Overview

VPN Unlimited SDK for Android is shipped as a distribution of an Android Library Project. It is written in Java and C++ and is available for Android 5.0 and later.

Setup

Drag and drop VpnUnlimitedSDK.aar into your project libs folder.
In order to use it you have to add following lines of code to yours build.gradle

```
implementation 'com.keepsolid.androidkeepsolidcommon:VpnUnlimitedSDK@aar'
implementation 'com.madgag.spongycastle:bcpkix-jdk15on:1.58.0.0'
implementation 'com.madgag.spongycastle:bcpg-jdk15on:1.58.0.0'
api 'com.google.code.gson:gson:2.8.6'
api 'com.squareup.okhttp3:okhttp:3.14.9'
api 'org.conscrypt:conscrypt-android:2.2.1'
implementation 'androidx.work:work-runtime-ktx:2.7.1'
```

Add the following entities to Your ***AndroidManifest.xml*** application block, replacing the ***YOUR_APPLICATION_ID*** with your actual application ID:

```
<receiver
   android:name=".ServiceNotificationClickBroadcastReceiver"
   android:label="ServiceNotificationClickBroadcastReceiver"
   android:exported="true">
   <intent-filter>
       <action android:name="YOUR_APPLICATION_ID.VPNUService.ON_NOTIFICATION_CLICK_ACTION"
/>
   </intent-filter>
</receiver>
```

```
<activity
   android:name=".activity.BaseDialogActivity"
   android:excludeFromRecents="true"
   android:screenOrientation="portrait"
   android:theme="@style/Theme.AppCompat.Translucent"
   android:exported="true">
   <intent-filter>
       <action android:name="YOUR_APPLICATION_ID.DialogActivity.LAUNCH_INTENT_ACTION" />
       <category android:name="android.intent.category.DEFAULT" />
   </intent-filter>
</activity>
```

```
<receiver

android:name="com.keepsolid.androidkeepsolidcommon.vpnunlimitedsdk.vpn.controllers.
VPNUStateController"
    android:label="VPNUStateController"
    android:exported="true">
    <intent-filter>
        <action
android:name="com.keepsolid.vpnunlimitedsample.OpenVpnService.OPEN_VPN_STATE_BROADC
AST_ACTION" />
```

```
        <action
android:name="com.keepsolid.vpnunlimitedsample.OpenVpnService.ENABLE_VPN_CONNECTION
_BROADCAST_ACTION" />
        <action
android:name="com.keepsolid.vpnunlimitedsample.OpenVpnService.DISABLE_VPN_CONNECTIO
N_BROADCAST_ACTION" />
        <action
android:name="com.keepsolid.vpnunlimitedsample.OpenVpnService.CHECK_SERVICE_CONNECT
ION_BROADCAST_ACTION" />
        <action android:name="android.intent.action.BOOT_COMPLETED" />
    </intent-filter>
</receiver>
```

Create a class for BaseDialog Activity. It should be empty activity class just as in example below::

```
package YOUR_PACKAGE_NAME.activity;

import android.os.Bundle;

import
com.keepsolid.androidkeepsolidcommon.vpnunlimitedsdk.api.dialogs.KSDefaultDialogActivity;


public class BaseDialogActivity extends KSDefaultDialogActivity {

   @Override
   protected void onCreate(Bundle bundle) {
       super.onCreate(bundle);
   }
}
```

