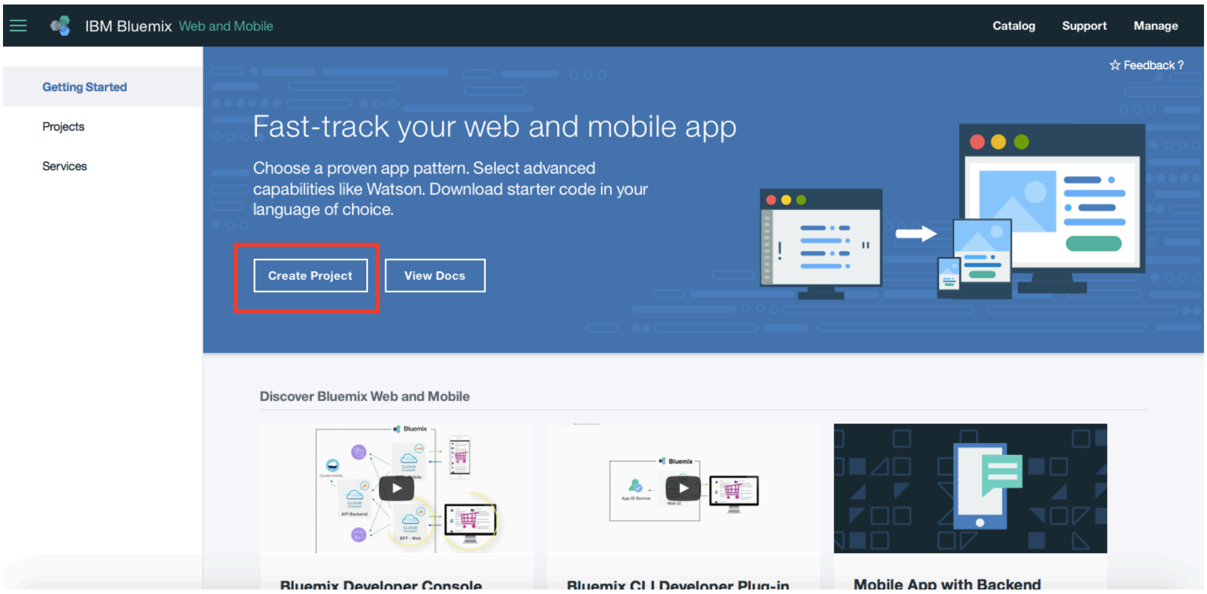
Build a Watson powered Chat Bot using Mobile Services, Watson Conversation, Text to Speech

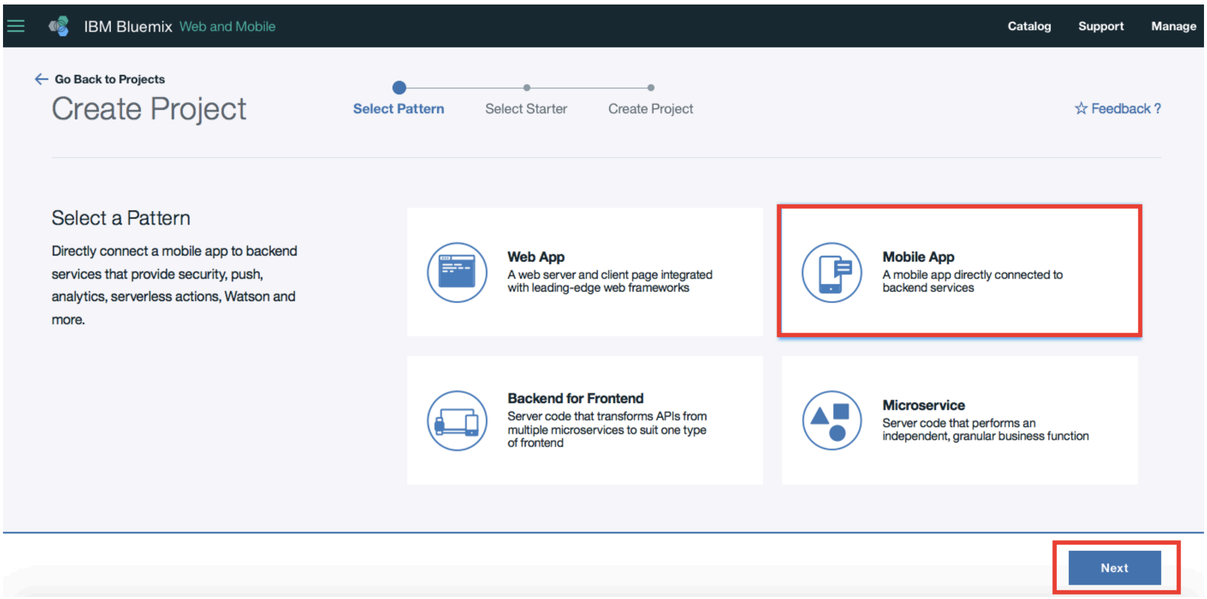
Step 1: Go to URL https://console.ng.bluemix.net/developer/getting-started/ and click on **CREATE PROJECT**


