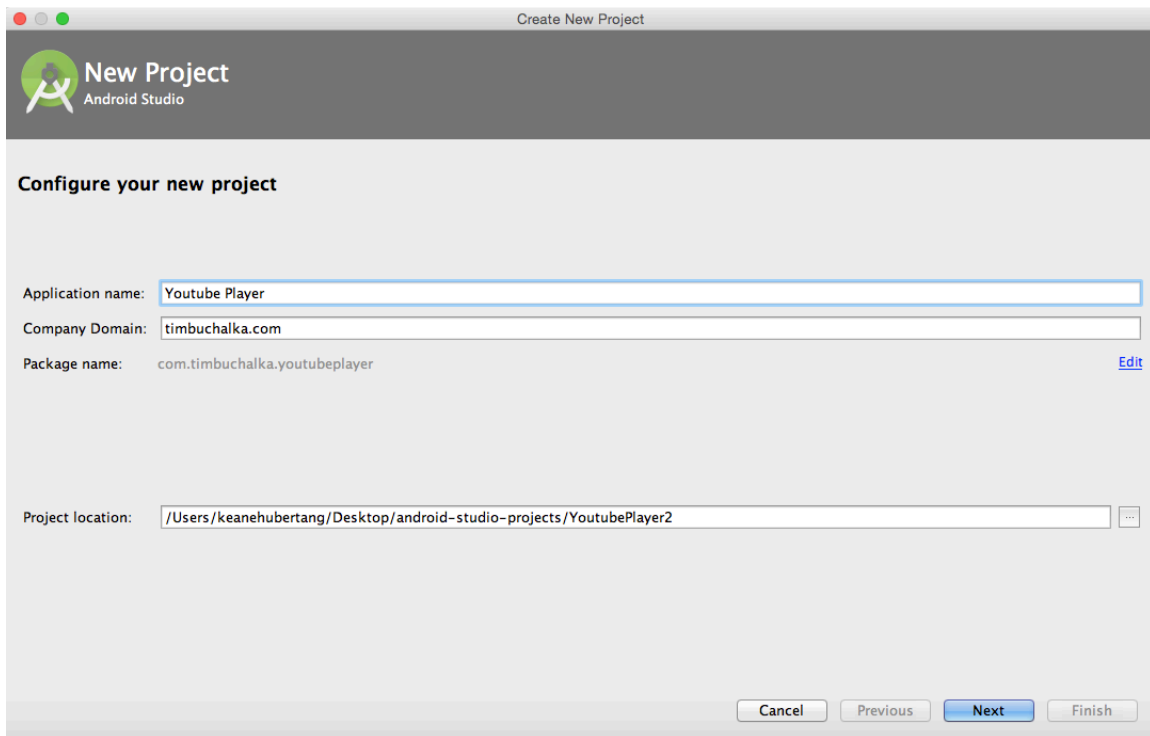# Youtube App For Android Studio 1.4

Lecture 42 – Youtube App Setup

Create a new Android Studio Project, set the Application Name to "Youtube Player" and set the Company Domain to "timbuchalka.com". Click On Next to proceed.

Make sure Phone and Tablet SDK is checked, and is set to API 16. Leave the other boxes uncheck and press Next.

Choose Blank Activity and press Next and on the last screen leave everything at default and press Finish.

Then go to https://developers.google.com/youtube/android/player and download the Youtube Player jar file. Once it has been downloaded, extract the zip file and copy and paste the .jar file into the YoutubePlayer app's libs folder in **YouTubePlayer/app/libs**.
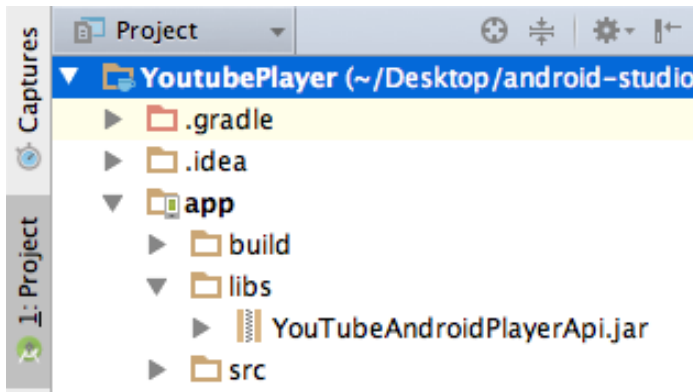
Then open up the **build.gradle** file in the **YouTubePlayer/app**. And add the following line "*compile files('libs/YoutubeAndroidPlayerApi.jar')*" in the dependencies area.

