IBM Watson Chatbots

IBM Watson

IBM Watson

Conversation

Add a natural language interface to your application

Developer Friendly

Easy to begin, easy to use. Get faster time to value, and integrate across channels, networks and environments.

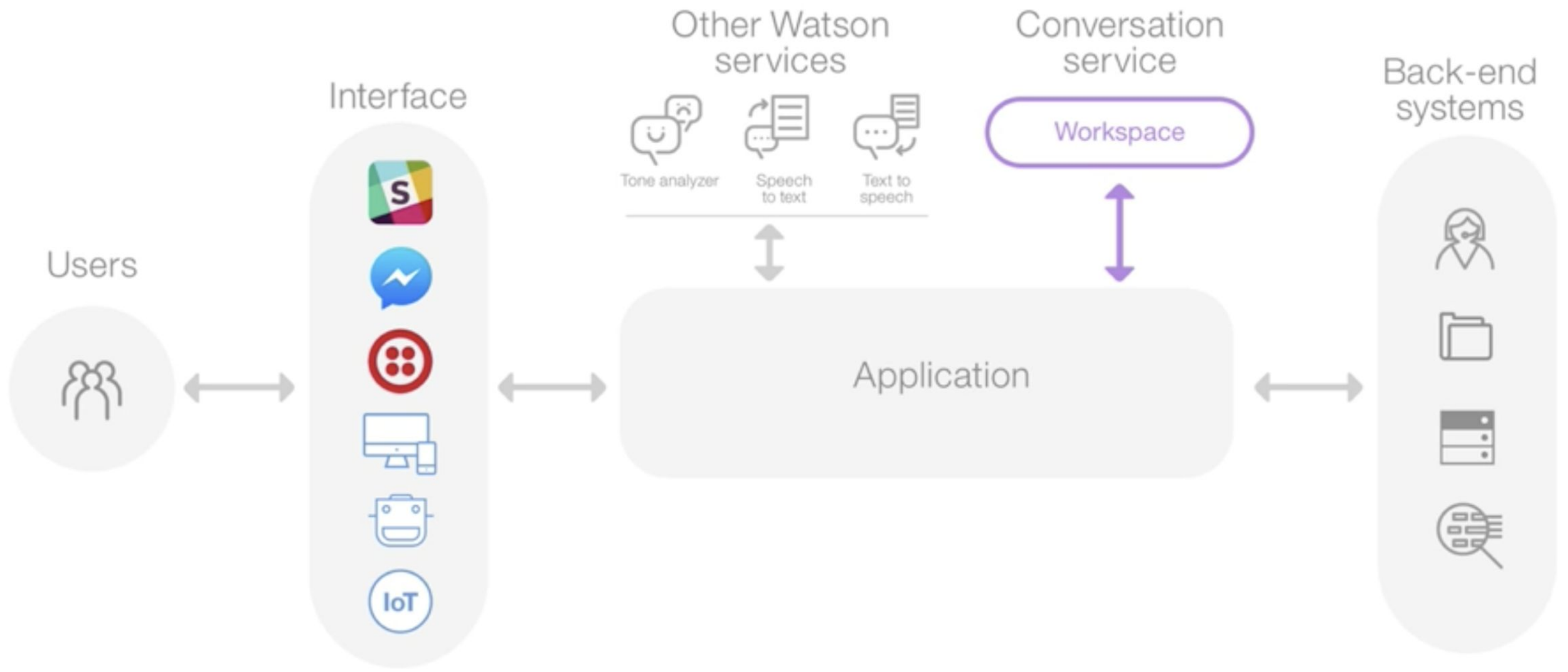
Enterprise Grade

Conversation features a reliable infrastructure that scales with individual use cases.