IBM Bluemix Web and Mobile
Catalog
Support
Manage
Getting Started
Projects
Services
☆ Feedback ?
Create Project
View Docs
Discover Bluemix Web and Mobile
Bluemix Developer Console
Bluemix CLI Developer Plug-in
Mobile App with Backend

Fast-track your web and mobile app

Choose a proven app pattern. Select advanced capabilities like Watson. Download starter code in your language of choice.

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



Create Project

Select a Pattern

Directly connect a mobile app to backend services that provide security, push, analytics, serverless actions, Watson and more.

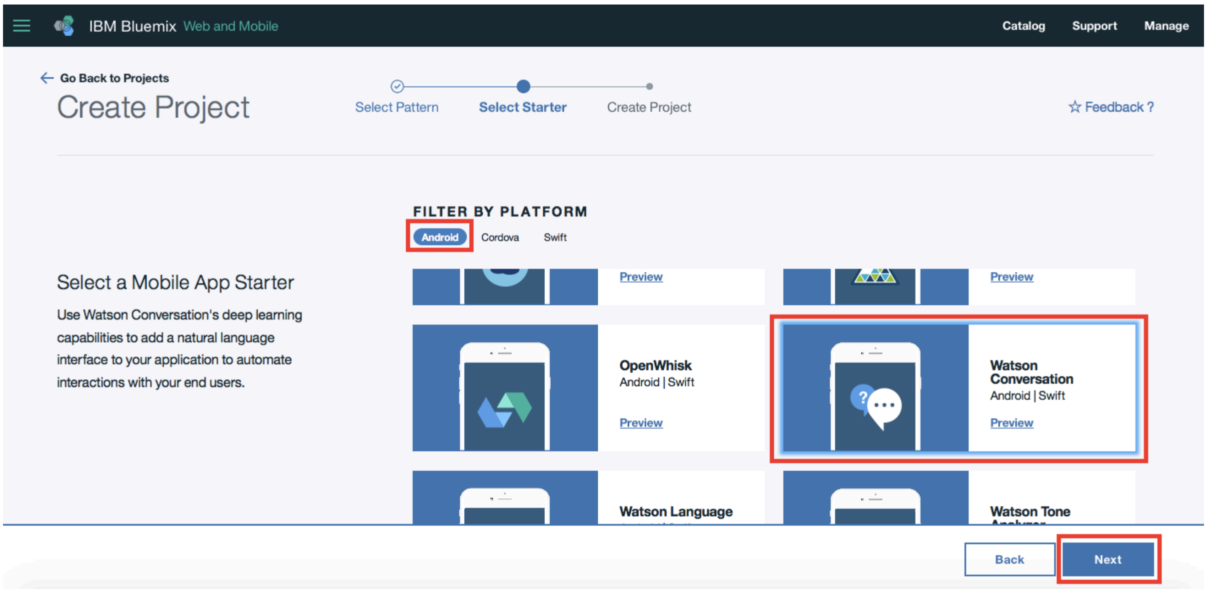
Web App A web server and client page integrated with leading-edge web frameworks