http://i.imgur.com/scqzLUr.png
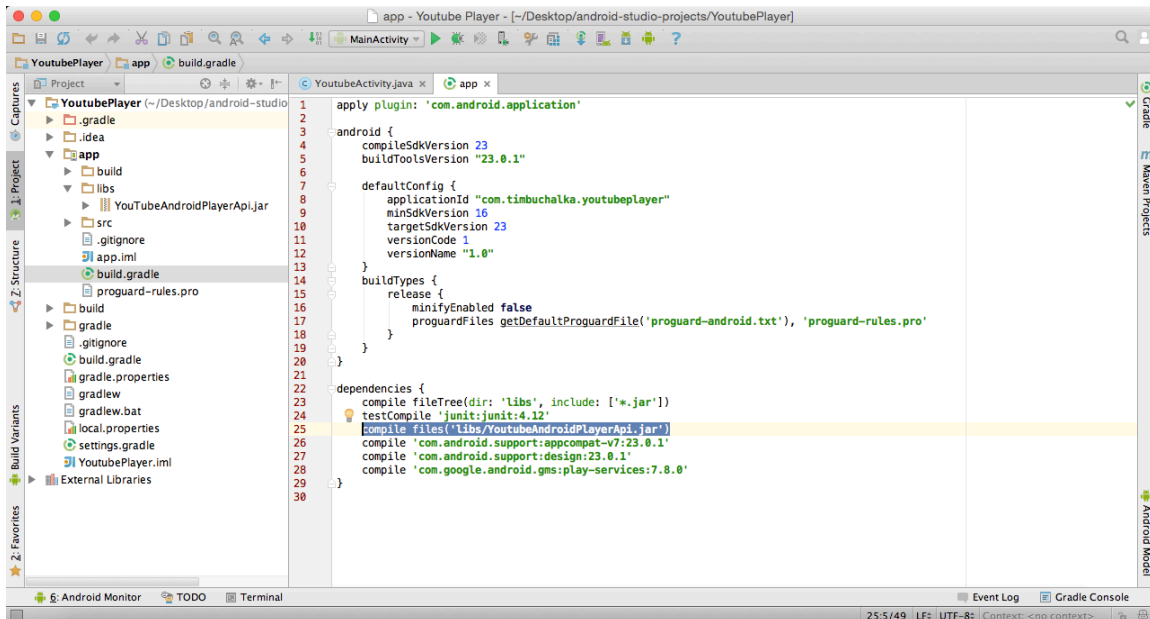
Next open up the MainActivity.java file in the **YouTubePlayer/app/src/main/java/packagename** and comment out the following lines by pressing CMD+/ or manually commenting it by adding // on the beginning of each line:

FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {

```
        @Override
        public void onClick(View view) {
            Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
        }
    });
```

This what it looks like after commenting the lines.

```java
public class MainActivity extends AppCompatActivity{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

//        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
//        fab.setOnClickListener(new View.OnClickListener() {
//            @Override
//            public void onClick(View view) {
//                Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
//                        .setAction("Action", null).show();
//            }
//        });
    }
```

Next, open up the activity_main.xml file in the
YouTubePlayer/app/src/main/res/layout and comment out the following lines:

```xml
    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="@dimen/fab_margin"
        android:src="@android:drawable/ic_dialog_email"/>
```

This is what it looks like after commenting the lines:

```
    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_main"/>

    <!--<android.support.design.widget.FloatingActionButton-->
        <!--android:id="@+id/fab"-->
        <!--android:layout_width="wrap_content"-->
        <!--android:layout_height="wrap_content"-->
        <!--android:layout_gravity="bottom|end"-->
        <!--android:layout_margin="@dimen/fab_margin"-->
        <!--android:src="@android:drawable/ic_dialog_email"/>-->

</android.support.design.widget.CoordinatorLayout>
```