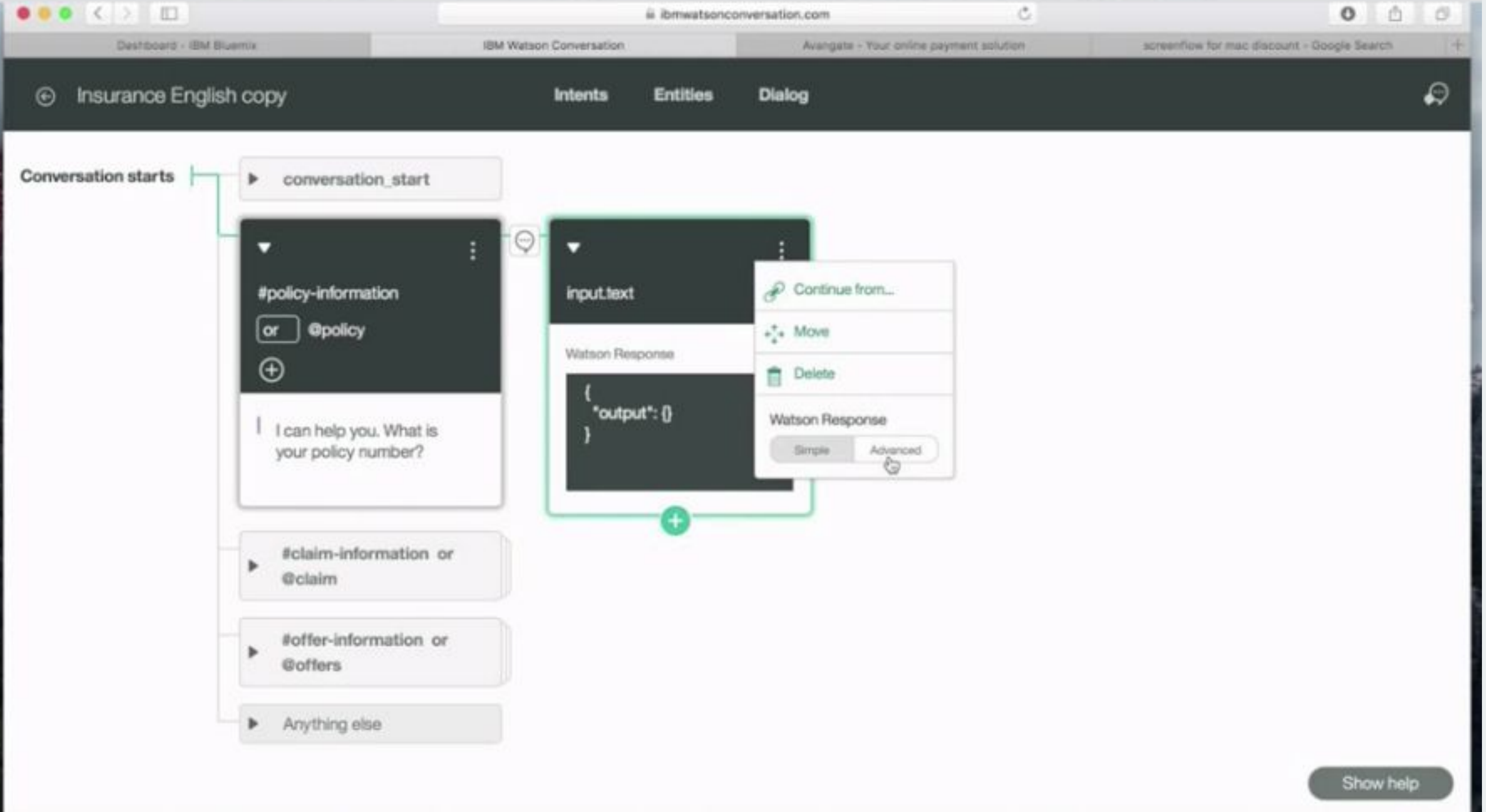
Chatbots made easy

Understands natural-language, connect to messaging channels, web environments

How does IBM Watson work?