Mobile App A mobile app directly connected to backend services

Backend for Frontend Server code that transforms APIs from multiple microservices to suit one type of frontend

Microservice Server code that performs an independent, granular business function

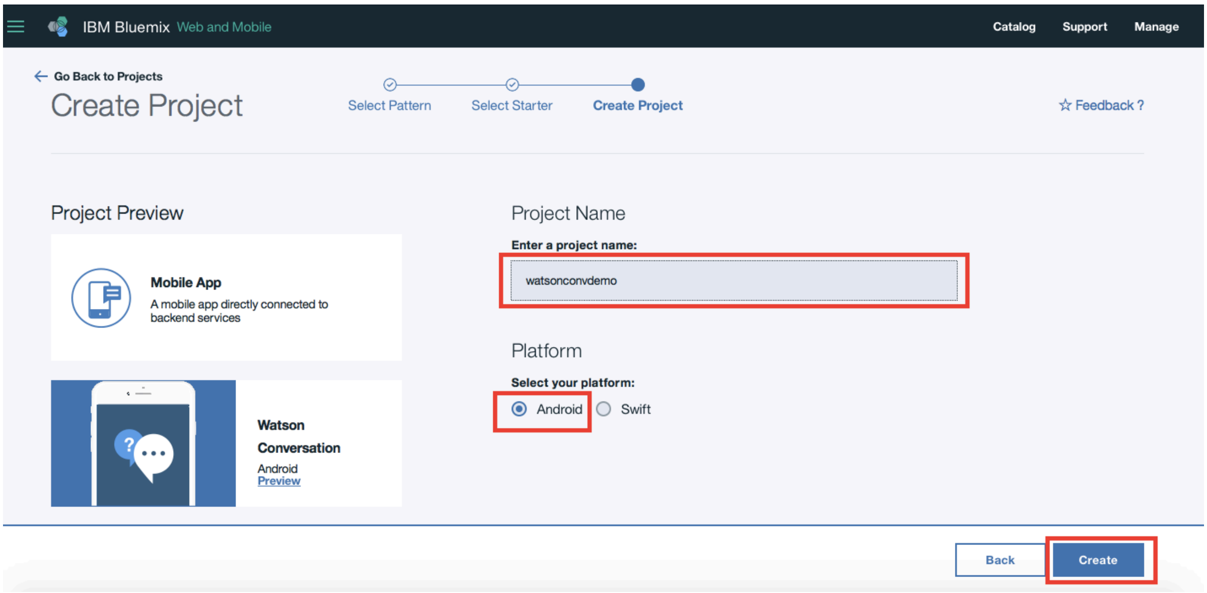
Step 3: Filter by **ANDROID** and then click on **Watson Conversation** and Click on **NEXT**



Select a Mobile App Starter

Use Watson Conversation's deep learning capabilities to add a natural language interface to your application to automate interactions with your end users.