Next, right click on the **YouTubePlayer/app/src/main/java/packagename** and create a new Blank Activity. Then set the Activity Name to **YoutubeActivity**, set the Layout Name to activity_youtube, and then clicked Finish.
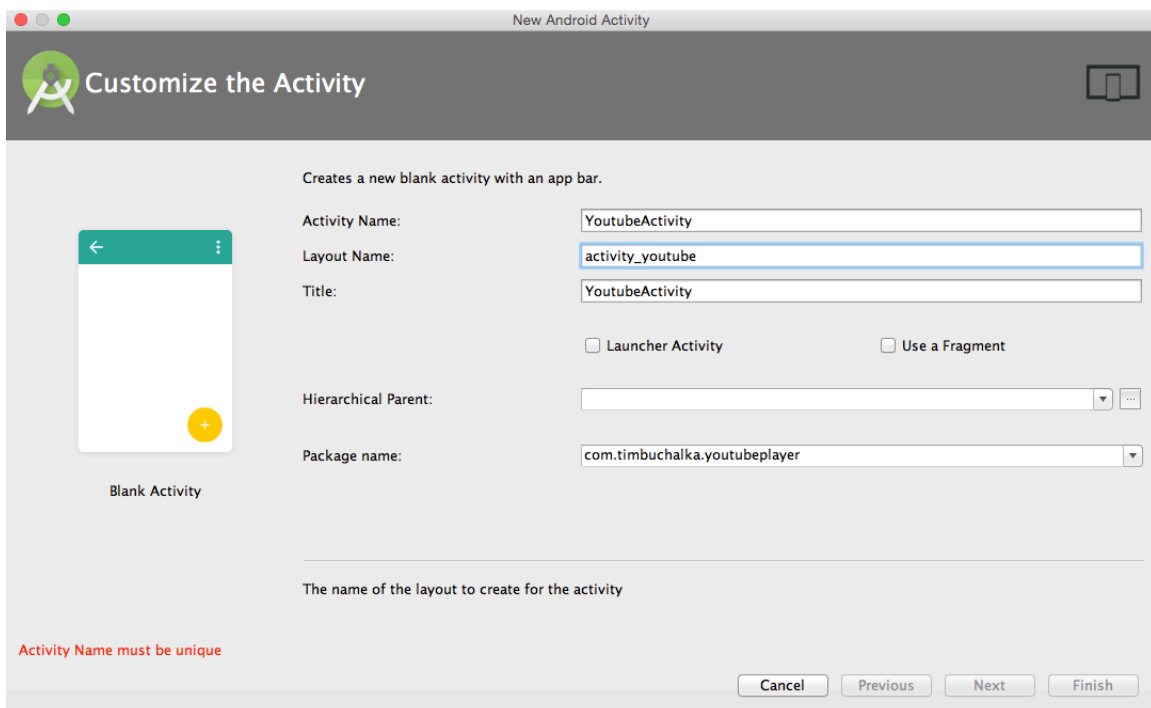
After the activity has been created open up YoutubeActivity in the **YouTubePlayer/app/src/main/java/packagename** and comment out the following lines by pressing CMD+/ or manually commenting it by adding // on the beginning of each line:

```
FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
```

```
        public void onClick(View view) {
            Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                .setAction("Action", null).show();
        }
    });
```

This what it looks like after commenting the lines.

```
//          FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
//          fab.setOnClickListener(new View.OnClickListener() {
//              @Override
//              public void onClick(View view) {
//                  Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
//                      .setAction("Action", null).show();
//              }
//          });
```

Next, comment out the following lines in the onCreate method.

```
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
```

```
//          Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
//          setSupportActionBar(toolbar);
```

Next, open up the activity_youtube.xml file in the
YouTubePlayer/app/src/main/res/layout and comment out the following lines:

```
    <android.support.design.widget.FloatingActionButton
      android:id="@+id/fab"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="bottom|end"
      android:layout_margin="@dimen/fab_margin"
      android:src="@android:drawable/ic_dialog_email"/>
```

