Introduction

Asset for working with VK SDK for iOS platform (version 1.4.6) and Android (version 2.0.2)

Unified interface for both platforms.

Supports authorization, friend list request, user information request, custom universal request.

If there is no VK application on the device, it opens authorization in the web interface.

Does not interfere with standard UnityEngine loaders.

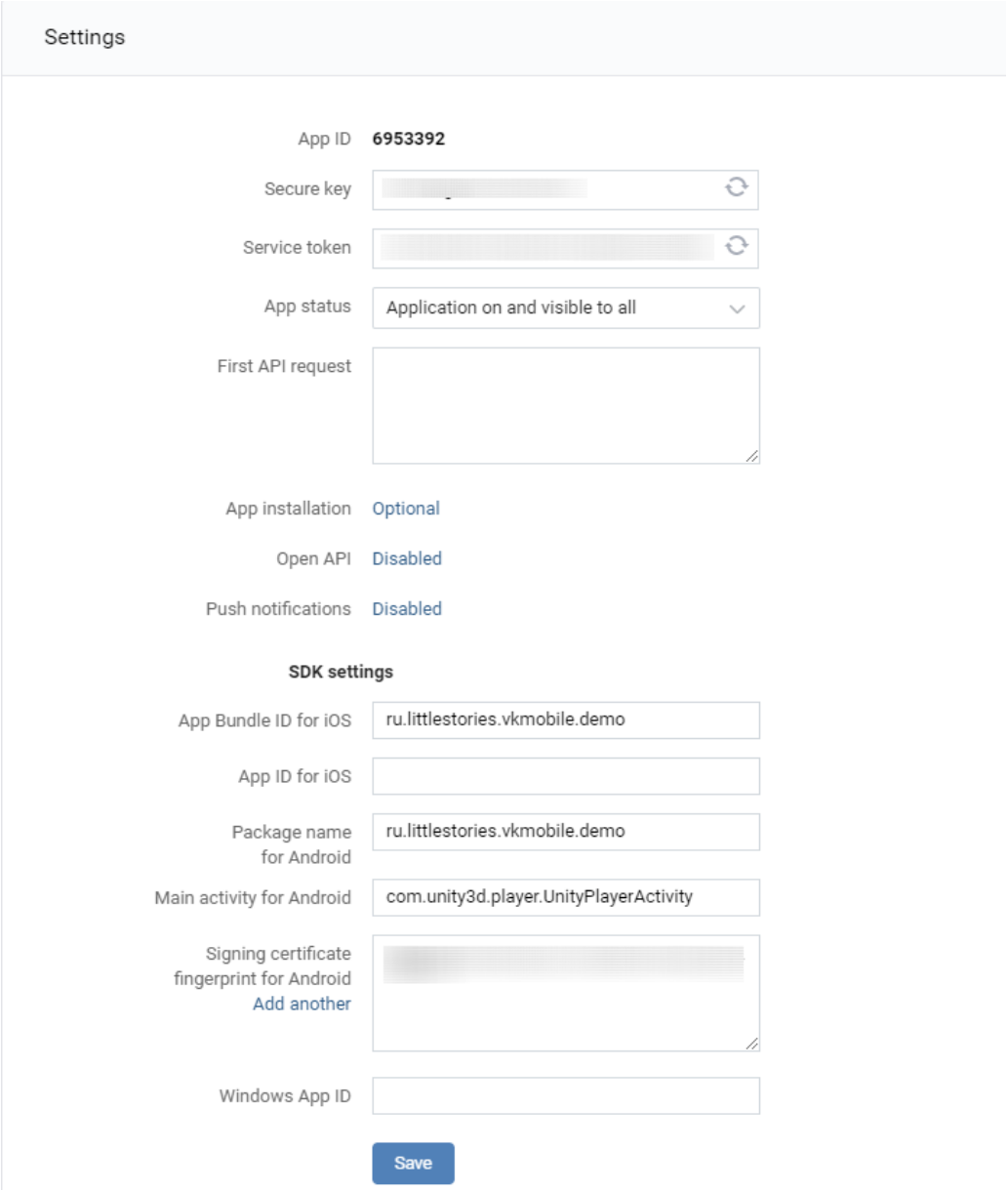
Uses Gradle and CocoaPods.

