Unique Computer Systems LLC

P.O. Box: 21419
Sharjah, U.A.E.
Tel: (971) 6 5254491
Fax: (971) 6 5254343
E-Mail: info@ucssolutions.com
www.ucssolutions.com

Reson*8* Retriever API integration with Android Applications

Version No: 1.0

Date Updated: 23/03/2020

Prepared By: Vaisak Priyadersy

Unique Computer Systems LLC

P.O. Box: 21419
Sharjah, U.A.E.
Tel: (971) 6 5254491
Fax: (971) 6 5254343
E-Mail: info@ucssolutions.com
www.ucssolutions.com

Contents

Unique Computer Systems LLC

P.O. Box: 21419
Sharjah, U.A.E.
Tel: (971) 6 5254491
Fax: (971) 6 5254343
E-Mail: info@ucssolutions.com
www.ucssolutions.com

Introduction

With the SMS Retriever API, you can perform SMS-based user verification in your Android app automatically, without requiring the user to manually type verification codes, and without requiring any extra app permissions.

Purpose

This document shows steps to integrate SMS Retriever API in your android application along with sample codes.

Work flow

- The App requests your server to send OTP with application Hash-Key and other required parameters.
- Your app calls the SMS Retriever API at the same time and listening for an SMS from your server.
- Your server sends an SMS message that includes a verification code and a hash to identify your app.
- When user's device receives the SMS message, the SMS Retriever API reads the SMS in your app.
- Parse the received text message in your BroadCastReciever's onReceive() method. You can use an interface listener to send the message text back to your Activity/Fragment class.

Message Format from Server

The standard SMS format is given blow:

```
<#> Your ExampleApp code is: 123ABC78
FA+9qCX9VSu
```

SMS always starts with <#> sign and have a hash key like FA+9qCX9VSu to identify your app it is generated with your app's package id. You just need to get this has key from app and share with your server. In next few steps you will see how to create hash keys.

P.O. Box: 21419
Sharjah, U.A.E.
Tel: (971) 6 5254491
Fax: (971) 6 5254343
E-Mail: info@ucssolutions.com
www.ucssolutions.com

Dependencies

Below given are the dependencies used for Retriever API:

```
implementation 'com.google.android.gms:play-services-auth:17.0.0'
implementation 'com.google.android.gms:play-services-auth-api-phone:17.4.0'
```

Integration steps

1. Use AppSignatureHashHelper class (given below) to get Hash key associated with your app as per your package id.

   This is only required to get your app's hash key it would always be same unless you are changing app's package id or signature.

Unique Computer Systems LLC

P.O. Box: 21419
Sharjah, U.A.E.
Tel: (971) 6 5254491
Fax: (971) 6 5254343
E-Mail: info@ucssolutions.com
www.ucssolutions.com

```
import android.annotation.TargetApi;
import android.content.Context;
import android.content.ContextWrapper;
import android.content.pm.PackageManager;
import android.content.pm.Signature;
import android.util.Base64;
import android.util.Log;

import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.Arrays;

public class AppSignatureHashHelper extends ContextWrapper {
    public static final String TAG =
AppSignatureHashHelper.class.getSimpleName();

    private static final String HASH_TYPE = "SHA-256";
    public static final int NUM_HASHED_BYTES = 9;
    public static final int NUM_BASE64_CHAR = 11;

    public AppSignatureHashHelper(Context context) {
        super(context);
    }

    /**
     * Get all the app signatures for the current package
     *
     * @return
     */
    public ArrayList<String> getAppSignatures() {
        ArrayList<String> appSignaturesHashs = new ArrayList<>();

        try {
            // Get all package details
            String packageName = getPackageName();
            PackageManager packageManager = getPackageManager();
            Signature[] signatures =
packageManager.getPackageInfo(packageName,
                    PackageManager.GET_SIGNATURES).signatures;

            for (Signature signature : signatures) {
                String hash = hash(packageName, signature.toCharsString());
                if (hash != null) {
                    appSignaturesHashs.add(String.format("%s", hash));
                }
            }
        } catch (Exception e) {
            Log.e(TAG, "Package not found", e);
        }
        return appSignaturesHashs;
    }

    @TargetApi(19)
```

P.O. Box: 21419
Sharjah, U.A.E.
Tel: (971) 6 5254491
Fax: (971) 6 5254343
E-Mail: info@ucssolutions.com
www.ucssolutions.com

2. Declare BroadCastReceiver in your app's manifest file in side application tag.

```
<receiver android:name=".MySMSBroadcastReceiver" android:exported="true"

android:permission="com.google.android.gms.auth.api.phone.permission.SEN
D">
    <intent-filter>
        <action
android:name="com.google.android.gms.auth.api.phone.SMS_RETRIEVED"/>
    </intent-filter>
</receiver>
```