This is what it looks like after commenting the lines:

```
    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_main"/>

    <!--<android.support.design.widget.FloatingActionButton-->
        <!--android:id="@+id/fab"-->
        <!--android:layout_width="wrap_content"-->
        <!--android:layout_height="wrap_content"-->
        <!--android:layout_gravity="bottom|end"-->
        <!--android:layout_margin="@dimen/fab_margin"-->
        <!--android:src="@android:drawable/ic_dialog_email"/>-->

</android.support.design.widget.CoordinatorLayout>
```

Then go back to YoutubeActivity.java and replace the following line "*public class YoutubeActivity extends AppCompatActivity*" to "*public class YoutubeActivity extends YouTubeBaseActivity implements YouTubePlayer.OnInitializedListener*"

Then add the following fields within the class:

private String GOOGLE_API_KEY = "TBA";
private String YOUTUBE_VIDEO_ID = "TBA";

Your current YoutubeActivity code should look like the following:

public class YoutubeActivity extends YouTubeBaseActivity implements YouTubePlayer.OnInitializedListener{

private String GOOGLE_API_KEY = "AIzaSyB4xh2zIJ5nUfytHeCb6huCQk-S_jvAHyo";
private String YOUTUBE_VIDEO_ID = "XUb97Wv9R6o";

@Override
protected void onCreate(Bundle savedInstanceState) {
*   super.onCreate(savedInstanceState);*
*   setContentView(R.layout.activity_youtube);*
//   Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
//   setSupportActionBar(toolbar);

//   FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
//   fab.setOnClickListener(new View.OnClickListener() {
//   @Override
//   public void onClick(View view) {
//   Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
//   .setAction("Action", null).show();

```
//      }
//    });
  }
}
```

Next open up the content_youtube.xml and replace the contents with the following code below:

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/activity_youtube"
    tools:context="com.timbuchalka.youtubeplayer.YoutubeActivity">

    <com.google.android.youtube.player.YouTubePlayerView
        android:id="@+id/youtube_player"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#ffffff"/>

</LinearLayout>
```

Lecture 43 – Youtube App - Add activity & Listeners

Open up YoutubeActivity.java and add the following lines within the onCreate method.

> *YouTubePlayerView youTubePlayerView = (YouTubePlayerView) findViewById(R.id.youtube_player);*
> *youTubePlayerView.initialize(GOOGLE_API_KEY, this);*

```
        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_youtube);
            YouTubePlayerView youTubePlayerView = (YouTubePlayerView) findViewById(R.id.youtube_player);
            youTubePlayerView.initialize(GOOGLE_API_KEY, this);
//          Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
//          setSupportActionBar(toolbar);

//          FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
//          fab.setOnClickListener(new View.OnClickListener() {
//              @Override
//              public void onClick(View view) {
//                  Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
//                          .setAction("Action", null).show();
//              }
//          });
        }
```

Then generate Override Methods and Select **onInitializationSuccess** and **onInitializationFailure** which can be found way below the Override Methods window. Press Ok after selecting both.

Within the parentheses of the public void onInitializationSuccess method, change **boolean b** to **boolean wasRestored** .

Add the following line within the **onInitializationFailure** method.

Toast.makeText(this, "Cannot initialize Youtube player", Toast.LENGTH_LONG).show();

Add the following lines of code in the **onInitializationSuccess** method.

> *youTubePlayer.setPlayerStateChangeListener(playerStateChangeListener);*
> *youTubePlayer.setPlaybackEventListener(playbackEventListener);*
>
> *if(!wasRestored) {*
> * youTubePlayer.cueVideo(YOUTUBE_VIDEO_ID);*
> *}*

It should look like the following:

```
    @Override
    public void onInitializationSuccess(YouTubePlayer.Provider provider, YouTubePlayer youTubePlayer, boolean wasRestored) {
        youTubePlayer.setPlayerStateChangeListener(playerStateChangeListener);
        youTubePlayer.setPlaybackEventListener(playbackEventListener);

        if(!wasRestored) {
            youTubePlayer.cueVideo(YOUTUBE_VIDEO_ID);
        }
    }