Test guide

Prepare for Using VK SDK

To use VK SDK primarily you need to create a new VK application **here** by choosing the Standalone application type. Choose a title and confirm the action via SMS and you will be redirected to the application settings page.

You will require your Application ID (referenced as **APP_ID** in the documentation). Fill in the "Batch name for Android", "Main Activity for Android" and "Certificate fingerprint for Android". Fill in the App Bundle for iOS field.



Android Certificate Fingerprint

To receive your certificate's fingerprint you can use one of the following methods.

A. Fingerprint Receiving via Keytool

1) You need to find the keystore location for your app. The *debug* store is usually located in these directories:

- ~/.android/ for OS X and Linux,
- C:\Documents and Settings\<user>\.android\ for Windows XP,
- C:\Users\<user>\.android\ for Windows Vista, Windows 7 and Windows 8.

The keystore for the release version is usually created by a programmer, so you should create it or recall its location.
2) After the keystore's location has been found, use keytool utilite (it is supplied with the Java SDK). You can get keys list with the following command:

```
keytool -exportcert -alias androiddebugkey -keystore path-to-debug-or-production-keystore -list -v
```

You will observe a similar result:

```
Certificate fingerprint: SHA1:
DA:39:A3:EE:5E:6B:4B:0D:32:55:BF:EF:95:60:18:90:AF:D8:07:09
```

By deleting all the colons you'll get your key's fingerprint.

B. Fingerprint Receiveing via VK Mobile plugin

If you've already added SDK to your project, you can use the following functions.

```
string[] fingerprints = VKMob.GetCertificateFingerprint();
```

As a rule, fingerprint contains a single line. It's a fingerprint of your certificate (depends on the certificate used for your app's signing)

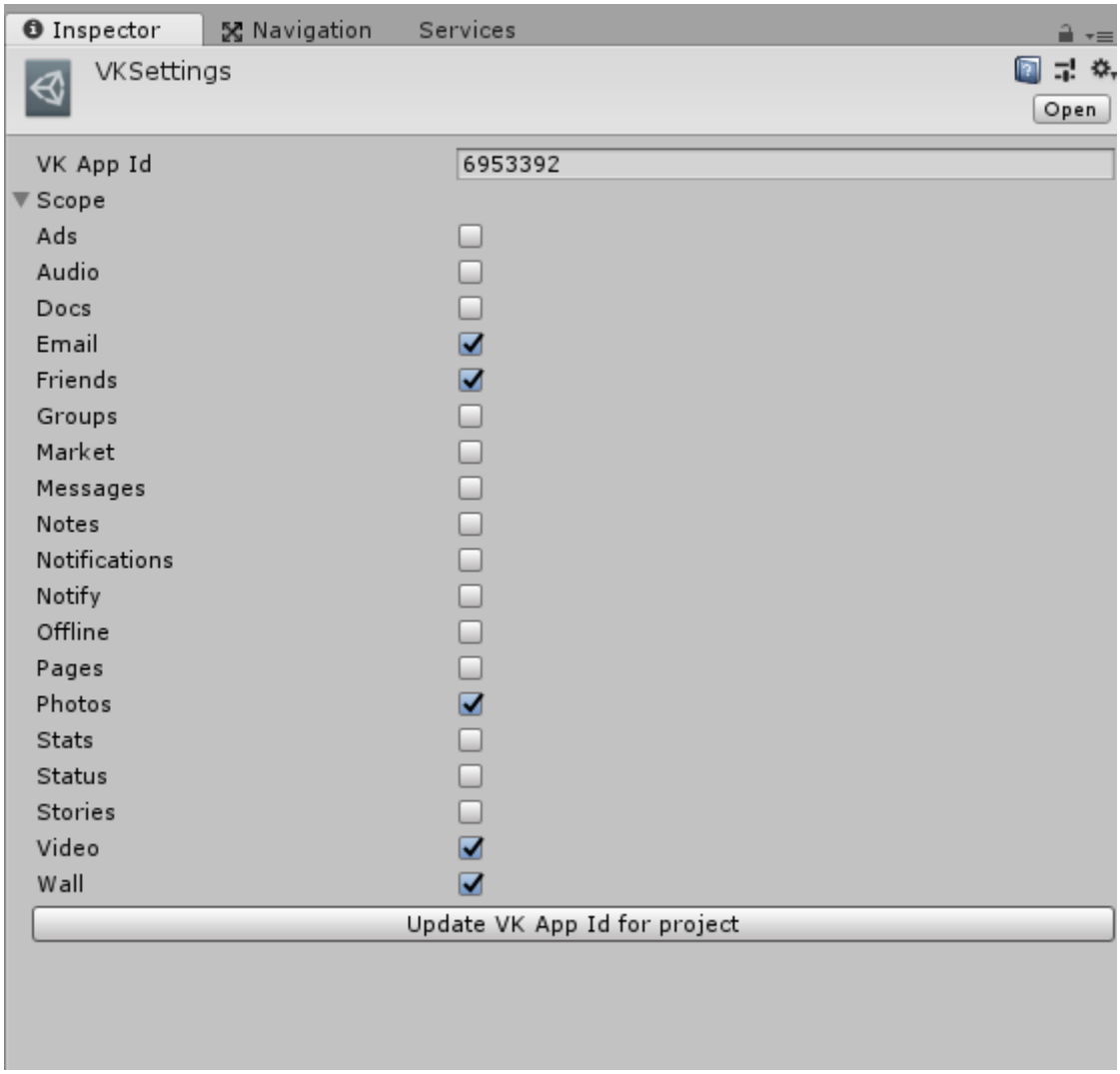
You can add more than one fingerprint in your app settings, e.g., debug and release fingerprint