P.O. Box: 21419
Sharjah, U.A.E.
Tel: (971) 6 5254491
Fax: (971) 6 5254343
E-Mail: info@ucssolutions.com
www.ucssolutions.com

3. Create BroadCastReceiver listener and initiate SmsRetrieverClient in your MainActivity.

```
   private void initSmsRetrieverAPI() {

        MySMSBroadcastReceiver  smsReceiver = new
MySMSBroadcastReceiver();
        smsReceiver.setOTPListener(this);

        IntentFilter intentFilter = new IntentFilter();
        intentFilter.addAction(SmsRetriever.SMS_RETRIEVED_ACTION);
        this.registerReceiver(smsReceiver, intentFilter);

        // Get an instance of SmsRetrieverClient, used to start
listening for a matching
// SMS message.
        SmsRetrieverClient client = SmsRetriever.getClient(this /*
context */);

// Starts SmsRetriever, which waits for ONE matching SMS message until
timeout
// (5 minutes). The matching SMS message will be sent via a Broadcast
Intent with
// action SmsRetriever#SMS_RETRIEVED_ACTION.
        Task<Void> task = client.startSmsRetriever();

// Listen for success/failure of the start Task. If in a background
thread, this
// can be made blocking using Tasks.await(task, [timeout]);
        task.addOnSuccessListener(new OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                // Successfully started retriever, expect broadcast
intent
                // ...
                Log.d(TAG, "Retriever API onSuccess");
            }
        });

        task.addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                // Failed to start retriever, inspect Exception for more
details
                // ...
                Log.d(TAG, "Retriever API onFailure");
            }
        });

    }
```

P.O. Box: 21419
Sharjah, U.A.E.
Tel: (971) 6 5254491
Fax: (971) 6 5254343
E-Mail: info@ucssolutions.com
www.ucssolutions.com

4. You will receive OTP in call back methods implemented in you Activity/Fragment.

```
@Override
public void onOTPReceived(String otp) {
    showToast("OTP Received: " + otp);
    if (smsReceiver != null) {
        unregisterReceiver(smsReceiver);
        smsReceiver = null;
    }
}

@Override
public void onOTPTimeOut() {
    showToast("OTP Time out");
}

@Override
public void onOTPReceivedError(String error) {
    showToast(error);
}
```

Unique Computer Systems LLC

P.O. Box: 21419
Sharjah, U.A.E.
Tel: (971) 6 5254491
Fax: (971) 6 5254343
E-Mail: info@ucssolutions.com
www.ucssolutions.com

5. Sample code for BroadCastReciever is given below:

```
public class MySMSBroadcastReceiver extends BroadcastReceiver {
    private String TAG =  MySMSBroadcastReceiver.class.getSimpleName();
    private OTPReceiveListener otpListener;

    public void setOTPListener(OTPReceiveListener otpListener) {
        this.otpListener = otpListener;
    }

    @Override
    public void onReceive(Context context, Intent intent) {
        if (SmsRetriever.SMS_RETRIEVED_ACTION.equals(intent.getAction())) {
            Bundle extras = intent.getExtras();
            Status status = (Status) extras.get(SmsRetriever.EXTRA_STATUS);
            Log.d(TAG, "Status Code :: "+status.getStatusCode());

            switch(status.getStatusCode()) {

                case CommonStatusCodes.SUCCESS:
                    // Get SMS message contents
                    String message = (String)
extras.get(SmsRetriever.EXTRA_SMS_MESSAGE);
                    // Extract one-time code from the message and complete
verification
                    // by sending the code back to your server.
                    if (otpListener != null) {
                        otpListener.onOTPReceived(message);
                    }
                    Log.d(TAG, message);
                    break;

                case CommonStatusCodes.TIMEOUT:
                    // Waiting for SMS timed out (5 minutes)
                    // Handle the error ...
                    if (otpListener != null) {
                        otpListener.onOTPTimeOut();
                    }
                    break;

                case CommonStatusCodes.API_NOT_CONNECTED:

                    if (otpListener != null) {
                        otpListener.onOTPReceivedError("API NOT CONNECTED");
                    }
                    break;

                case CommonStatusCodes.NETWORK_ERROR:

                    if (otpListener != null) {
                        otpListener.onOTPReceivedError("NETWORK ERROR");
                    }

                    break;

                case CommonStatusCodes.ERROR:
```

P.O. Box: 21419
Sharjah, U.A.E.
Tel: (971) 6 5254491
Fax: (971) 6 5254343
E-Mail: info@ucssolutions.com
www.ucssolutions.com

6. Sample of Interface for listeners is given below:

```
public interface OTPReceiveListener {
    void onOTPReceived(String otp);

    void onOTPTimeOut();

    void onOTPReceivedError(String error);
}
```

Support Info

Please feel free to contact us for any support while testing with our Retriever API.

Email: reson8support@ucssolutions.com

Tel: +971 6 525 4491

Sunday to Thursday 9:00 AM to 6:00 PM