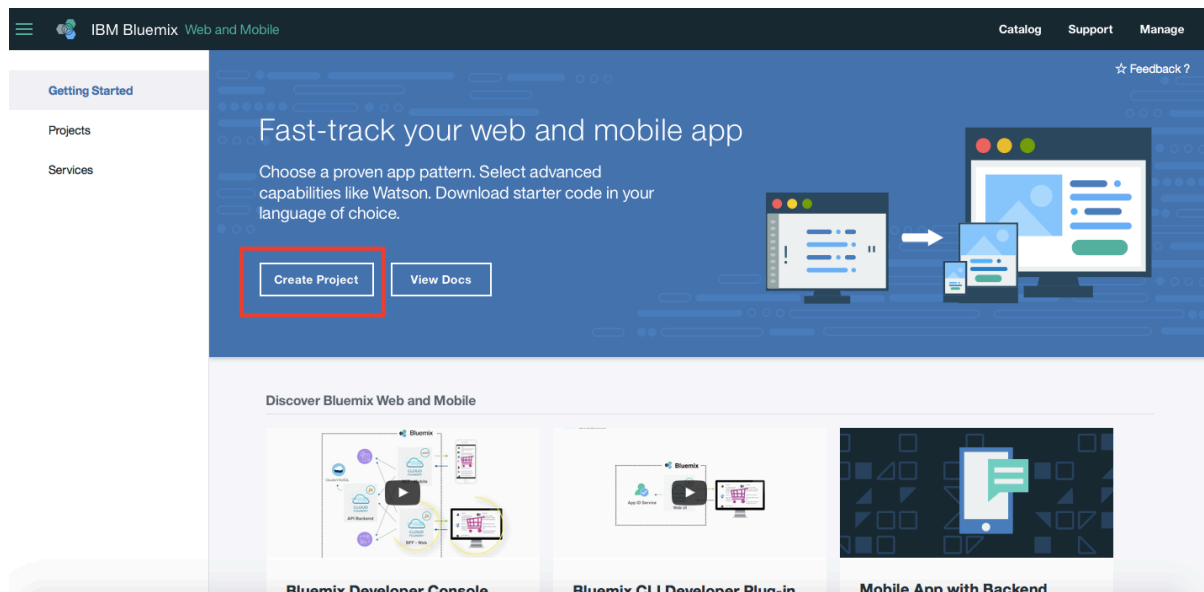
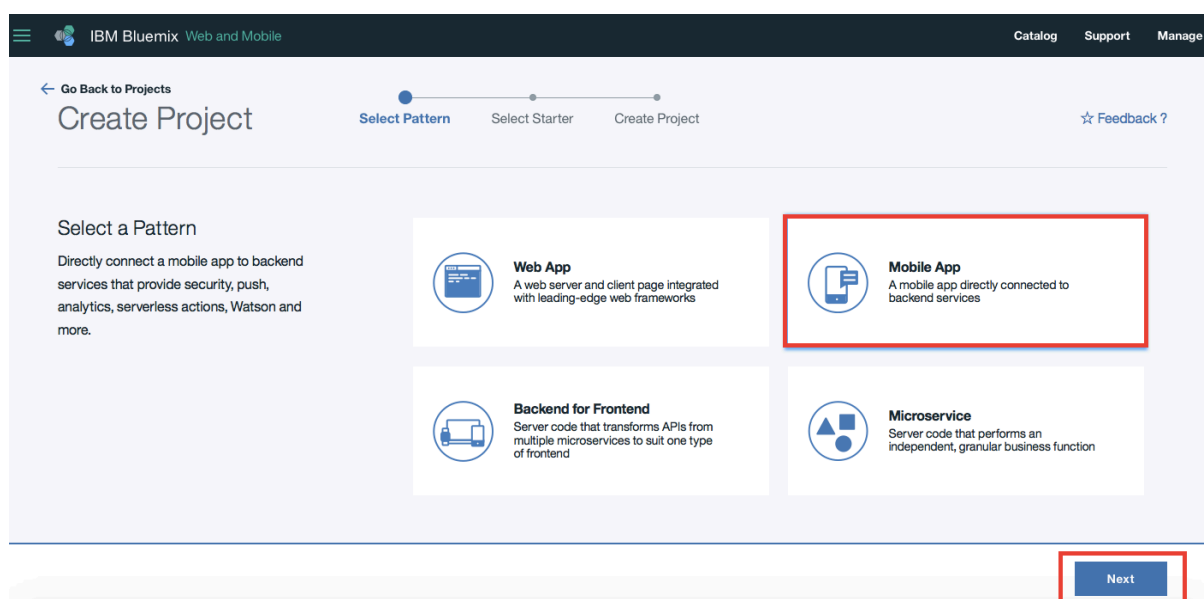