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



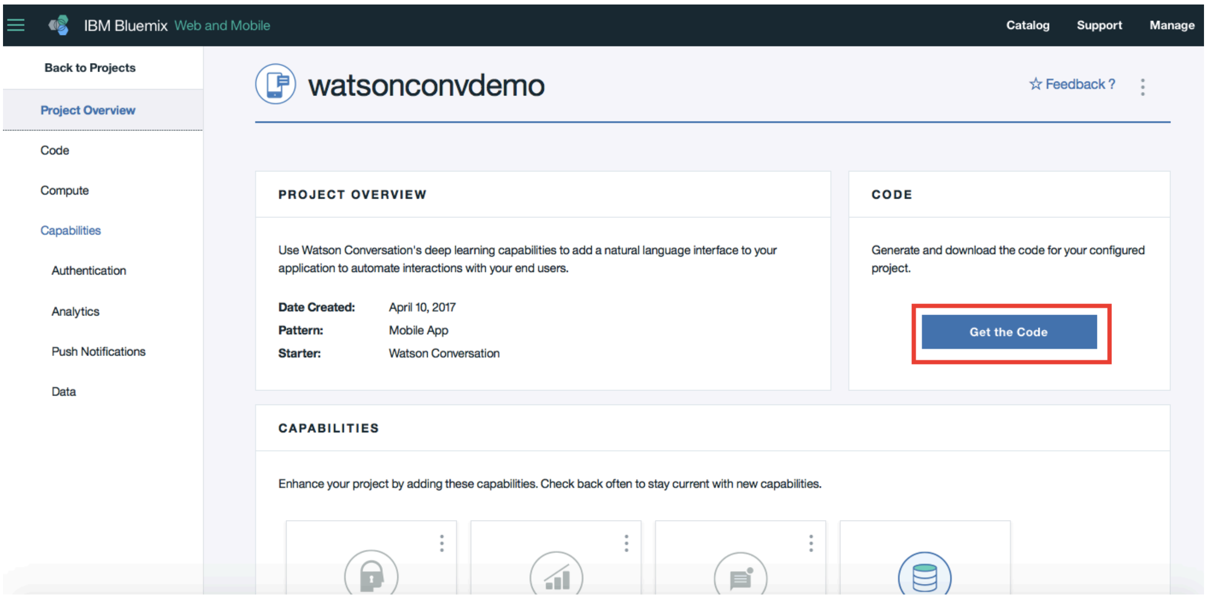
Project Name

Enter a project name: watsonconvdemo

Platform

Select your platform: ● Android ○ Swift

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



watsonconvdemo

PROJECT OVERVIEW

Use Watson Conversation's deep learning capabilities to add a natural language interface to your application to automate interactions with your end users.

Date Created: April 10, 2017
Pattern: Mobile App
Starter: Watson Conversation

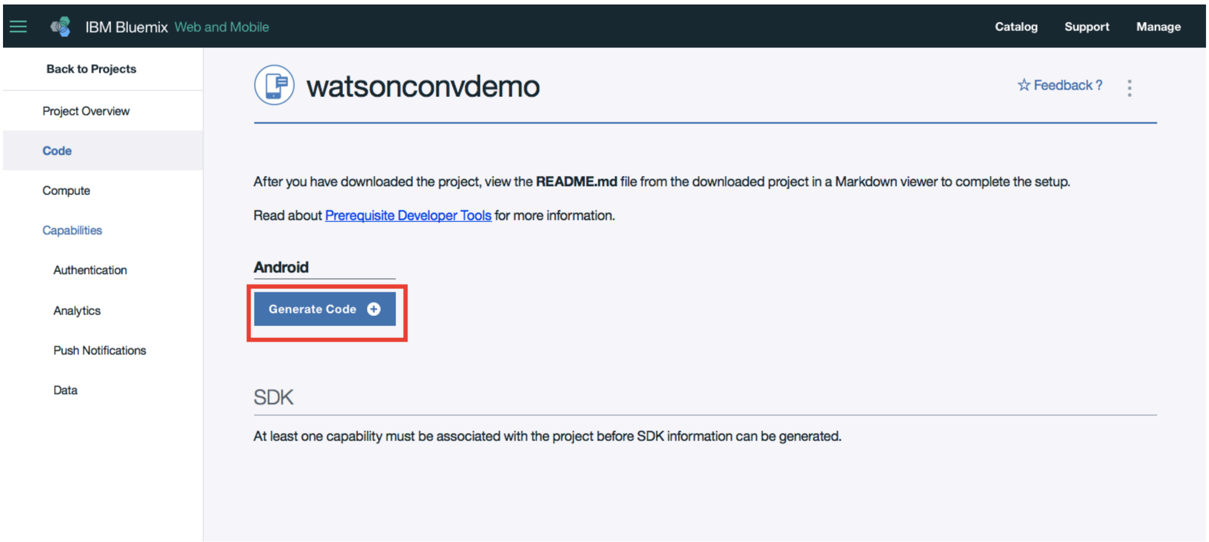
CODE

Generate and download the code for your configured project.

CAPABILITIES

Enhance your project by adding these capabilities. Check back often to stay current with new capabilities.

