

# Programming in Android



**Session: 10**

**Web Services in Android**

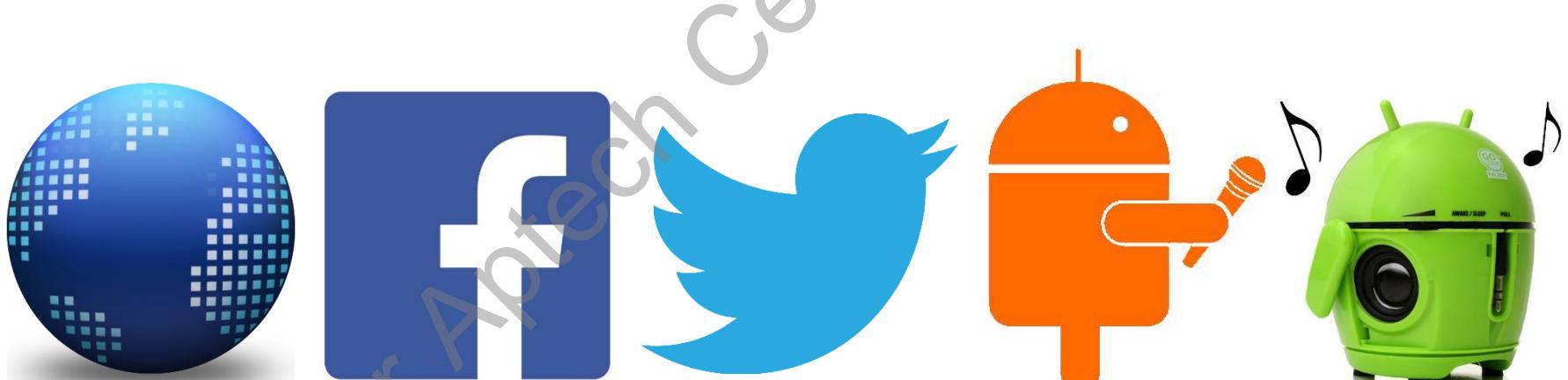
# Objectives

- ◆ Explain and use Web Services
- ◆ Explain standards for Web Services Communication
- ◆ Use third party API
- ◆ Create Web Services
- ◆ Understand the correct use of Web Service
- ◆ Explain and use Text to Speech and Voice Recognition



# Introduction

- ◆ Web Services can enhance the application by providing computational offloading and information services
- ◆ Text to Speech and Voice Recognition provide unique and efficient ways for interacting with the application
- ◆ Coupled together, the developer can make an extremely user friendly and easy to use applications



# Web Services

- ◆ Web Service is an application component which allows communication between devices over a network
- ◆ There are several platforms which provide useful resources which can be utilized in developing Android Applications
- ◆ This section describes the process of utilizing third party Web services, along with the various standards



# Types of Web Services 1-2

## ◆ ReSTful

- ❖ ReST is the abbreviation for Representational State Transfer
- ❖ There are no concrete standards for implementing a Web service using ReST
- ❖ Requests are sent as standard HTTP ‘GET’ requests and the reply can be in any format that is mutually understandable among the communicating parties

## ◆ SOAP

- ❖ SOAP is an abbreviation for Simple Object Access Protocol
- ❖ SOAP is an XML based Remote Procedure Call protocol. It relies on application layer protocols such as HTTP and SMTP
- ❖ The client executes a remote method (procedure) on the Web server using an XML envelope to send the request
- ❖ The Web service generates a response and sends it back to the client in an XML envelope

## Types of Web Services 2-2

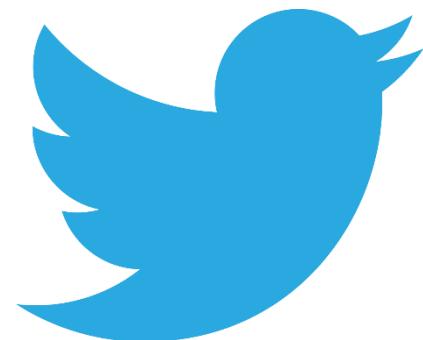
- ◆ **Custom/Proprietary**

- ◆ Besides the two open standards, several Web services provide custom formats for communication with the Web service
- ◆ These Web services usually come with extensive documentation on using these services or with libraries for use with the platforms

For Aptech Centre Use Only

## Using Third Party API

- ◆ There are other API providers besides Google who provide similar or better functionality
- ◆ If the application is using authentication, then it will be a wise decision to integrate Facebook login into applications
- ◆ This reduces the hassle for registration and provides a trustable authentication system for the application
- ◆ It is also possible to use social features from Facebook such as friend's lists and post to wall



## Generating a Hash Key 1-2

- ◆ Generate an Application key for signing the application
- ◆ Download OpenSSL for Windows from the URL  
[https://code.google.com/p/openssl-for-windows/downloads/detail?name=openssl-0.9.8k\\_WIN32.zip&can=2&q](https://code.google.com/p/openssl-for-windows/downloads/detail?name=openssl-0.9.8k_WIN32.zip&can=2&q)
- ◆ Extract the contents of the zip file to C:\openssl
- ◆ Navigate to the keytool
- ◆ Enter the command as shown in the following Code Snippet:

```
keytool -exportcert -alias "aptech key" -keystore "C:\AndroidApps\keystore.jks"  
| "C:\openssl\bin\openssl" sha1 -binary | "C:\openssl\bin\openssl" base64
```