# Build a Watson powered Chat Bot using Mobile Services, Watson Conversation, Text to Speech

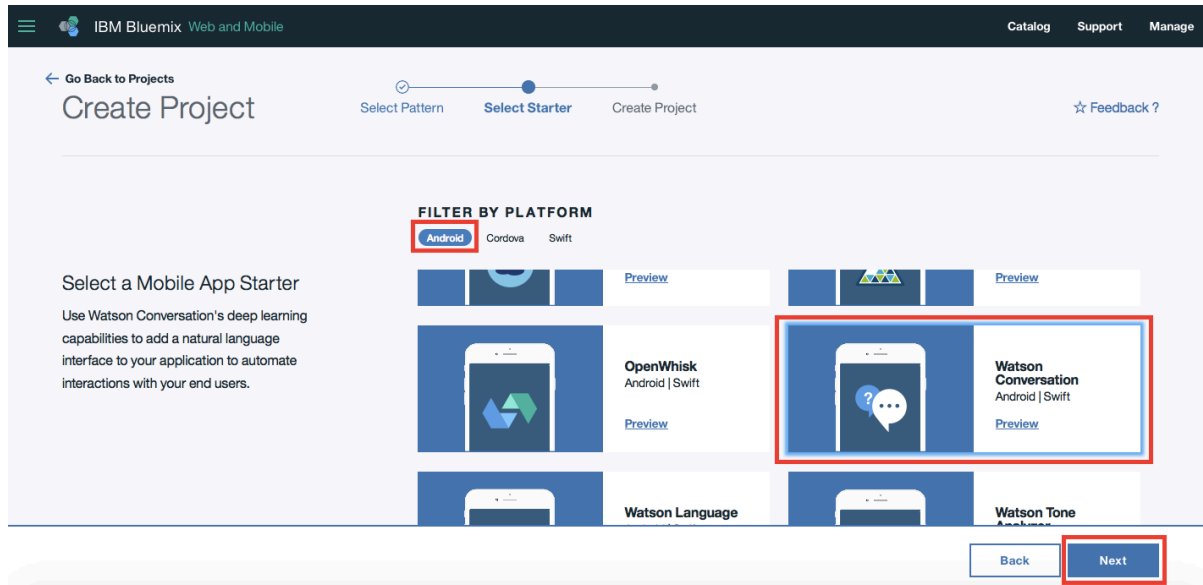
Step 1: Go to URL <https://console.ng.bluemix.net/developer/getting-started/> and click on **CREATE PROJECT**