IBM Watson Interface

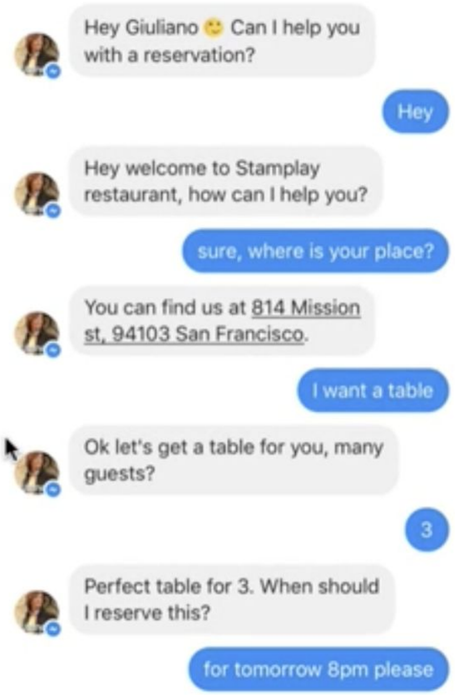


I can help you. What is your policy number?

Watson Response

```
{
  "output": {}
}
```

Example: Reservation Bot



Hey Giuliano 🙂 Can I help you with a reservation?

Hey

Hey welcome to Stamplay restaurant, how can I help you?

sure, where is your place?

You can find us at 814 Mission st, 94103 San Francisco.

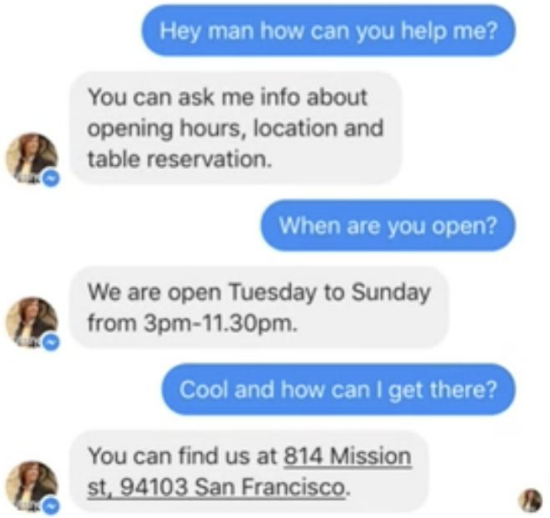
I want a table

Ok let's get a table for you, many guests?

3

Perfect table for 3. When should I reserve this?

for tomorrow 8pm please



Hey man how can you help me?

You can ask me info about opening hours, location and table reservation.

When are you open?

We are open Tuesday to Sunday from 3pm-11.30pm.

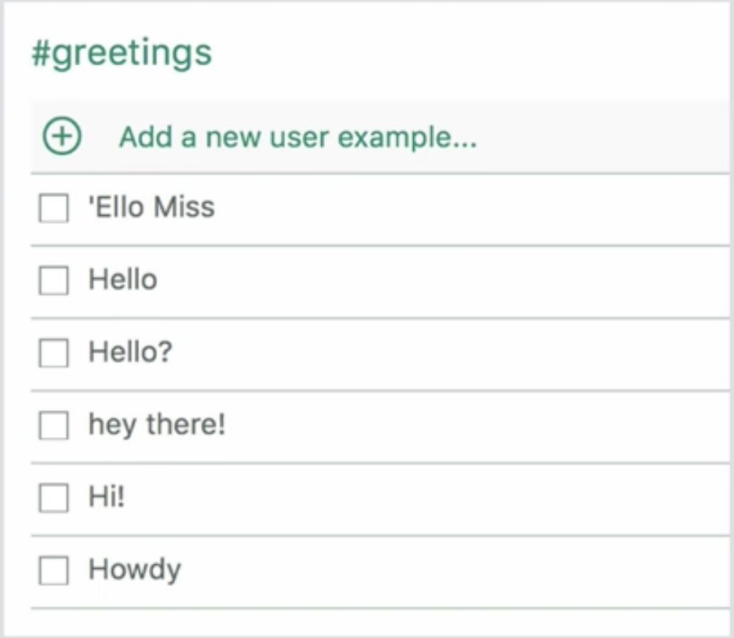
Cool and how can I get there?