    @Override
    public void onInitializationFailure(YouTubePlayer.Provider provider, YouTubeInitializationResult youTubeInitializationResu
        Toast.makeText(this, "Cannot initialize Youtube player", Toast.LENGTH_LONG).show();
    }
}
```

http://i.imgur.com/xgHwbni.png

Then next we need add the **PlaybackEventListener** and
PlayerStateChangeListener , so add the following line of code in YoutubeActivity.

```
    private YouTubePlayer.PlaybackEventListener playbackEventListener = new
YouTubePlayer.PlaybackEventListener() {
        @Override
        public void onPlaying() {

        }

        @Override
        public void onPaused() {

        }

        @Override
        public void onStopped() {

        }

        @Override
        public void onBuffering(boolean b) {

        }

        @Override
        public void onSeekTo(int i) {

        }
    };

    private YouTubePlayer.PlayerStateChangeListener playerStateChangeListener =
new YouTubePlayer.PlayerStateChangeListener() {
        @Override
```

```
    public void onLoading() {

    }

    @Override
    public void onLoaded(String s) {

    }

    @Override
    public void onAdStarted() {

    }

    @Override
    public void onVideoStarted() {

    }

    @Override
    public void onVideoEnded() {

    }

    @Override
    public void onError(YouTubePlayer.ErrorReason errorReason) {

    }
};
```

Then add the line "*Toast.makeText(YoutubeActivity.this, "Good, video is playing...",
Toast.LENGTH_LONG).show();*" in the **public void onPlaying** method. Next, add the
line "*Toast.makeText(YoutubeActivity.this, "Oh you paused???",
Toast.LENGTH_LONG).show();*" in the **public void onPaused()** method. These are in
the **private YouTubePlayer.PlaybackEventListener**

The **private YouTubePlayer.PlaybackEventListener** should look like this in the
end.

```
private YouTubePlayer.PlaybackEventListener playbackEventListener = new YouTubePlayer.PlaybackEventListener() {
    @Override
    public void onPlaying() {
        Toast.makeText(YoutubeActivity.this, "Good, video is playing...", Toast.LENGTH_LONG).show();

    }

    @Override
    public void onPaused() {
        Toast.makeText(YoutubeActivity.this, "Oh you paused???", Toast.LENGTH_LONG).show();

    }

    @Override
    public void onStopped() {

    }

    @Override
    public void onBuffering(boolean b) {

    }

    @Override
    public void onSeekTo(int i) {

    }
};
```

http://i.imgur.com/yAnoP0K.png

Then add the line "*Toast.makeText(YoutubeActivity.this, "Make the video content author rich???", Toast.LENGTH_LONG).show();*" in the **public void onAdStarted()** method. Next, add the line "*Toast.makeText(YoutubeActivity.this, "Video started!!", Toast.LENGTH_LONG).show();*" in the **public void onVideoStarted()** method. These are in the **private YouTubePlayer.PlayerStateChangeListener**

The **private YouTubePlayer.PlayerStateChangeListener** should look like this in the end.

```
private YouTubePlayer.PlayerStateChangeListener playerStateChangeListener = new YouTubePlayer.PlayerStateChangeListener()
    @Override
    public void onLoading() {

    }

    @Override
    public void onLoaded(String s) {

    }

    @Override
    public void onAdStarted() {
        Toast.makeText(YoutubeActivity.this, "Make the video content author rich???", Toast.LENGTH_LONG).show();

    }

    @Override
    public void onVideoStarted() {
        Toast.makeText(YoutubeActivity.this, "Video started!!", Toast.LENGTH_LONG).show();

    }

    @Override
    public void onVideoEnded() {

    }

    @Override
    public void onError(YouTubePlayer.ErrorReason errorReason) {

    }
};
```

http://i.imgur.com/I5h4NMD.png

Lecture 44 – Youtube App – Getting The API KEY

Create a Youtube API and paste the API key in the *"private String GOOGLE_API_KEY = "TBA";"* replace the TBA with the API key. Then get the a Youtube Video ID from an random youtube video and replace the TBA within the *"private String YOUTUBE_VIDEO_ID = "TBA";"* line. These lines are in the YoutubeActivity.

They should look like this when adding the key and video id.