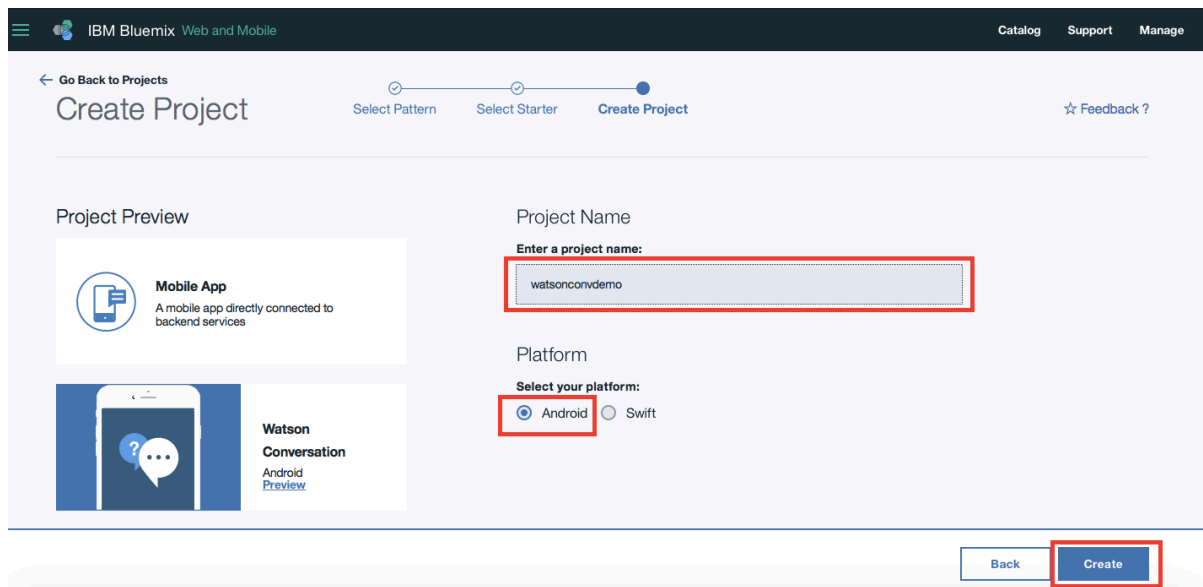
Step 2: Click on **MOBILE APP** and Click on **NEXT**



Step 3: Filter by **ANDROID** and then click on **Watson Conversation** and Click on **NEXT**



Step 4: Enter **Project Name** and Select platform as **Android** and click on **CREATE**