Step 6: Click on **Generate Code**



watsonconvdemo

After you have downloaded the project, view the **README.md** file from the downloaded project in a Markdown viewer to complete the setup.

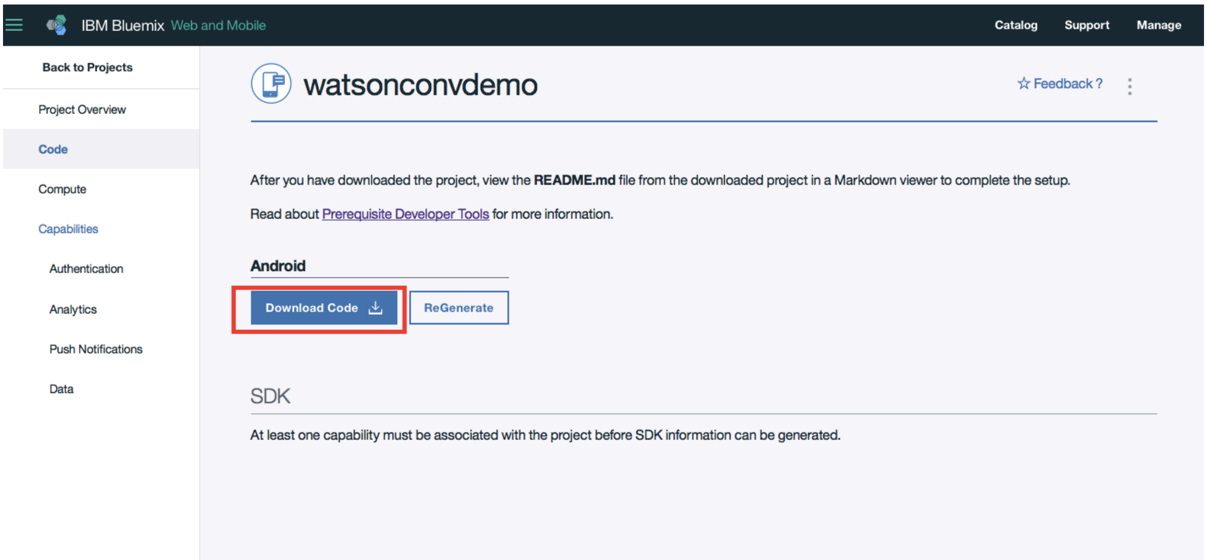
Read about Prerequisite Developer Tools for more information.

Android

SDK

At least one capability must be associated with the project before SDK information can be generated.

Step 7: **Download the generated code.**



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.

At least one capability must be associated with the project before SDK information can be generated.

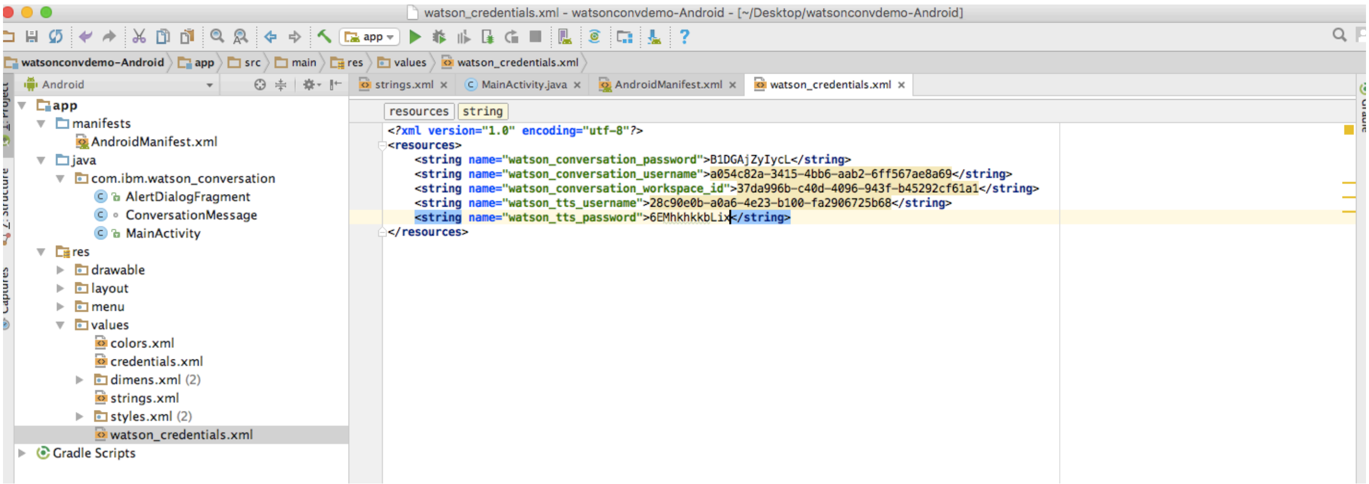
Step 8: Unzip the generated code and then Import it into Android Studio