For customization purposes You also need to create String Provider class that will be used in initialization process lately. Create class similar to this one (You can find same example class inside Sample VPNU SDK project:

```
package YOUR_PROJECT_ID.utils;

import androidx.annotation.NonNull;
import com.keepsolid.androidkeepsolidcommon.commonsdk.utils.stringutils.KSStringProvider;

public class SampleStringProvider extends KSStringProvider {
   @Override
   public String getUserOffice2FaLink() { return null;}

   @Override
   public String getAgeCertificationLink() { return null; }

   @NonNull
   @Override
   public String getCompanyName() { return "YOUR COMPANY NAME HERE"; //  required }

   @NonNull
   @Override
   public String getProductName() {
```

```java
        return "YOUR PRODUCT NAME HERE"; //required
    }

    @NonNull
    @Override
    public String getSupportAddress() {
        return "support@domain.com"; // required
    }

    @Override
    public String getFeedbackAddress() {
        return null;
    }

    @Override
    public String getPrivacyLink() {
        return null;
    }

    @Override
    public String getEuaLink() {
        return null;
    }

    @Override
    public String getTellFriendLinks() {
        return null;
    }

    @Override
    public String getFAQLink() {
        return null;
    }

    @Override
    public String getDownloadsLink() {
        return null;
    }

    @Override
    public String getAppWebsiteLink() {
        return null;
    }

    @Override
    public String getAppShareLink() {
        return null;
    }

    @Override
    public String getDataUsageLink() {
        return null;
    }
}
```

All fields marked as “required” must be filed in.

Initializing SDK

Use the following setup code in your Application class onCreate() method to initialize VPN Unlimited SDK using your App ID, App Secret and your app version:

```
VPNUFacade.getInstance().prepare(getApplicationContext(), "YOUR_APPLICATION_ID",
"YOUR_APPLICATION_SECRET", "YOUR_APP_VERSION");
```

Don't forget to initialize string provider with class described above:

```
KSStringProviderManager.getInstance().init(new SampleStringProvider());
```

After following this steps You will have complete setted-up SDK and ready to use it for Your purposes.

Multithreading note!

All network calls in this SDK are performed synchronously. That means that you should perform them all on a non-main thread, otherwise it would block it and never come back. Every method which falls into this category is explicitly documented.

FAQ:

Q: How to perform login without registration (guest-login flow)?
A:
Guest login makes it possible to authorize user without credentials. It uses user's device ID to generate authorization data, so any change in device ID may lead to losing user's "fake" account. You can perform fake login with this method:

```
public KSAccountUserInfo doGuestLogin(String companyDomain) throws KSException {
    KSAccountUserInfo userInfo;
    try {
        userInfo =
VPNUFacade.getInstance().getAuthorizer().authorizeWithTemporaryLogin(companyDomain)
;
    } catch (KSException e) {
        if (e.getResponse().getResponseCode() ==
KSResponse.KS_ERROR_INVALID_CREDENTIALS) {
            userInfo =
VPNUFacade.getInstance().getAuthorizer().registerWithTemporaryLogin(companyDomain,
null);
        } else {
            throw e;
        }
    }

return userInfo;
```

!!!! You need to provide the **company domain as a parameter to this method** for proper account generation. Please, use only your real domain name. The fake domains will be rejected by our back-end.!!!!

Q: How to authorize user?

A:
First, setup VPNUAuthorizer instance.

```
VPNUAuthorizer vpnuAuthorizer = VPNUFacade.getInstance().getVpnuAuthorizer();
```

Then, you can log in using user's credentials:

```
String login = "login@some.domain";
String password = "password";
try {
    VPNUAccountUserInfo userInfo = vpnuAuthorizer.authorizeWithLogin(login,
password);
} catch (VPNUException exception) {
    Log.v(LOG_TAG, "authorization failed with error code: " +
exception.getResponse().getResponseCode()
             + ", and error msg " + exception.getResponse().getResponseMessage());
    exception.printStackTrace();
}
```

If authorization is successful, the VPNUTransport instance is managing the newly created session and reconnects automatically when this session is expired. Retrieved authorization info is saved into VPNUAccountUserInfo

Q: How to register a new user?
A:
First, setup VPNUAuthorizer instance.

```
VPNUAuthorizer vpnuAuthorizer = VPNUFacade.getInstance().getVpnuAuthorizer();
```