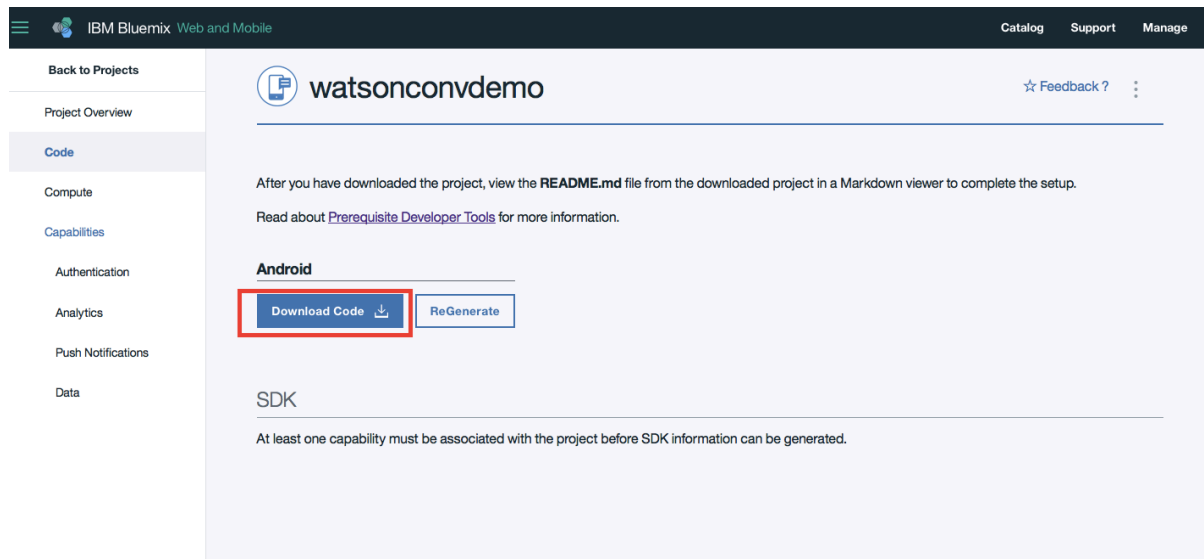
## Step 5: Click on **Get the Code**

The screenshot shows the IBM Bluemix Web and Mobile interface. The top navigation bar includes 'Catalog', 'Support', and 'Manage'. The left sidebar lists navigation options: 'Back to Projects', 'Project Overview' (selected), 'Code', 'Compute', 'Capabilities', 'Authentication', 'Analytics', 'Push Notifications', and 'Data'. The main content area for 'watsonconvdemo' is divided into three sections: 'PROJECT OVERVIEW', 'CODE', and 'CAPABILITIES'. The 'PROJECT OVERVIEW' section contains the text: 'Use Watson Conversation's deep learning capabilities to add a natural language interface to your application to automate interactions with your end users.' Below this, it lists 'Date Created: April 10, 2017', 'Pattern: Mobile App', and 'Starter: Watson Conversation'. The 'CODE' section contains the text: 'Generate and download the code for your configured project.' and a blue button labeled 'Get the Code' which is highlighted with a red rectangle. The 'CAPABILITIES' section contains the text: 'Enhance your project by adding these capabilities. Check back often to stay current with new capabilities.' and shows four capability icons.

## Step 6: Click on **Generate Code**

The screenshot shows the IBM Bluemix Web and Mobile interface. The top navigation bar includes 'Catalog', 'Support', and 'Manage'. The left sidebar lists navigation options: 'Back to Projects', 'Project Overview', 'Code' (selected), 'Compute', 'Capabilities', 'Authentication', 'Analytics', 'Push Notifications', and 'Data'. The main content area for 'watsonconvdemo' is divided into two sections: 'Android' and 'SDK'. The 'Android' section contains the text: 'After you have downloaded the project, view the README.md file from the downloaded project in a Markdown viewer to complete the setup. Read about [Prerequisite Developer Tools](#) for more information.' Below this, there is a blue button labeled 'Generate Code' with a plus icon, which is highlighted with a red rectangle. The 'SDK' section contains the text: 'At least one capability must be associated with the project before SDK information can be generated.'

## Step 7: Download the generated code.



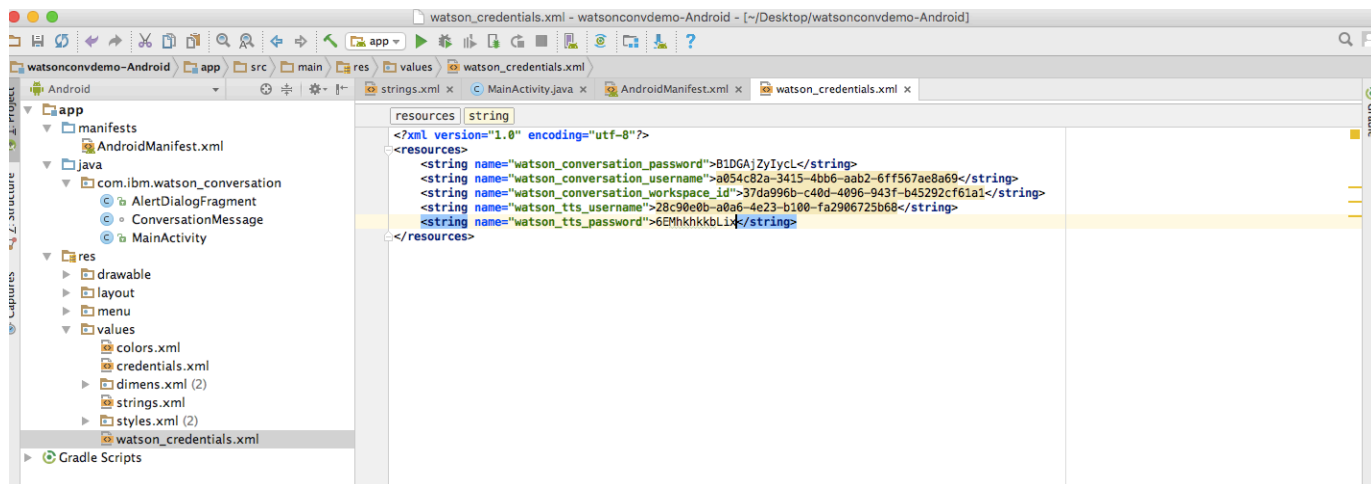
## Step 8: Unzip the generated code and then Import it into Android Studio

