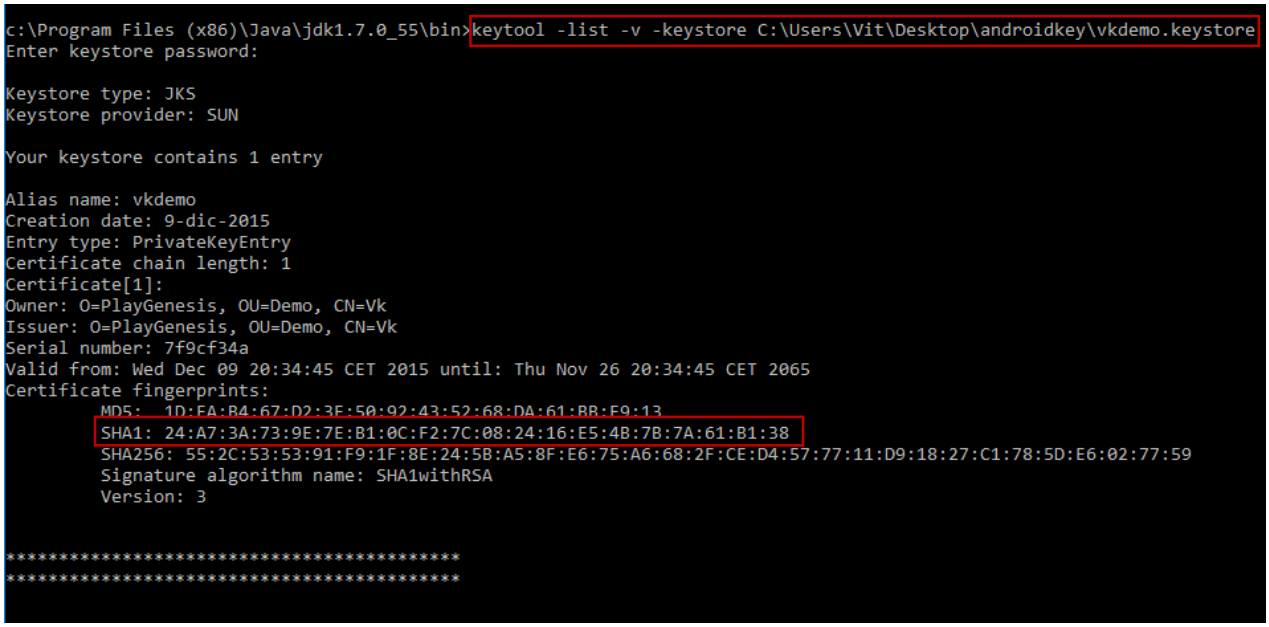


Setup bundle and key fingerprint information on the same page



In unity you should set enter the same bundle id.

To obtain android certificate fingerprint open command prompt and run keytool -list -v -keystore PathToYourKeystoreFile



About http request

Interaction with vk done by http request mechanism. In few words it is web you hit certain url and get response with some info.

Vk, Facebook etc uses this mechanism to let you access to some their user data and build applications on top of that.

First you'll need to get an access token, it is a digital signature than let the server know that the user is ok with providing you that information. We won't discuss here an authorization workflow as it is handle for you in this plugin.

Url example
https://api.vk.com/method/users.get?user_ids=205387401&photo_50&access_token=dflaksdjflaksjdflaksjdlfkas

In generic form it

https://api.vk.com/method/METHOD_NAME?PARAMETERS&access_token=YOUR_ACCESS_TOKEN

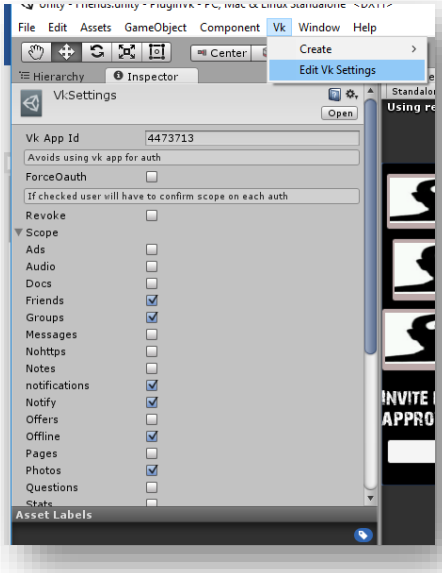
In this plugin you will need to form such a url, but first and last part will be added for you so your url will be like this

METHOD_NAME?PARAMETERS (ex. users.get?user_ids=205387401&photo_50)

Read more about methods and parameters at https://new.vk.com/dev/methods

Unity Setup

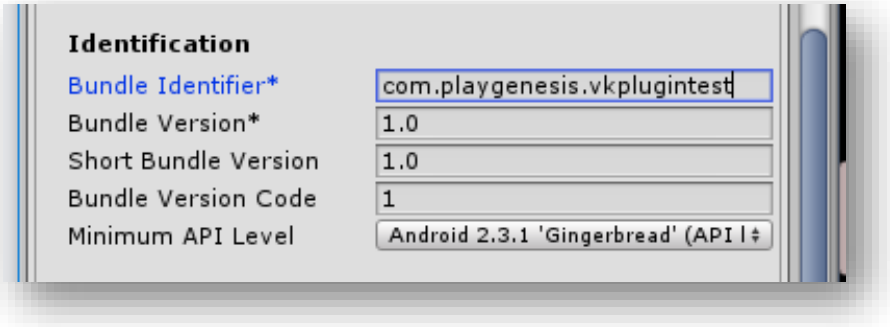
1) Open vk setting. Vk app id is the application id form vk website. **ForseOauth** will forse authorisation without using vk application, user will need to enter ther credentials on vk website. **Revoke** will forse users to confirm that they are ok with letting your app access certain kind of information. If ther confirmed previously and revoke is not checked, this step will be skiped. **Scope** is diclaration of what kind of information you want to access, will be shown to users at first authorisation, and every next time if revoke is checked.