```
private String GOOGLE_API_KEY = "AIzaSyB4xh2zIJ5nUfytHeCb6huCQk-S_jvAHyo";
private String YOUTUBE_VIDEO_ID = "XUb97Wv9R6o";
```

Next, open up AndroidManifest.xml file in the **YouTubePlayer/app/src/main/res** and paste in the following lines of code within the <activity> tag for YoutubeActivity.

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

It should look like this in the end.

```
<activity
    android:name=".YoutubeActivity"
    android:label="@string/title_activity_youtube"
    android:theme="@style/AppTheme.NoActionBar" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Then run the YoutubeActivity by right clicking on the YoutubeActivity on the project pane and select Run YoutubeActivity.

Lecture 45 – Youtube App – Standalone Activity

Right click on the **YouTubePlayer/app/src/main/java/packagename** and create a new Blank Activity. Then set the Activity Name to **StandaloneActivity**, set the Layout Name to activity_standalone, and then clicked Finish.
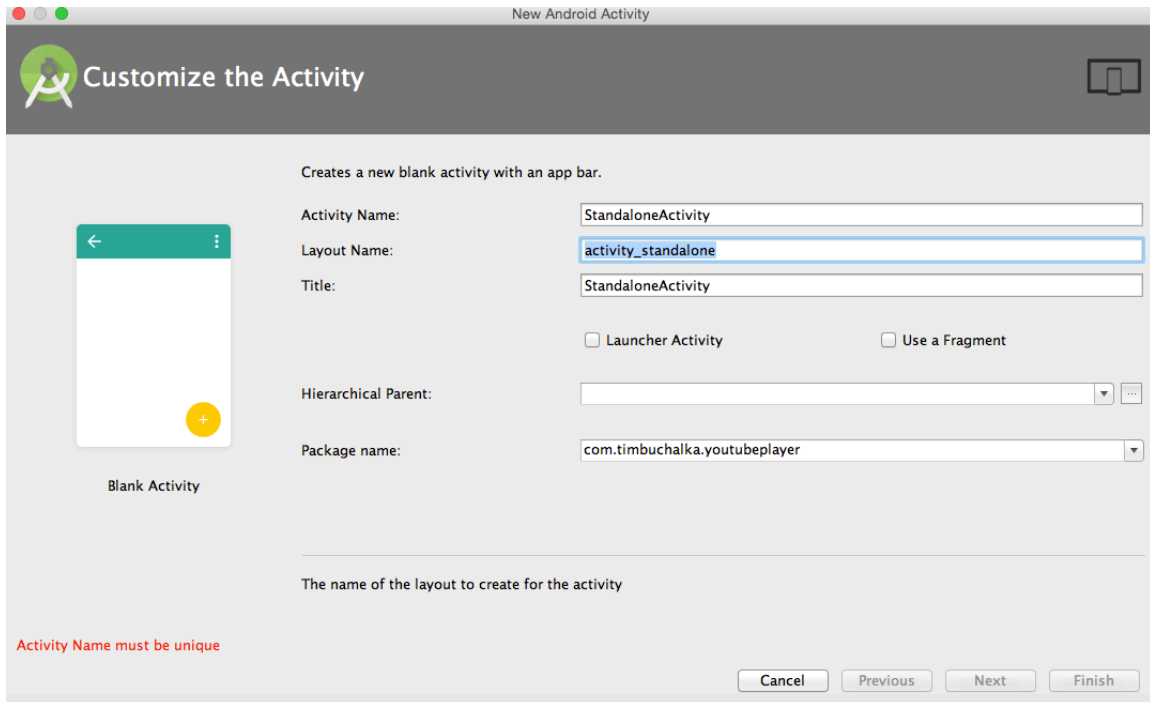
After the activity has been created open up StandaloneActivity in the **YouTubePlayer/app/src/main/java/packagename** and comment out the following lines by pressing CMD+/ or manually commenting it by adding // on the beginning of each line:

```
    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
    fab.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
        Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
            .setAction("Action", null).show();
      }
    });
```