## Step 9: In the build.gradle(app), insert dependency

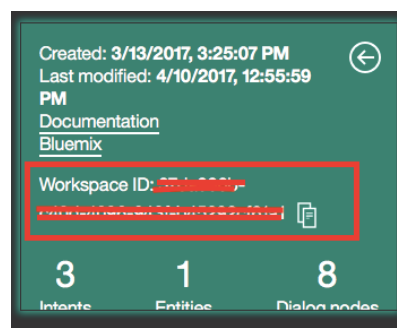
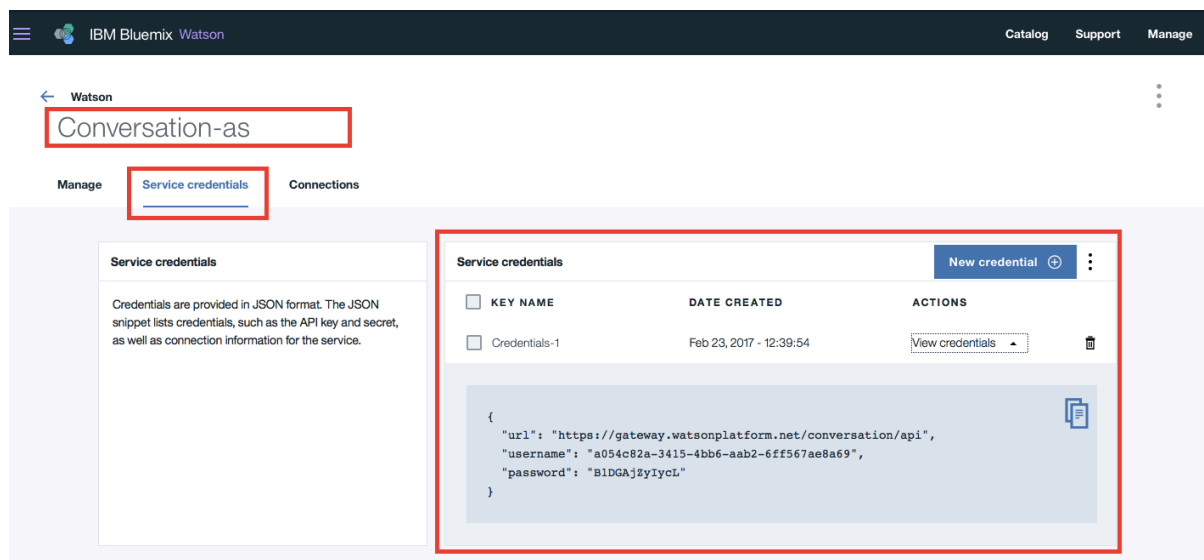
```
compile 'com.ibm.watson.developer_cloud:android-sdk:0.2.3'
```