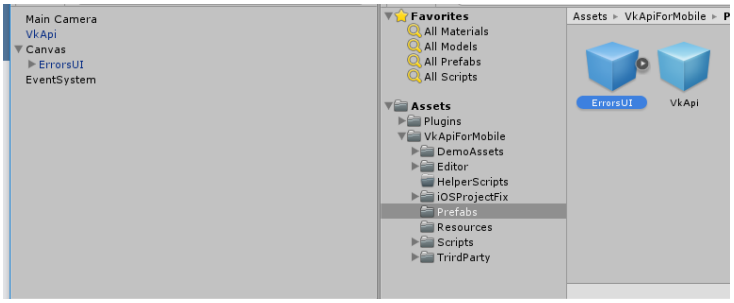
In order to work with vk from unity editor you' need to enter auth url. Hit the button «connect editor to vk». This will open a webpage. Agree to let the app access info. You will be redirected to blank page with warning not to copy the url. So copy that url and paste in unity to auth url filed.

2) Step set up bundle ids the same you've entered in vk website at your application settings page.

   For example on android



3) Add VkApi and ErrorsUI prefabs to the scene. VkApi should be higher in hierarchy than your own Game Objects using vk, so it will be initialized first. ErrorUI handle all the errors from vk server when user interaction is needed. Also it can be overlaid by other UI elements, will get on top automatically when needed. You can also customize it with you own textures and fonts so it fits your game.



Calling vk methods

```
3 using com.playGenesis.VkUnityPlugin;
4 using com.playGenesis.VkUnityPlugin.MiniJSON
```

Start Login

```
VkApi.VkApiInstance.Login ();
```

Calling vk method

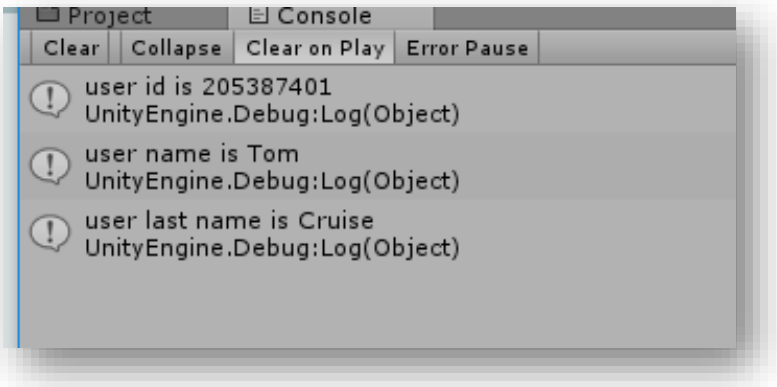
```
VKRequest r = new VKRequest
{
    url="users.get?user_ids=205387401&photo_50",
    CallBackFunction=OnGotUserInfo
};
VkApi.VkApiInstance.Call (r);
```

Let’s have a closer look at callback function

```
public void OnGotUserInfo (VKRequest r)
{
    if(r.error!=null)
    {
        //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"];
    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);
```



You can find this script and scene at Assets\VkApiForMobile\DemoAssets\Scenes

Android and iOS this is it. For Windows in order to add vk sdk look at Assets\VkApiForMobile\Editor\windows

Working with JSON


```
{
  "response": {
      "count": 717,
      "items": [
         {
            "id": 46,
            "first_name": "Andrey",
            "last_name": "Lesokhin",
            "domain": "lesokhin",
            "city": {
               "id": 2,
               "title": "Saint Petersburg"
            },
            "online": 1
         },
         {
            "id": 11191,
            "first_name": "Andrey",
            "last_name": "Lopatin",
            "domain": "kotehok",
            "city": {
               "id": 2,
               "title": "Saint Petersburg"
            },
            "online": 0
         },
         {
            "id": 172823,
            "first_name": "Andrey",
            "last_name": "Melnik",
            "domain": "a.melnik",
            "city": {
               "id": 2,
               "title": "Saint Petersburg"
            },
            "online": 1
         }
      ]
   }
}
```


A little note MiniJSON parses integers as longs and floats as doubles

So, if you need to get and integer, for example, you do id like this:
Example json {“myint”:3}
long myint=(long)dict[“myint”]

You can think about json as a dictionary with keys and values. Keys are always strings and values can be ints, floats, strings or objects or lists of objects.

“**response**” is the key followed by “:” and then the value, if value is enclosed in “{}” this mean, that it is an object with its own keys and values inside.
So how do we deserealise it with miniJSON?
First we create

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

Now let’s get value of “**response**”
I know the “**response**” is an object(remember enclosed in {})

```
var response=(object)dict[“response”];
```

Inside response object i see field “**items**”, it is an array (enclosed in[])
Each element of this array enclosed in {}, so **items** is an array of objects
Let’s get all this objects

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

It is pretty time consuming, but once you’ve reached any **vk type** it gets easy.In this example each item it items is a **VKUser** object. I’ve already defined this C# Classes for you. They have static Deserialise function, that does all the work for you. Here I’ll use linq

```
List<VKUser> friends=new List<VKUser>();
items.ForEach(i=>friends.add(VKUser.Deserialise(i));
```

Or you can do like this

```
foreach(var item in items)
{
      friends.Add(VKUser.Deserialize(item));
}
```