Then, you can register new user using user's credentials. VPNUAuthorizer#registerWithLogin will automatically log in a newly created user.

```
String login = "login@some.domain";
String password = "password";
try {
    VPNUAccountUserInfo userInfo = vpnuAuthorizer.registerWithLogin(login,
password, true
);
} catch (VPNUException exception) {
    Log.v(LOG_TAG, "registration failed with error code: " +
exception.getResponse().getResponseCode()
             + ", and error msg " + exception.getResponse().getResponseMessage());
    exception.printStackTrace();
}
```

Q: How to log out a logged in user?
A:
You have to be logged in in order to perform this operation successfully. See two previous answers.
boolean signOutSuccessful = false;

```
try {
    signOutSuccessful = vpnuAuthorizer.signOut();
} catch (VPNUException exception) {
    Log.v(LOG_TAG, "Log out failed with error code: " +
exception.getResponse().getResponseCode()
```

```
                 + ", and error msg " +
exception.getResponse().getResponseMessage());
    exception.printStackTrace();
}
```

Q: How to get account status for a logged in user?
A:
Assuming you are logged in:

```
try {
    VPNUAccountManager vpnuAccountManager =
VPNUFacade.getInstance().getVpnuAccountManager();
    VPNUAccountStatus accountStatus = vpnuAccountManager.getStatus();
} catch (VPNUException exception) {
    Log.v(LOG_TAG, "Status loading failed with error code: " +
exception.getResponse().getResponseCode()
        + ", and error msg " + exception.getResponse().getResponseMessage());
    exception.printStackTrace();
}
```

Q: How to get list of available VPN Servers?
A:

```
try {
    VPNUServersManager vpnuServersManager = VPNUFacade.getInstance().getVpnuServersManager();
    List<VPNUServer>  servers = vpnuServersManager.getServers();
} catch (VPNUException exception) {
    Log.v(LOG_TAG, "Servers list loading failed with error code: " +
exception.getResponse().getResponseCode()
        + ", and error msg " + exception.getResponse().getResponseMessage());
    exception.printStackTrace();
}
```

Q: How to connect to the selected VPN Server?
A:
First, setup VPNUConfigurator instance.

```
VPNUConfigurator vpnuConfigurator = VPNUFacade.getInstance().getVpnuConfigurator();
```

Then you can setup the selected server item (see previous question on how to retrieve it). You can choose protocol settings for VPN connection by passing VPNUProtoConfig entity as a parameter to the VPNUConfigurator#setup method.

```
try {
    VPNUConfigurator vpnuConfigurator =
VPNUFacade.getInstance().getVpnuConfigurator();
    vpnuConfigurator.prepare();
    vpnuConfigurator.setup(server, new VPNUProtoConfig(%PROTO CONSTRUCT HERE%);
} catch (VPNUException exception) {
    Log.v(LOG_TAG, "set up of Vpn failed with error code: " +
exception.getResponse().getResponseCode()
    + ", and error msg " + exception.getResponse().getResponseMessage());
    exception.printStackTrace();
}
```

You then can try to connect to a selected server. In order to obtain the connection status, use the VpnStatusChangedListener by passing an instance to the ***VPNUConfigurator#addOnStatusChangedListener*** method of VPNUConfigurator. This method blocks a calling thread.

```
try {
    VPNUConfigurator vpnuConfigurator =
VPNUFacade.getInstance().getVpnuConfigurator();
    vpnuConfigurator.startVpn();
} catch (VPNUException exception) {
    Log.v(LOG_TAG, "start Vpn failed with error code: " +
exception.getResponse().getResponseCode()
    + ", and error msg " + exception.getResponse().getResponseMessage());
    exception.printStackTrace();
}
```

Q: How to stop the VPN connection?
A:

```
try {
    VPNUConfigurator vpnuConfigurator = VPNUFacade.getInstance().getVpnuConfigurator();
    vpnuConfigurator.stopVpn();
} catch (VPNUException exception) {
    Log.v(LOG_TAG, "Stop Vpn failed with error code: " + exception.getResponse().getResponseCode()
    + ", and error msg " + exception.getResponse().getResponseMessage());
    exception.printStackTrace();
}
```