## Generating a Hash Key 2-2

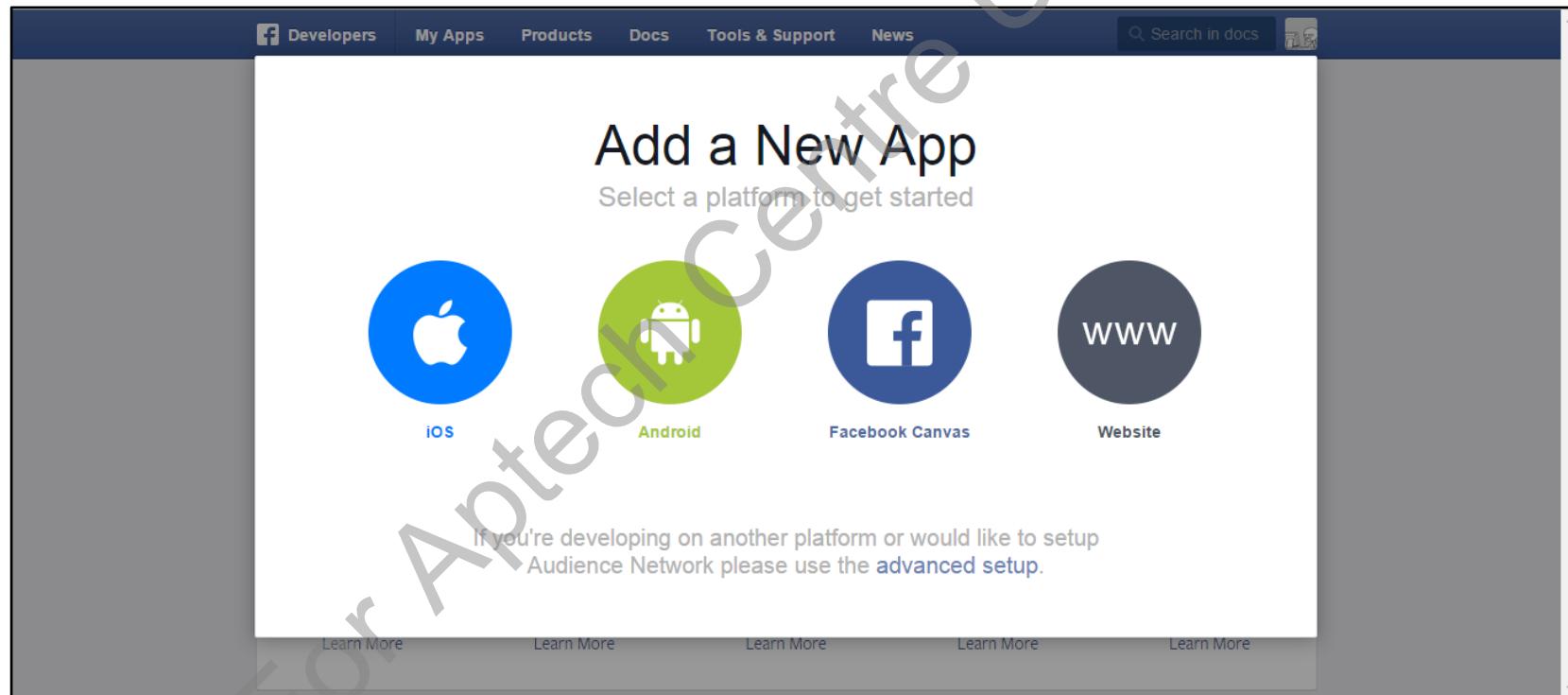
- ◆ Enter the password
- ◆ The key is displayed as shown in the following figure:

```
C:\Program Files\Java\jdk1.8.0_40\bin>keytool -exportcert -alias "aptech key"  
keystore "C:\Android\app\keystore.jks" | "C:\openssl\bin\openssl" sha1 -binary  
"C:\openssl\bin\openssl" base64  
Enter keystore password: pressure  
6L5/p10Qzr8=
```

- ◆ Store the key value

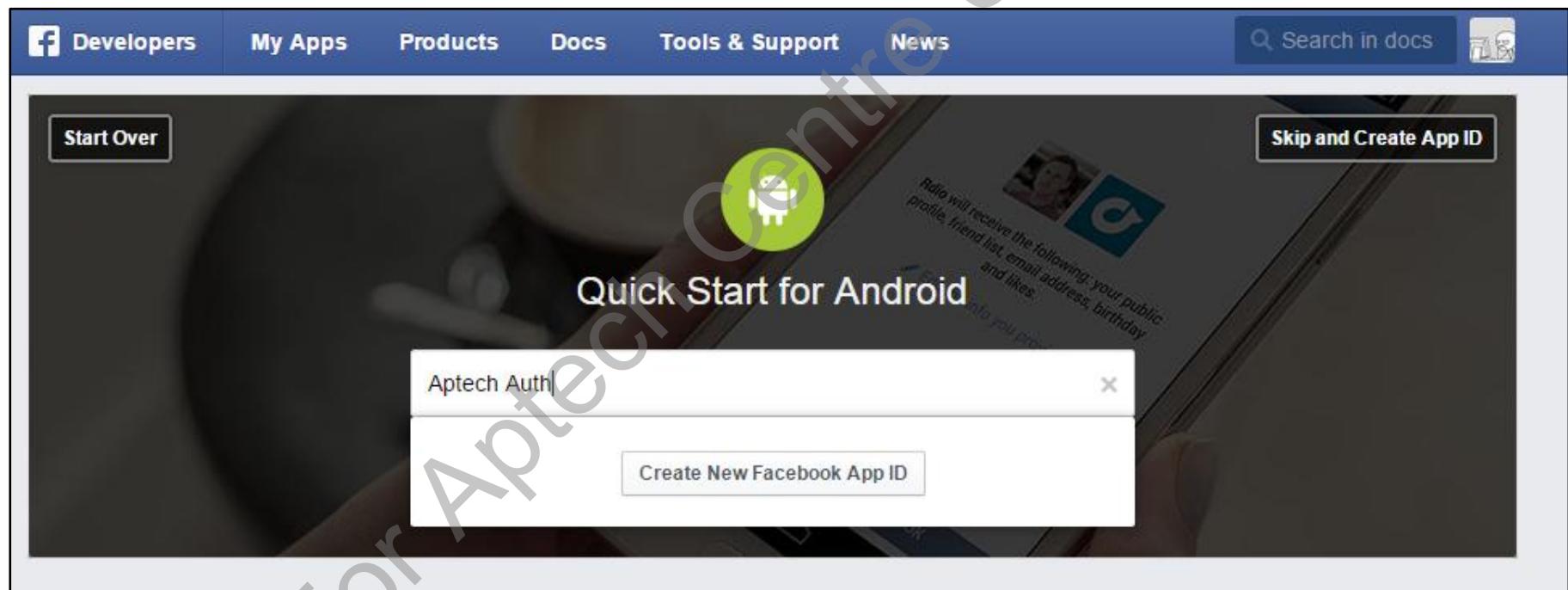
# Generating an API Key 1-8

- Open <https://developers.facebook.com> and enable developer account
- From the My Apps menu, select Add a New App. The app type selection screen is displayed as shown in the following figure:



# Generating an API Key 2-8

- Select Android and enter the app name as Aptech Auth and select Create New Facebook App ID as shown in the following figure:



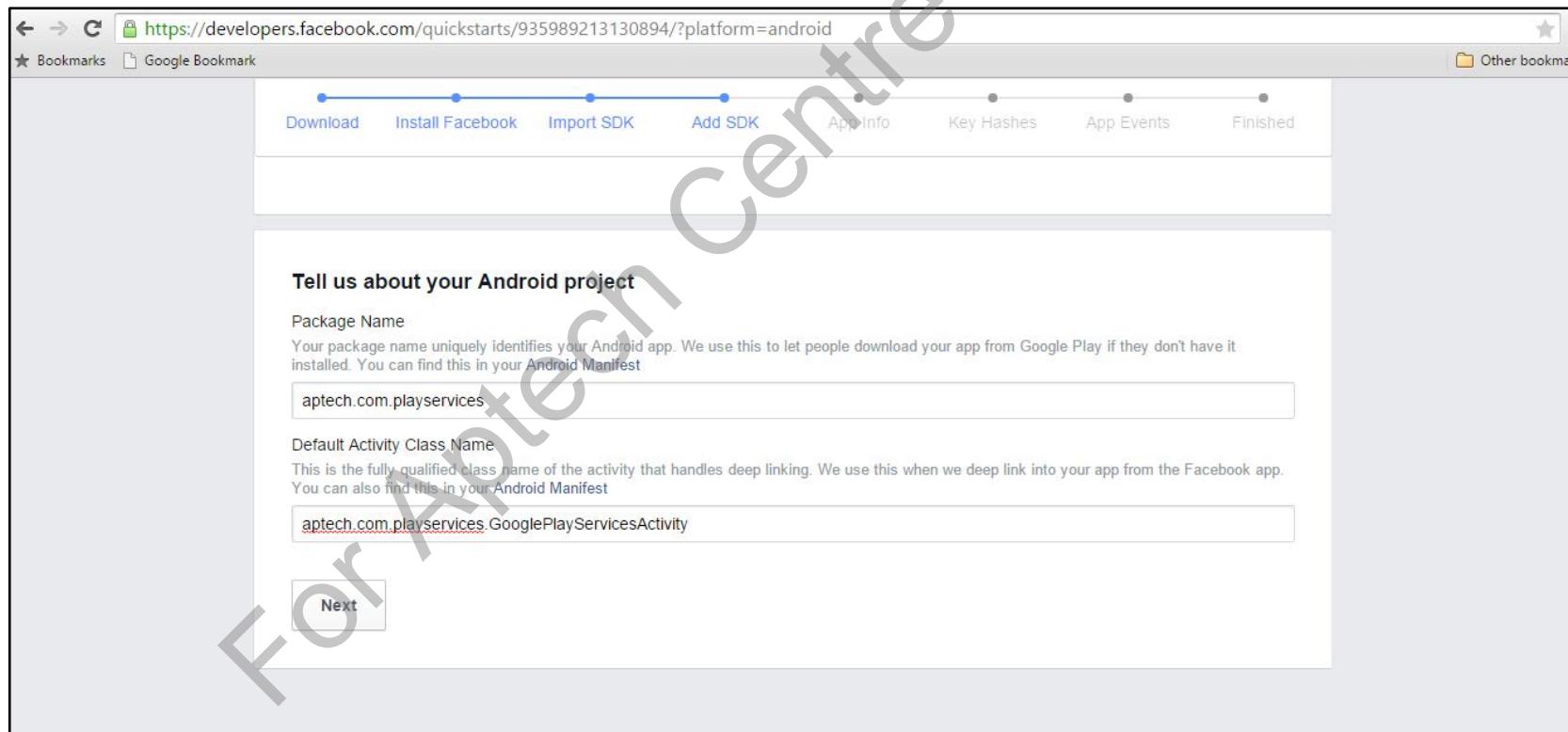
## Generating an API Key 3-8

- Select a category and click Create a New App ID as shown in the following figure:



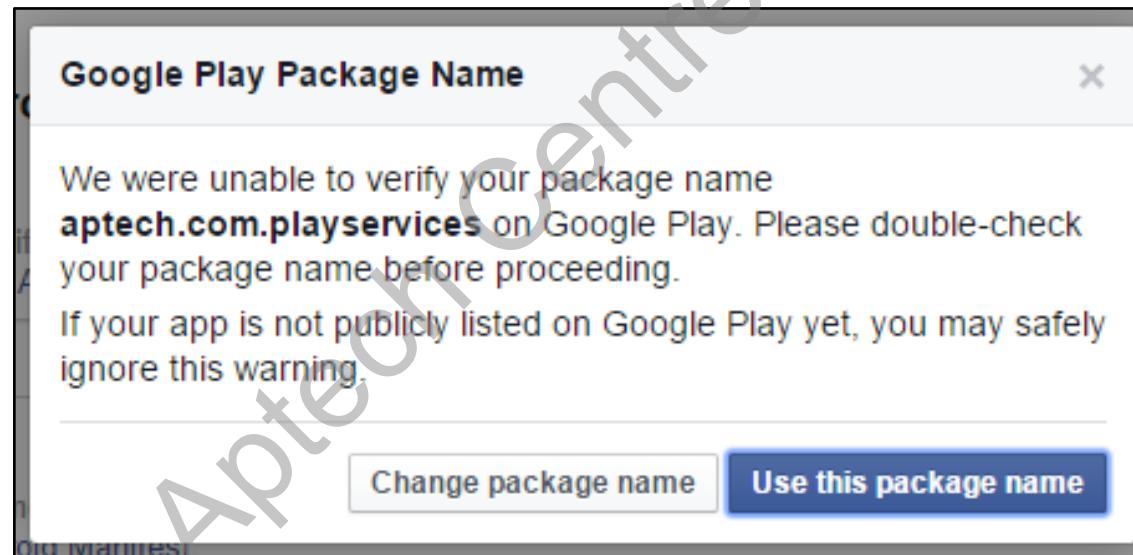
# Generating an API Key 4-8

- Download the SDK (If using the Eclipse IDE) and the APK file (If the Facebook App is not installed on your device) and navigate to the bottom of the page
- Enter the app details as shown in the following figure:



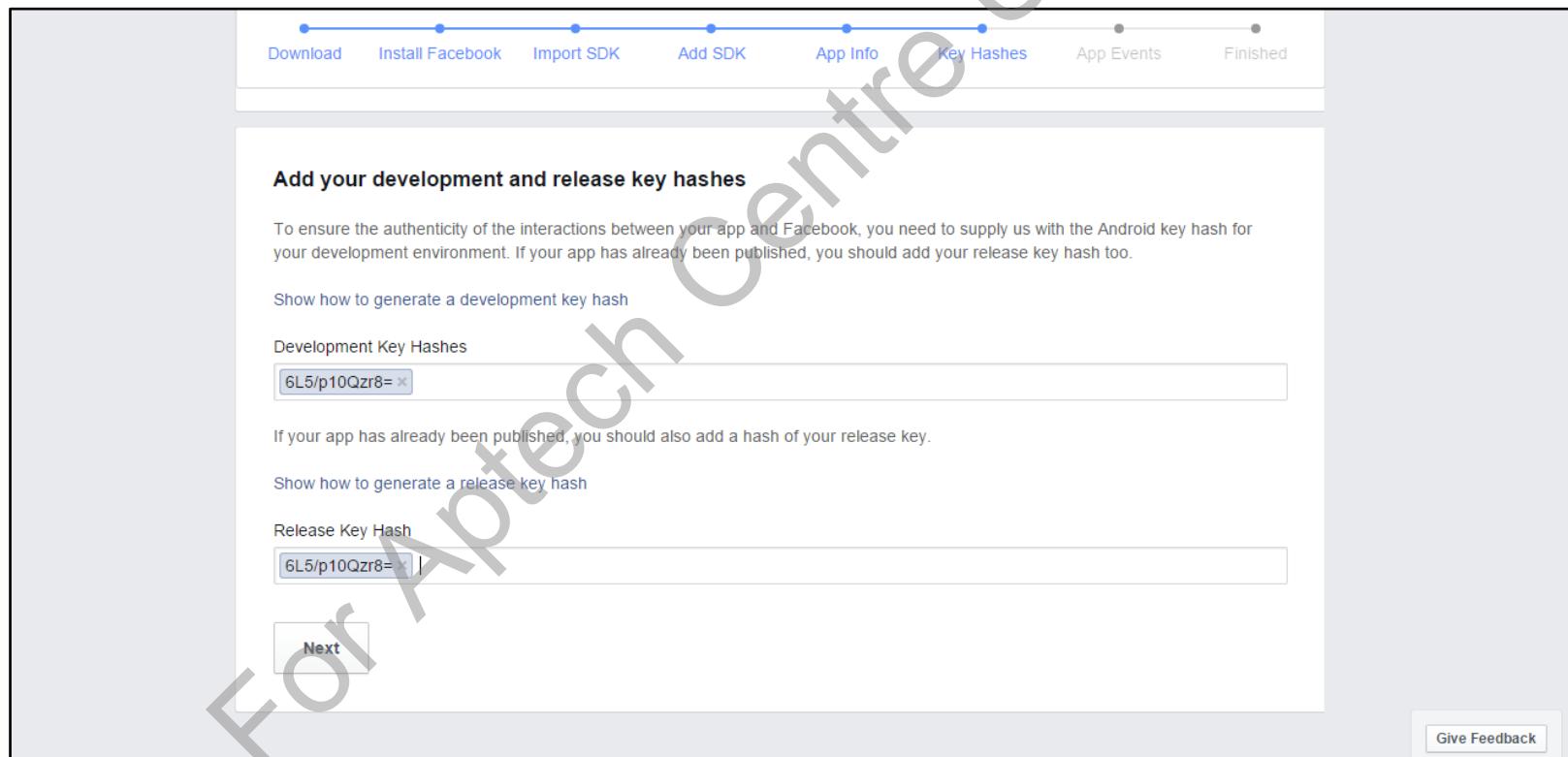
## Generating an API Key 5-8

- Click Next. If a warning pops up, click Use this package name as shown in the following figure:



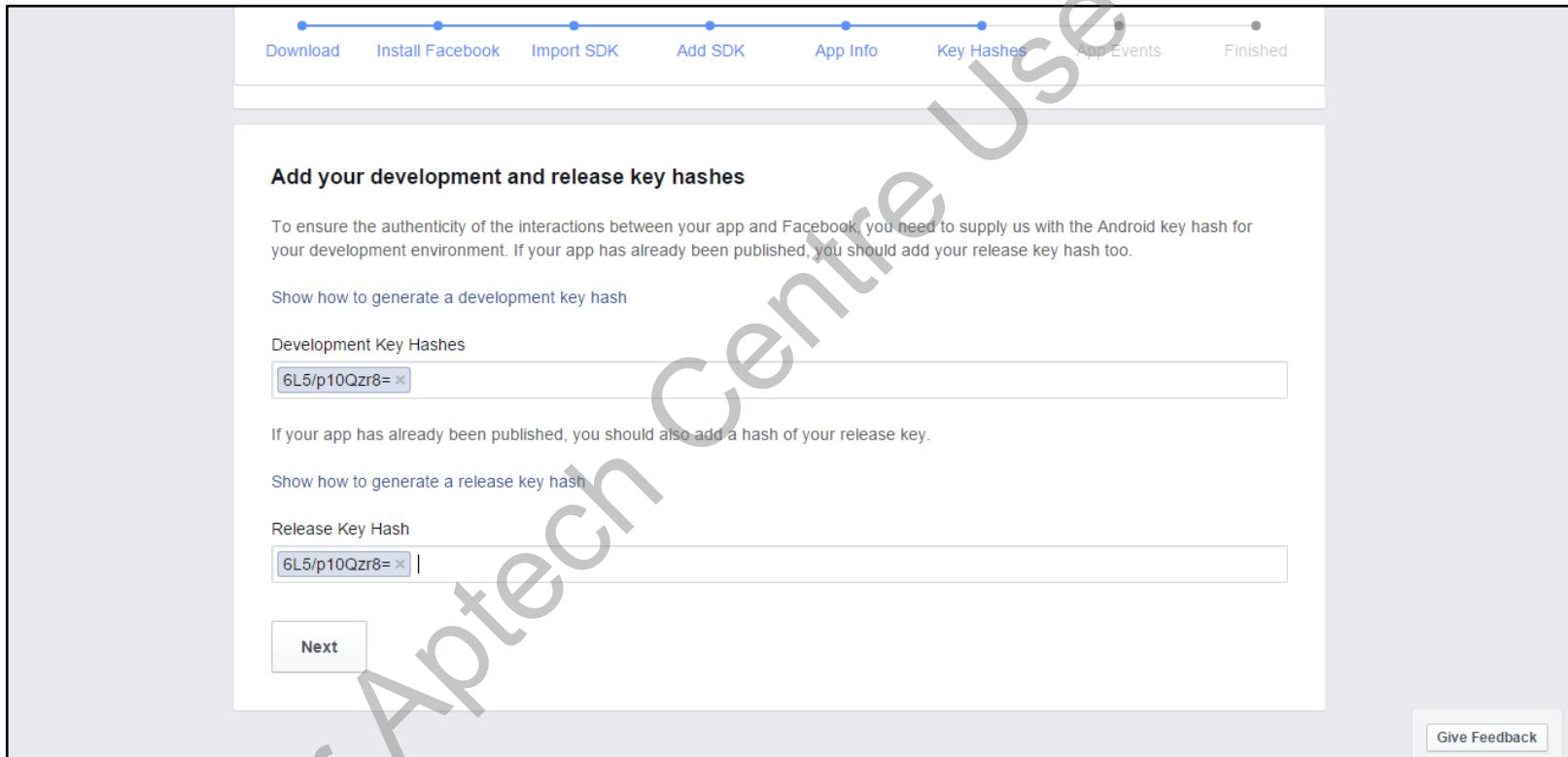
# Generating an API Key 6-8

- Enter the hash key generated earlier as shown in the following figure:



# Generating an API Key 7-8

- Enter the hash key generated earlier as shown in the following figure:



- Click Next
- Scroll down to the bottom of the page to see the finished image

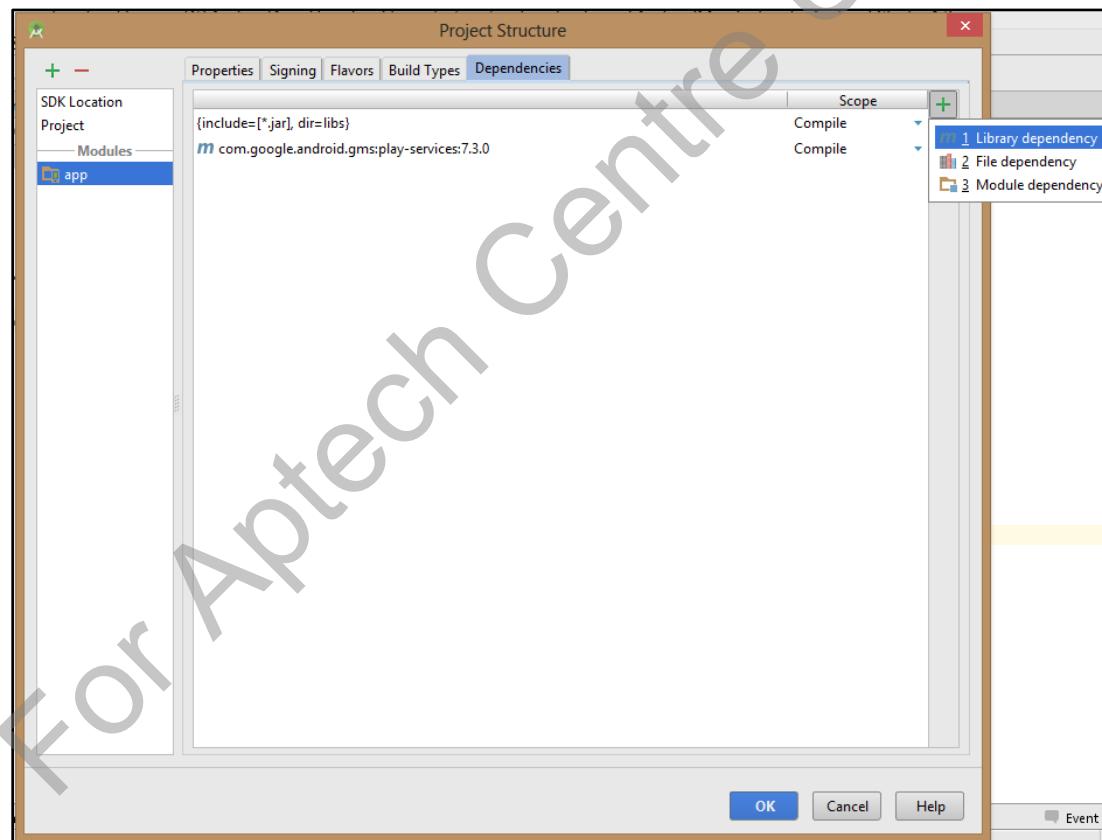
# Generating an API Key 8-8

- Navigate to My Apps menu and select the created application Aptech Auth
- Copy the App ID displayed as shown in the following figure:

The screenshot shows the Facebook Developers My Apps dashboard for the 'Aptech Auth' application. The left sidebar lists several options: Dashboard (which is selected and highlighted in blue), Settings, Status & Review, App Details, Roles, and Open Graph. The main content area is titled 'Dashboard' and features a large icon of an atom or gear. To its right, the app name 'Aptech Auth' is displayed with a green 'o' icon indicating it's in development mode. Below the name, the 'App ID' is listed as 935989213130894, the 'API Version' is set to v2.3, and the 'App Secret' is shown as a series of asterisks. A 'Show' button is located next to the secret. At the bottom of the dashboard, there's a 'Getting Started' section with a button labeled 'Getting Started'.

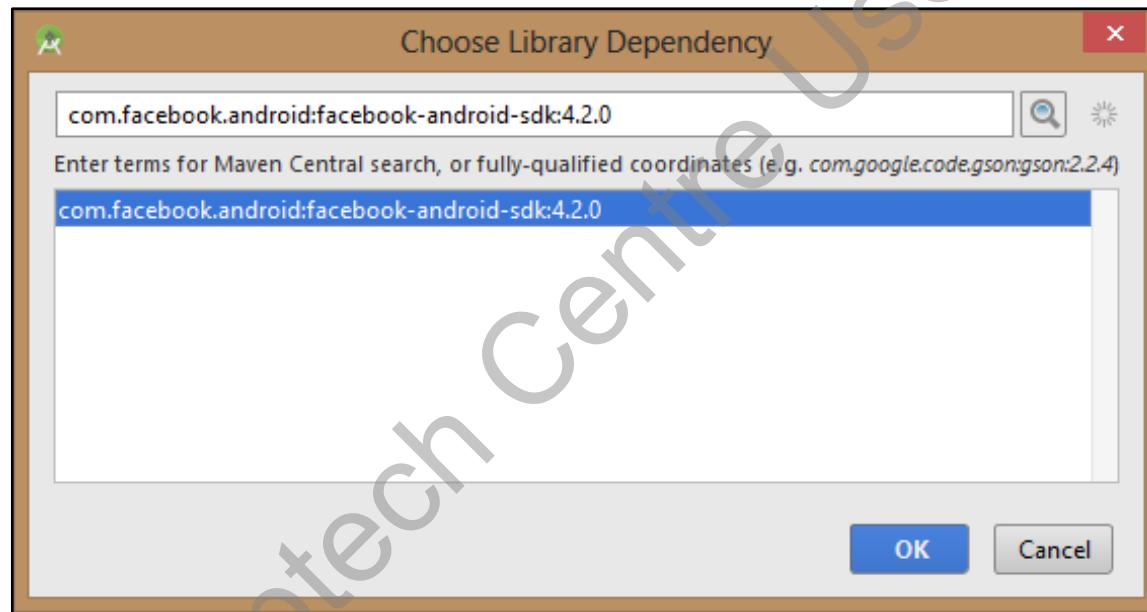
# Setting up Android Studio 1-2

- Open the project PlayServices
- Right-click app and select Open Module Settings
- Navigate to the Dependencies tab and click the ‘+’ sign
- Select Library dependency as shown in the following figure:



## Setting up Android Studio 2-2

- In the popup box, enter the name com.facebook.android:facebook-android-sdk: and select the Facebook library as shown in the following figure:



- Click OK
- Click OK again in the Dependencies tab

## Setting up the Target Device

- ◆ Navigate to the Play Store and search for the Facebook application and click Install
- ◆ The app will be downloaded and installed on the device
- ◆ Download the Facebook application from the Developer Page as described in the previous section
- ◆ Connect the target device using USB with adb
- ◆ Execute the command as shown in the following Code Snippet:

```
adb install C:\Path_here\Facebook-{version_here}.apk
```

# Facebook Login Button Example

- The code to add a Facebook Sign in Button is shown in the following Code Snippet:

```
<LinearLayout>

    <com.facebook.login.widget.LoginButton
        android:id="@+id/fbLoginButton"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_weight="0.11"
        android:minHeight="50dp"
        android:padding="10dp" />

</LinearLayout>
```

# Requesting User Information Example

- The code to request Logged in user information is shown in the following Code Snippet:

```
FacebookSdk.sdkInitialize(getApplicationContext());
        callbackManager = CallbackManager.Factory.create();
        fbPermissions = new ArrayList<String>();
        fbPermissions.add("public_profile");
        fbPermissions.add("email");
        fbPermissions.add("user_friends");
        fbLoginButton.setReadPermissions(fbPermissions);
        fbLoginButton.registerCallback(callbackManager, new
FacebookCallback<LoginResult>() {
    @Override
    public void onSuccess(LoginResult loginResult) accessToken =
loginResult.getAccessToken();
        loadFbInfo();
    }
}
```