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

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

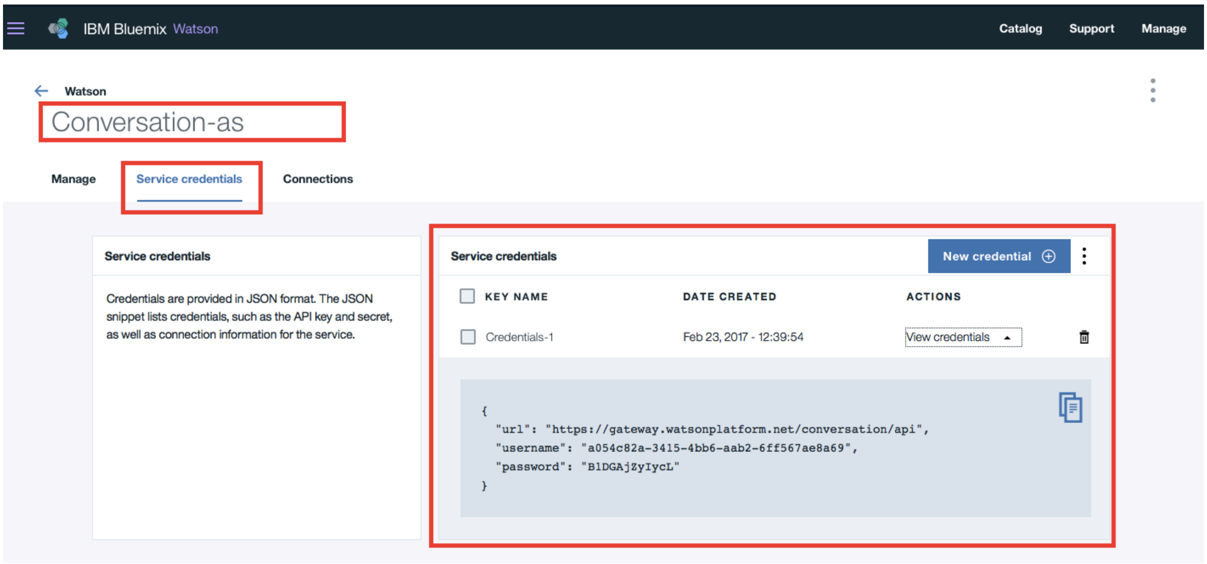
Step 10: In res>values>watson_credentials.xml enter the Watson Conversation password, Username, Workspace ID and insert last two lines to insert Text to Speech Username & Password

```
<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">6EMhkhkkbLpx</string>
</resources>
```



```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="watson_conversation_password">B1DGAjZyIycL</string>
    <string name="watson_conversation_username">a054c82a-3415-4bb6-aab2-6ff567ae8a69</string>
    <string name="watson_conversation_workspace_id">37da996b-c40d-4096-943f-b45292cf61a1</string>
    <string name="watson_tts_username">28c90e0b-a0a6-4e23-b100-fa2906725b68</string>
    <string name="watson_tts_password">6EMhkhkkbLix</string>
</resources>
```

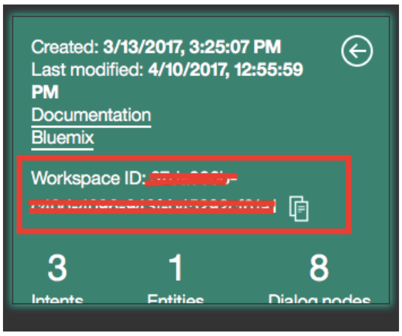
You will get the credentials when you create Watson Conversation Service & Text to Speech Service in Bluemix



Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service.