This what it looks like after commenting the lines.

```
//        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
//        fab.setOnClickListener(new View.OnClickListener() {
//            @Override
//            public void onClick(View view) {
//                Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
//                        .setAction("Action", null).show();
//            }
//        });
```

Next, open up the activity_standalone.xml file in the
YouTubePlayer/app/src/main/res/layout and comment out the following lines:

<android.support.design.widget.FloatingActionButton
  android:id="@+id/fab"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_gravity="bottom|end"
  android:layout_margin="@dimen/fab_margin"
  android:src="@android:drawable/ic_dialog_email"/>

This is what it looks like after commenting the lines:

```
    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_main"/>

    <!--<android.support.design.widget.FloatingActionButton-->
        <!--android:id="@+id/fab"-->
        <!--android:layout_width="wrap_content"-->
        <!--android:layout_height="wrap_content"-->
        <!--android:layout_gravity="bottom|end"-->
        <!--android:layout_margin="@dimen/fab_margin"-->
        <!--android:src="@android:drawable/ic_dialog_email"/>-->

</android.support.design.widget.CoordinatorLayout>
```

Then go open up the content_standalone.xml file in the
YouTubePlayer/app/src/main/res/layout folder. Replace the contents with the
following lines of code.

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  android:layout_width="match_parent"
  android:layout_height="match_parent"

```
    android:orientation="vertical"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/activity_standalone"
    tools:context="com.timbuchalka.youtubeplayer.StandaloneActivity">

    <Button
        android:id="@+id/btnPlayVideo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:text="Play Video"/>

    <Button
        android:id="@+id/btnPlaylist"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:text="Play Playlist"/>

</LinearLayout>
```

Then open up the StandaloneActivity and copy and paste in the GOOGLE_API_KEY and YOUTUBE_VIDEO_ID fields from YoutubeActivity to the StandaloneActivity class.

```
private String GOOGLE_API_KEY = "AIzaSyB4xh2zIJ5nUfytHeCb6huCQk-S_jvAHyo";
private String YOUTUBE_VIDEO_ID = "XUb97Wv9R6o";
```

 Then add in the following lines of code below it.

```
private String YOUTUBE_PLAYLIST_ID = "TBA";

private Button btnPlayVideo;
private Button btnPlaylist;
```

Next, edit "*public class YoutubeActivity extends YouTubeBaseActivity*" to "*public class YoutubeActivity extends YouTubeBaseActivity implements YouTubePlayer.OnInitializedListener*"

Then generate a Override Method, and select onClick View which is located way below the Override Method window.

You should get this lines of code

```
@Override
public void onClick(View v) {
```

```
        }
```

Next add the following lines in the onCreate method.

```
btnPlayVideo = (Button) findViewById(R.id.btnPlayVideo);
btnPlaylist = (Button) findViewById(R.id.btnPlaylist);
btnPlayVideo.setOnClickListener(this);
btnPlaylist.setOnClickListener(this);
```

The code in StandaloneActivity should look like this currently.

```
11
12 ⊙    public class StandaloneActivity extends AppCompatActivity implements View.OnClickListener{
13
14            private String GOOGLE_API_KEY = "AIzaSyB4xh2zIJ5nUfytHeCb6huCQk-S_jvAHyo";
15            private String YOUTUBE_VIDEO_ID = "XUb97Wv9R6o";
16            private String YOUTUBE_PLAYLIST_ID = "PLXtTjtWmQhg3aTiE0y_iO_DRlKtTiv519";
17
18            private Button btnPlayVideo;
19            private Button btnPlaylist;
20
21
22            @Override
23 ⊙        protected void onCreate(Bundle savedInstanceState) {
24                super.onCreate(savedInstanceState);
25                setContentView(R.layout.activity_standalone);
26                Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
27                setSupportActionBar(toolbar);
28                btnPlayVideo = (Button) findViewById(R.id.btnPlayVideo);
29                btnPlaylist = (Button) findViewById(R.id.btnPlaylist);
30                btnPlayVideo.setOnClickListener(this);
31                btnPlaylist.setOnClickListener(this);
32
33        //        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
34        //        fab.setOnClickListener(new View.OnClickListener() {
35        //            @Override
36        //            public void onClick(View view) {
37        //                Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
38        //                        .setAction("Action", null).show();
39        //            }
40        //        });
41            }
42
43            @Override
44 ⊙        public void onClick(View v) {
45            |
46            }
```