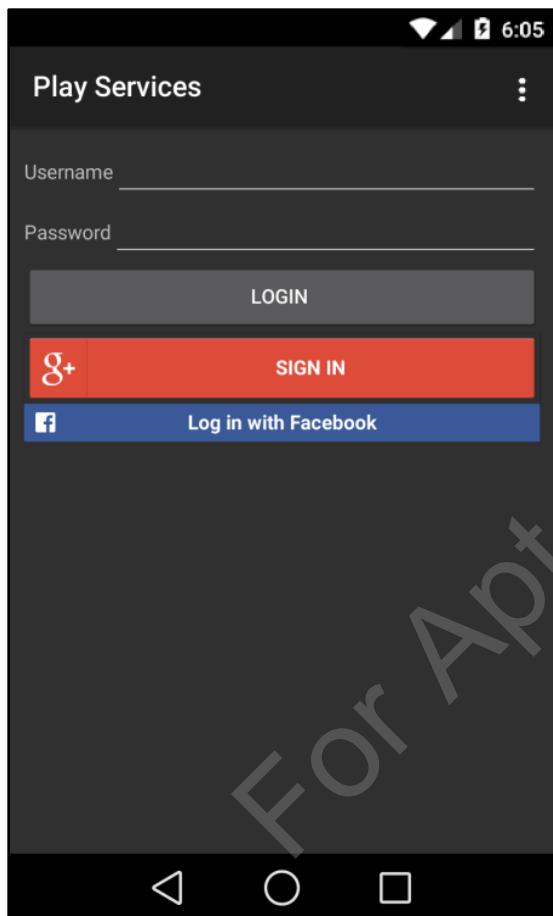
## Retrieving User Information Example

- The code to retrieve Logged in user information is shown in the following Code Snippet:

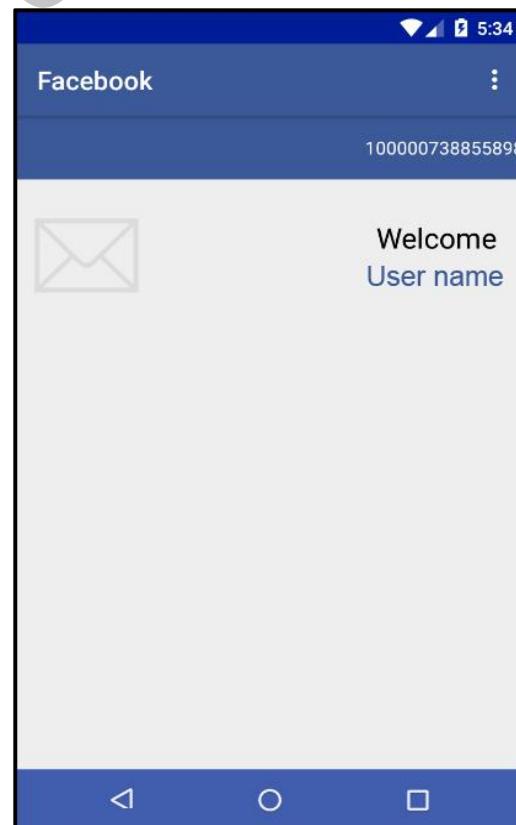
```
Profile userProfile = Profile.getCurrentProfile();  
  
String userId = userProfile.getId();  
String userRealName = userProfile.getName();  
String userPhotoUrl =  
userProfile.getProfilePictureUri(100,100).toString();
```

# Third Party API Application 1-2

- Using the code, an application for demonstrating third Party API is created as shown in the following figure:

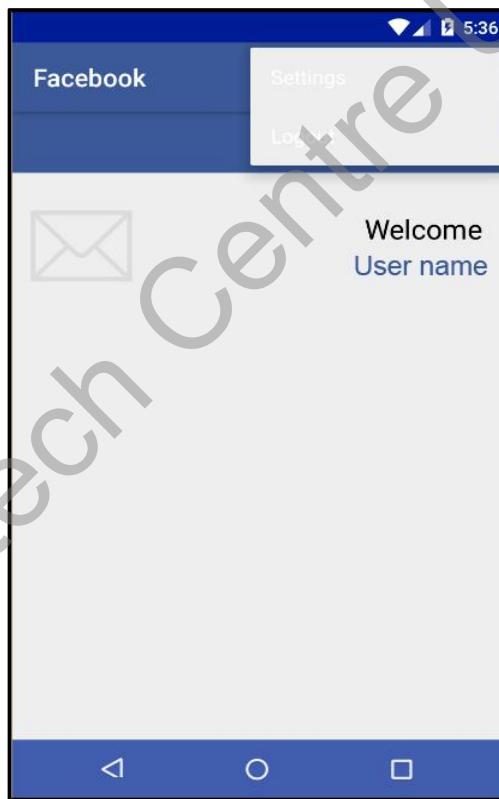


- Click the Log in with Facebook button and login to your Facebook account
- Once logged in, the user is redirected to the Facebook login homepage as shown in the following figure:



## Third Party API Application 2-2

- The users can logout from the action menu and return to the login page as shown in the following figure:



## Third Party API Application Logic

- ◆ A Facebook Sign in Button is added for the Main Activity
- ◆ An onSuccess() listener is registered for the Login Button
- ◆ Once the login process is completed, the login listener is triggered
- ◆ The user details are then retrieved from the Login Listener
- ◆ The user details are added as extras and the login Home Page is loaded



# Creating Web Services

- ◆ Before we create an application using the Web service, the Web service itself needs to be created
- ◆ For demonstration purposes, a simple PHP Web service is created running on an Apache Server
- ◆ The PHP code is shown in the following Code Snippet:

```
<?php  
  
$username = $_GET[ 'uname' ];  
$password = $_GET[ 'pass' ];  
  
if($username == "admin" && $password == "adminpass")  
{ echo "success"; }  
else  
{ echo "failure"; }  
  
?>
```