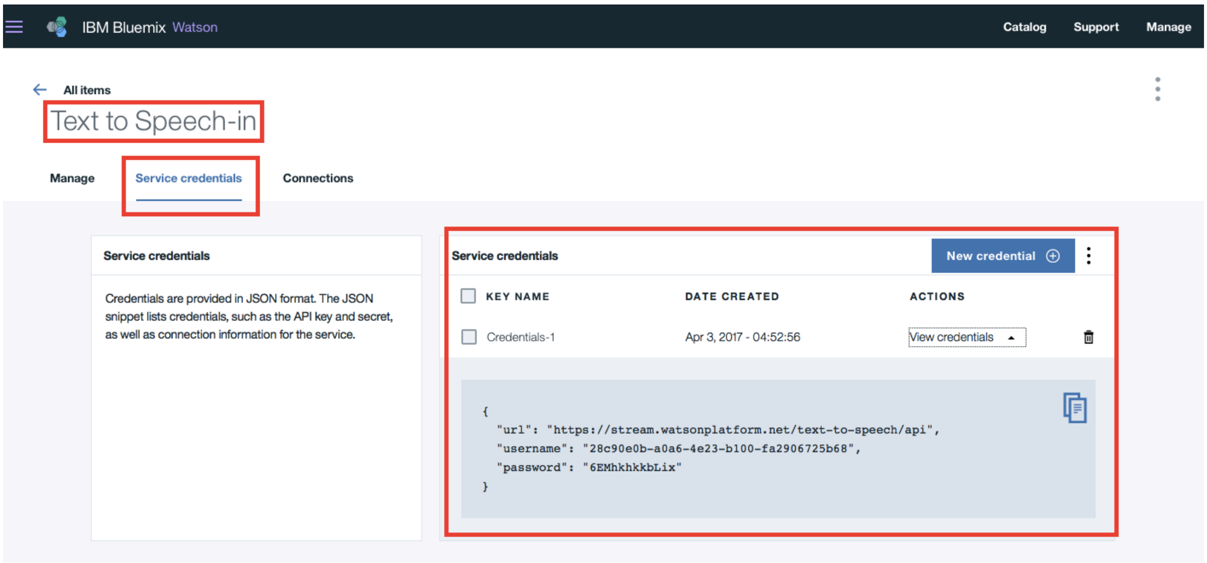
□ KEY NAME	DATE CREATED	ACTIONS
□ Credentials-1	Feb 23, 2017 - 12:39:54	View credentials ▲

```
{
  "url": "https://gateway.watsonplatform.net/conversation/api",
  "username": "a054c82a-3415-4bb6-aab2-6ff567ae8a69",
  "password": "B1DGAjZyIycL"
}
```



Created: **3/13/2017, 3:25:07 PM**
Last modified: **4/10/2017, 12:55:59 PM**
Documentation
Bluemix

Workspace ID: [redacted]



Service credentials

Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service.

Service credentials

	KEY NAME	DATE CREATED	ACTIONS
□	Credentials-1	Apr 3, 2017 - 04:52:56	View credentials

```
{
  "url": "https://stream.watsonplatform.net/text-to-speech/api",
  "username": "28c90e0b-a0a6-4e23-b100-fa2906725b68",
  "password": "6EMhkhkkbLix"
}
```

Step 11:

```
private TextToSpeech initTextToSpeechService()

{

TextToSpeech service = new TextToSpeech();

service.setUsernameAndPassword(getString(R.string.watson_tts_username),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) {

        String MessageText = texttoSpeech[0];

        TextToSpeech textToSpeech = initTextToSpeechService();
        streamPlayer = new StreamPlayer();

streamPlayer.playStream(textToSpeech.synthesize(MessageText,
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());
}
```