You can find us at 814 Mission st, 94103 San Francisco.

IBM Watson Concepts

Intents Things that the chat users are looking to do: change passwords, get status updates, etc

Example



#greetings

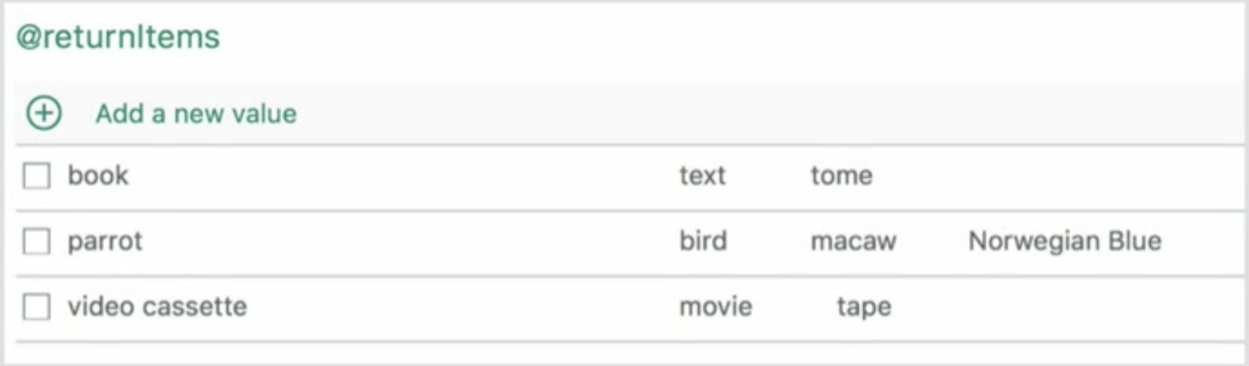
⊕ Add a new user example...

- ☐ 'Ello Miss
- ☐ Hello
- ☐ Hello?
- ☐ hey there!
- ☐ Hi!
- ☐ Howdy

IBM Watson Concepts

Entities Specific pieces of information to extract from a user response.