# Authenticating with the Web Services

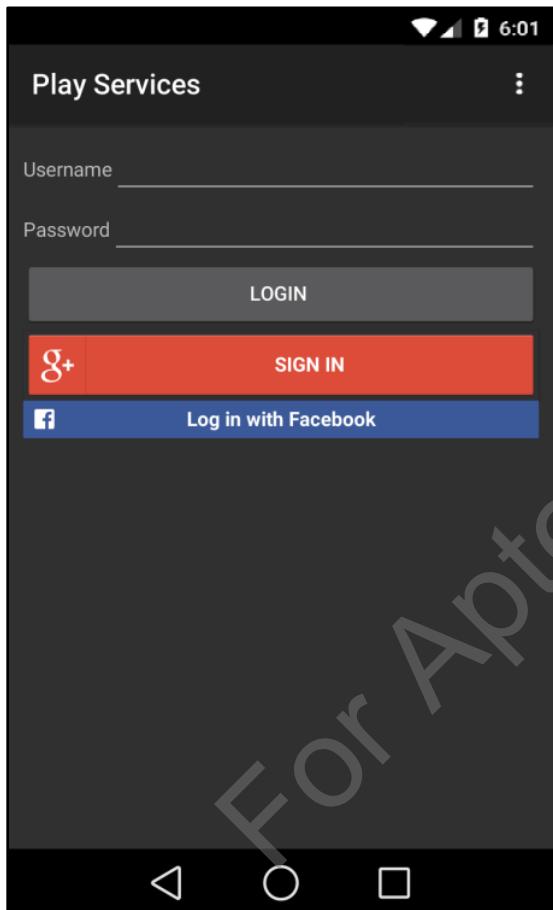
- The code to authenticate with the created Web Service is shown in the following Code Snippet:

```
Uri.Builder builder = new Uri.Builder();
    builder.scheme("http")
        .authority("192.168.1.19")
        .appendPath("webservice")
        .appendPath("index.php")
        .appendQueryParameter("uname", userName)
        .appendQueryParameter("pass", password);
Uri webService = builder.build();

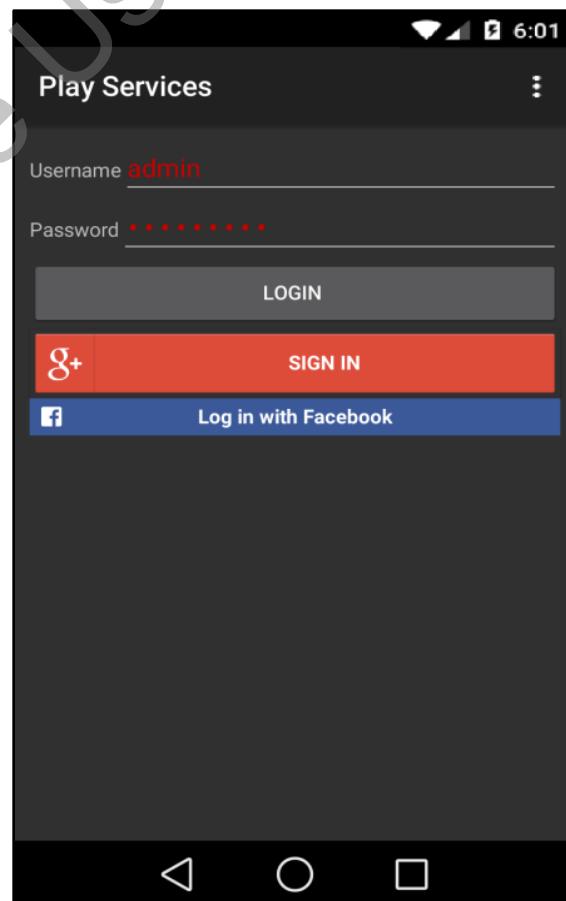
try {
    HttpClient = new DefaultHttpClient();
    HttpResponse response = httpclient.execute(new HttpGet(webService.toString()));
    checkResponse(stringBuffer.toString());
}
catch (IOException e) {
    e.printStackTrace();
}
```

## Custom Web Service Example 1-2

- Using the code, an application for demonstrating Custom Web Services is created as shown in the following figure:

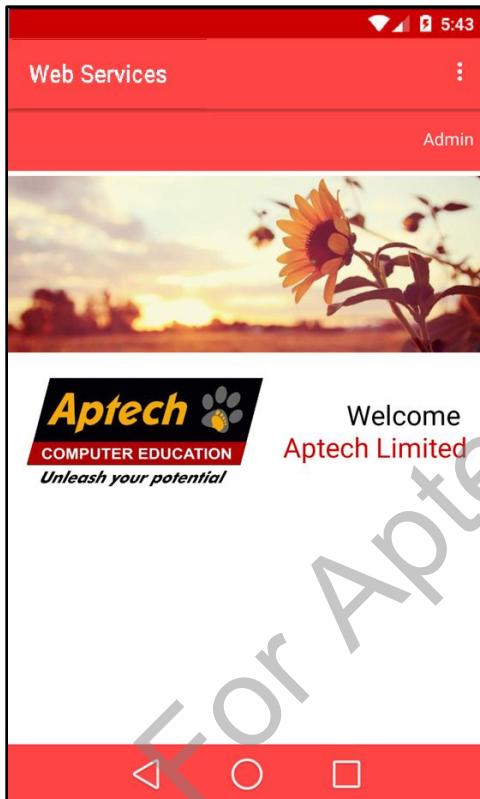


- Enter the Username as admin and Password as adminpass and click LOGIN as shown in the following figure:

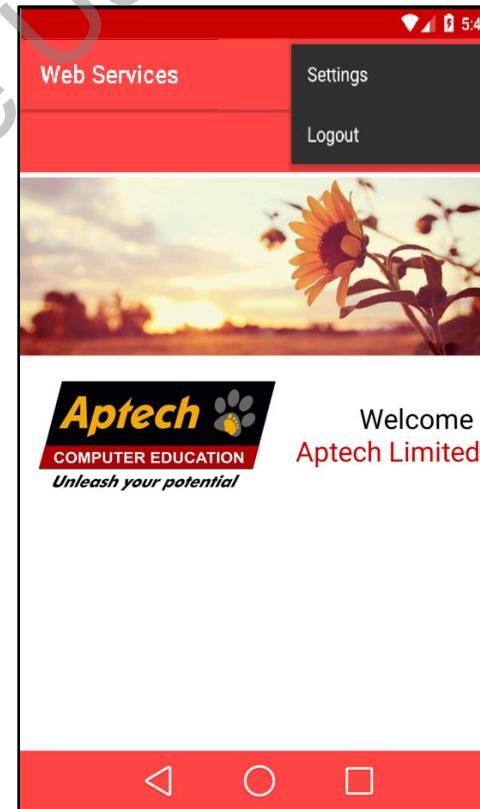


## Custom Web Service Example 2-2

- Once logged in, the user is redirected to the Web Service home screen as shown in the following figure:



- Users can logout from the Action menu to return to the home screen as shown in the following figure:



# Custom Web Service Application Logic

- ◆ A Custom Web Service is created for the application
- ◆ Once the user enters the login details and clicks the Login button, the details are retrieved and passed to the service using a get request
- ◆ The Service checks the username and password and if they match, returns “success”. Else it returns “failure”
- ◆ The application checks the server response and loads the activity if log in was successful



# Web Service Implementation

- ◆ There are service providers for almost all kinds of applications. There are also storage providers such as Google Drive, Computation providers such as Amazon EC2, and so on
- ◆ The first step is to decide the communication standard that will be used for the Web Service
- ◆ ReSTful is an architectural style whereas SOAP is a protocol implementation
- ◆ All the requirements need to be communicated beforehand
- ◆ Once the development is complete, the Android Developer is provided with the API and the associated documentation of the Web Service

# When to Not Implement Web Services?

- ◆ If the solution can be implemented locally on the device
- ◆ If battery usage is a concern
- ◆ If application is expected to respond quickly
- ◆ If data access is required
- ◆ When Sensitive data is involved
- ◆ If the number of requests is expected to be low
- ◆ If the application is expected to function offline