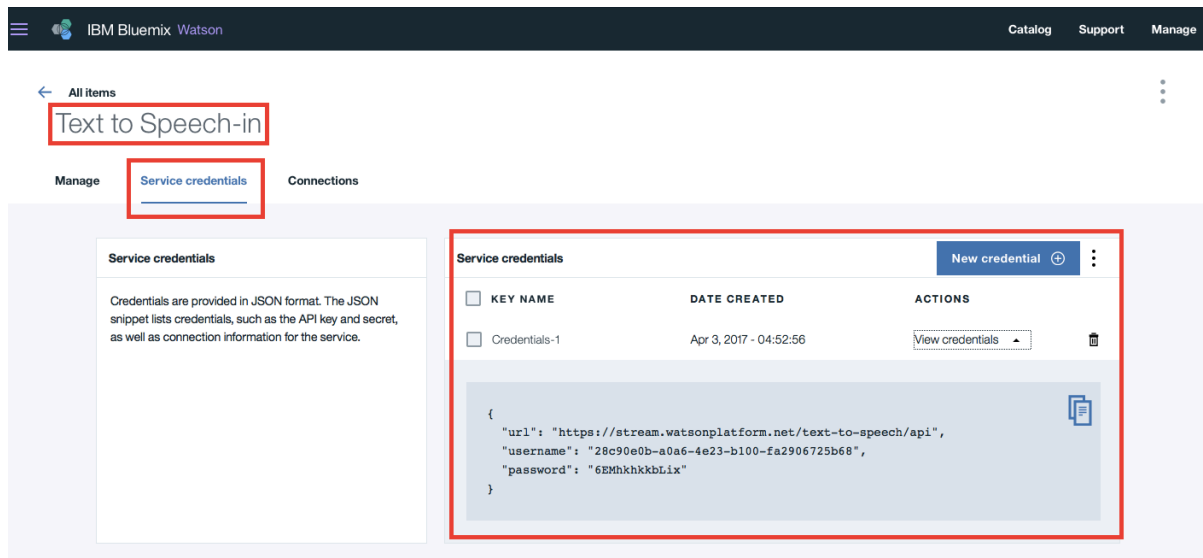
## Step 10: In res>values>watson\_credentials.xml enter the Watson Conversation password, Username, Workspace ID and insert last two lines to insert Text to Speech Username & Password

```
<resources>
<string name="watson_conversation_password">B1DGAjZyJycL</string>
<string name="watson_conversation_username">a054c82a-3415-4bb6-aab2-6ff567ae8a99</string>
<string name="watson_conversation_workspace_id">37da996b-c40d-4096-943f-b45292cf91a1</string>
<string name="watson_tts_username">28c90e0b-a0a6-4e23-b100-fa2906725b69</string>
<string name="watson_tts_password">6EMhkhkbbLpx</string>
</resources>
```



You will get the credentials when you create Watson Conversation Service & Text to Speech Service in Bluemix





Step 11:

```
private TextToSpeech initTextToSpeechService()
{
    TextToSpeech service = new TextToSpeech();

    service.setUsernameAndPassword(getString(R.string.watson_tts_u
        sername),getString(R.string.watson_tts_password));

    return service;
}
```

Step 12: Import

```
import com.ibm.watson.developer_cloud.text_to_speech.v1.TextToSpeech;
import com.ibm.watson.developer_cloud.text_to_speech.v1.model.Voice;
import com.ibm.watson.developer_cloud.android.library.audio.StreamPlayer;
```

Step 13: Create the method WatsonTask.

```
StreamPlayer streamPlayer;
```

```
private class WatsonTask extends AsyncTask<String, Void,  
String> {  
    @Override  
    protected String doInBackground(String...textToSpeech) {  
  
        TextToSpeech textToSpeech = initTextToSpeechService();  
        streamPlayer = new StreamPlayer();  
  
        streamPlayer.playStream(textToSpeech.synthesize(textToSpeech[0  
], Voice.EN_ALLISON).execute());  
        return "textToSpeech";  
    }  
  
    @Override  
    protected void onPostExecute(String result) {  
  
    }  
}
```

Step 14:

Call the method inside

```
addMessageFromUser(ConversationMessage message) {  
    if (message.getUser().equals(USER_Watson)){  
        .....  
        .....  
        .....  
        new WatsonTask().execute(message.getMessageText());  
    }
```