Build Project

1. Create a new empty project.

2. Import the package (Assets/Import Package/Custom Package).

3. Window – VK Mobile – VK Settings. Enter **APP_ID** into field “VK App Id”. Check Scopes. Click to button “Update VK App Id for project”



4. Open and add in Build Settings demo scene Assets/VKMobile/Demo/Scenes/Demo.unity.

5.1.1 (iOS) Check iOS resolver settings. Assets – Play Services Resolver – iOS Resolver – Settings

5.1.2 (iOS) Build X-Code project, Run project in X-Code.

5.2.1 (Android) Assets – Play Services Resolver – Android Resolver - Resolve

5.2.2 (Android) Build project to android device.

Using the plugin

Access to functions on the Android and iOS is identical.

VKMob.OnLogin(LoginResult loginResult)

Set event for login. Send LoginResult.OK or LoginResult.Fail. Example:

```
VKMob.OnLogin += result =>
{
  Debug.Log("Login Result:" + result);
};
```

VKMob.OnTokenExpiredHandler()

Set event for authorization token is expired. Not worked on iOS. Example:

```
VKMob.OnTokenExpiredHandler += () =>
{
  Debug.Log("Token is expired");
};
```

void VKMob.Login()

Start login to VK SDK.

void VKMob.Logout()

Logout to VK SDK.

bool VKMob.IsLoggedIn()

If already authorized, return true.

void VKMob.Execute(string method, string parameters, Action<ResultData> callback)

In callback return result universal request. ResultData.Success - request completion status. ResultData.ValueStr – result request or error message. Example:

```
VKMob.Execute("friends.get", "{ \"fields\": \"city, online, country, photo_100\", \"order\": \"random\", \"count\" : 3 }", (result)=> { Debug.Log(result.ValueStr) });
```

All methods see here.

void RequestUsers(List<int> uids, Action<bool,VKUser[]> callback)

In callback return users info. If uids is null, return only current login user info.

void RequestFriends(int uid, Action<bool,VKUser[]> callback)

In callback return friends of user uid. If uid is 0, return only friends for current login user.

string[] VKMob.GetCertificateFingerprint();

Return fingerprint

Class VKUser

```
public class VKUser
{
   public int id = 0;
   public string first_name = "";
   public string last_name = "";
   public string photo_100 = "";
   public string photo_200 = "";
}
```