# When to Implement Web Services?

- ◆ If a platform is already established
- ◆ If cross-platform services need to be provided
- ◆ When scaling is a concern. Web Services can scale up very easily
- ◆ Data Encapsulation is a concern. The mode and structure of storage of information is not revealed to the user
- ◆ When fault tolerance is desired. Web Services can quickly recover from major errors/crashes without revealing it to the user
- ◆ If services are planned to be changed/tuned. Web Services provide the option to completely rewrite the implementation without affecting the service
- ◆ If a task is intensive and offloading it can save energy and time on the device



# Text-to-Speech and Voice Recognition

- ◆ Text-to-Speech is process of narrating text to the user in audio form
- ◆ Voice Recognition is process of taking the user's audio input and generating text
- ◆ Android readily provides API for both these operations



## ◆ Voice Recognition

- ❖ **RecognizerIntent:** A broadcast intent that starts an activity for accepting user voice input. The extras for this Intent can be used for setting additional settings

## ◆ Text-to-Speech

### ❖ **TextToSpeech:**

- ❖ The central class used in text to speech systems
- ❖ It is responsible for generating speech from input text
- ❖ The setPitch() method is used to set the pitch of the voice and the setSpeechRate() method is used to set the speed of the voice
- ❖ The setVoice() method can be used to set the voice of the speech if multiple voice packs are available
- ❖ The synthesizeToFile() method can be used to save the speech output to a file

- ◆ **TextToSpeech.OnInitListener:**

- ◆ The event listener class for TextToSpeech
- ◆ The onInit() method is called once TextToSpeech has been initialized
- ◆ The method can be implemented to set the base settings for TTS

- ◆ **Voice:**

- ◆ The class representing a voice pack installed on the device
- ◆ The name of the voice can be retrieved by using getName() method
- ◆ The getQuality(), getLatency(), and getLocale() methods are used to retrieve the quality, latency, and locale information respectively

# Voice Recognition Example

- The code to receive voice input from the user is shown in the following Code Snippet:

```
Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);

intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);

        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, "en-US");
        intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Speak a few words for
TTS");
        try {
            startActivityForResult(intent,1);
        } catch (ActivityNotFoundException a) {
            Toast.makeText(getApplicationContext(),"Voice Recognition not
supported on this Device / Currently Unavailable",Toast.LENGTH_SHORT).show();
        }
    }

...
ArrayList<String> result =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
```

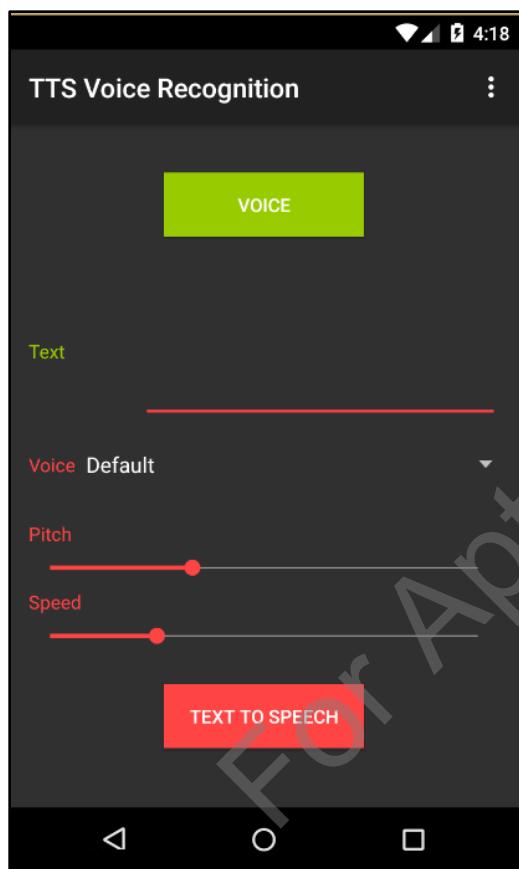
# Text to Speech Example

- The code convert text to speech is shown in the following Code Snippet:

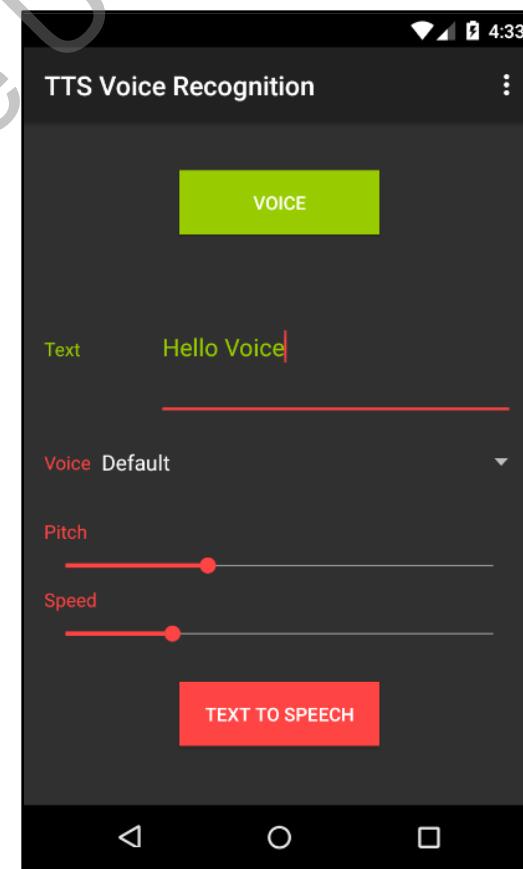
```
TextToSpeech textToSpeech = new TextToSpeech(this, new
TextToSpeech.OnInitListener() {
    @Override
    public void onInit(int status) {
        textToSpeech.setLanguage(Locale.US);
        List<String> voiceList = new ArrayList<String>();
        try {
            Set<Voice> voices = textToSpeech.getVoices();
            if (voices.size() > 0) {
                Iterator<Voice> iterator = voices.iterator();
                while (iterator.hasNext()) {
                    voiceList.add(iterator.next().getName());
                }
            }
        } catch(NullPointerException e) {
            voiceList.add("Default");
        }
    }
});
```

# TTS and Voice Recognition Application

- Using the explained code, an application for demonstrating TTS and Voice Recognition is created as shown in the following figure:



- Click the voice button and speak few words. Once completed, the voice is converted into text and the output is displayed in the text box as shown in the following figure:



## TTS and Voice Recognition Application Logic

- ◆ The Main Activity consist of a button to receive Voice Input, and another button to Convert the text to Voice
- ◆ An EditText is provided to display the input text and to allow the user to enter his own input text manually
- ◆ Controls are provided for Voice Settings
- ◆ Once the Text to Speech button is clicked, the text is converted to voice with the settings selected by the user
- ◆ The TextToSpeech's speak() method is used for this purpose



## Summary

- ◆ Web Services are application components that allow communicating devices to use features that are not available on the device locally
- ◆ An application can use both Google API as well as third party Web Services
- ◆ ReSTful and SOAP are well established communication standards for Web Services
- ◆ Unnecessary use of Web Services can hurt the application performance and usability
- ◆ Text-to-Speech (TTS) is the process of narration of text to the user. Android has built-in support for TTS which can be accessed via the TextToSpeech class
- ◆ Voice Recognition is the converse of TTS. It is the process of converting user audio input into text