http://i.imgur.com/lovLJFm.png

Next, add these lines of code in the **public void onClick method**.

```
Intent intent = null;
switch(v.getId()) {
   case R.id.btnPlayVideo:
       intent = YouTubeStandalonePlayer.createVideoIntent(this, GOOGLE_API_KEY,
YOUTUBE_VIDEO_ID);
       break;
   case R.id.btnPlaylist:
```

```
        intent = YouTubeStandalonePlayer.createPlaylistIntent(this,
GOOGLE_API_KEY, YOUTUBE_PLAYLIST_ID);
        break;

    default:
  }
  if(intent != null) {
    startActivity(intent);
  }
```

Then search for a youtube playlist and paste its ID in the TBA area of the *"private String YOUTUBE_PLAYLIST_ID = "TBA";"*

Finally, go back to the AndroidManifest.xml file and add in the following lines within the <activity> tag of the StandaloneActivity.

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

It should end up looking like this.

```
        <activity
            android:name=".StandaloneActivity"
            android:label="@string/title_activity_standalone"
            android:theme="@style/AppTheme.NoActionBar" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
```

Then run the StandaloneActivity by right clicking on the StandaloneActivity on the project pane and select Run StandaloneActivity.

Lecture 46 – Youtube App – Finish and Wrap Up

Open up the content_main.xml and replace with the following lines of code.

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/activity_main"
    tools:context="com.timbuchalka.youtubeplayer.MainActivity">

    <Button
        android:id="@+id/btnPlaySingle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:text="Play Single Video"/>

    <Button
        android:id="@+id/btnStandalone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:text="Standalone SubMenu"/>

</LinearLayout>
```

Open up the MainActivity.java file and add in the following lines of code.

```java
private Button btnSingle;
private Button btnStandalone;
```

And then edit "*public class MainActivity extends AppCompatActivity*" to "*public class MainActivity extends AppCompatActivity implements View.OnClickListener*"

Then generate a Override Method, and select onClick View which is located way below the Override Method window.

You should get this lines of code

```
    @Override
    public void onClick(View v) {

    }
```

Then add the following lines in the onCreate method.

```
    btnSingle = (Button) findViewById(R.id.btnPlaySingle);
    btnStandalone = (Button) findViewById(R.id.btnStandalone);
    btnSingle.setOnClickListener(this);
    btnStandalone.setOnClickListener(this);
```

It should look like this currently.

```java
public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    private Button btnSingle;
    private Button btnStandalone;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        btnSingle = (Button) findViewById(R.id.btnPlaySingle);
        btnStandalone = (Button) findViewById(R.id.btnStandalone);
        btnSingle.setOnClickListener(this);
        btnStandalone.setOnClickListener(this);

//        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
//        fab.setOnClickListener(new View.OnClickListener() {
//            @Override
//            public void onClick(View view) {
//                Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
//                        .setAction("Action", null).show();
//            }
//        });
    }

    @Override
    public void onClick(View v) {

    }
```

Then add these following lines of code in the **public void onClick method**.

```
    Intent intent = null;
    switch(v.getId()) {
        case R.id.btnPlaySingle:
            intent = new Intent(MainActivity.this, YoutubeActivity.class);
```

```
        break;
    case R.id.btnStandalone:
        intent = new Intent(MainActivity.this, StandaloneActivity.class);
        break;

    default:
}
if(intent != null) {
    startActivity(intent);
}
```

Then run the MainActivity by right clicking on the MainActivity on the project pane and select Run MainActivity.