Example



@returnItems

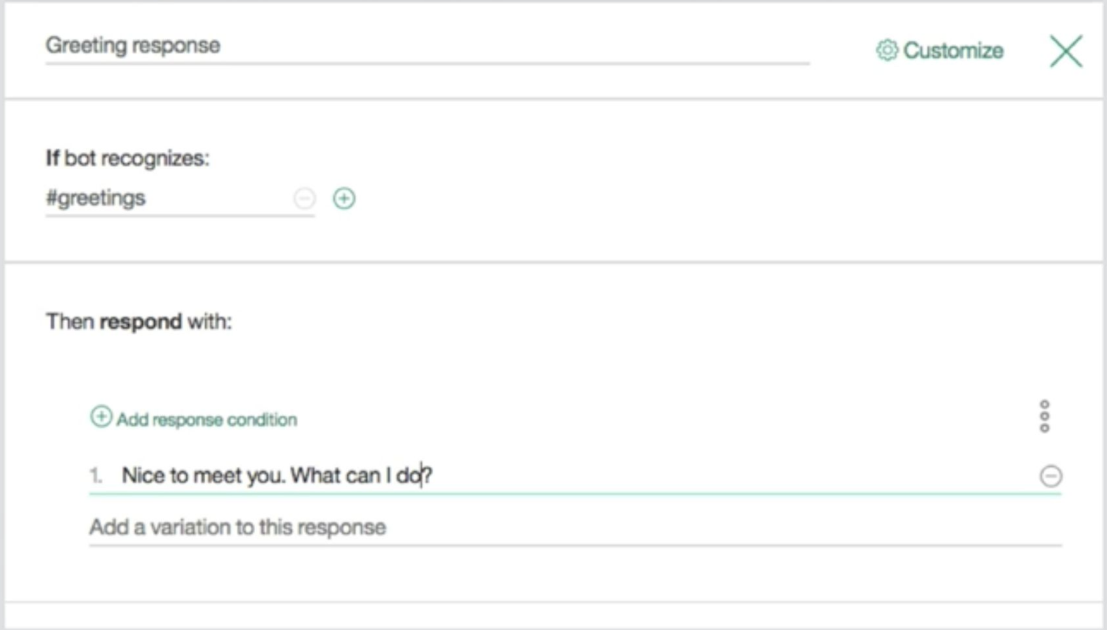
Add a new value

book	text	tome	
parrot	bird	macaw	Norwegian Blue
video cassette	movie	tape	

IBM Watson Concepts

Dialog Flow When entities and intents are specified, you can move on to constructing the dialog flow.

Example



Greeting response

Customize

If bot recognizes:

#greetings

Then **respond** with:

Add response condition

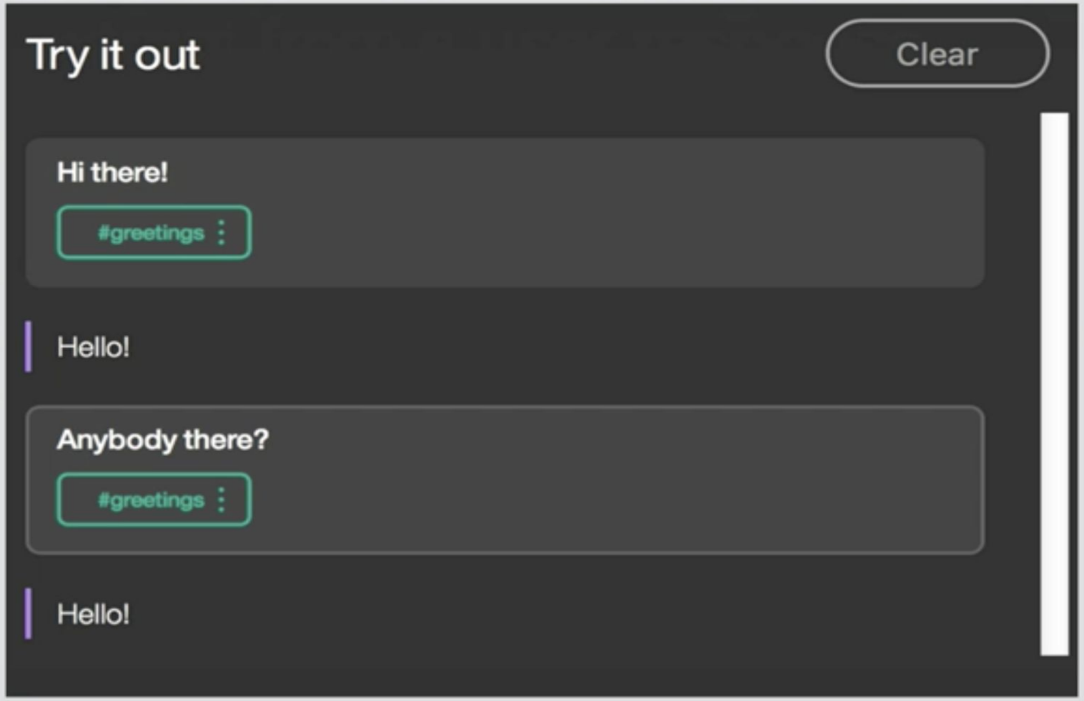
1. Nice to meet you. What can I do?

Add a variation to this response

IBM Watson Concepts

Testing Space to test your bot in action.

Example



Hi there!

#greetings

Hello!

Anybody there?

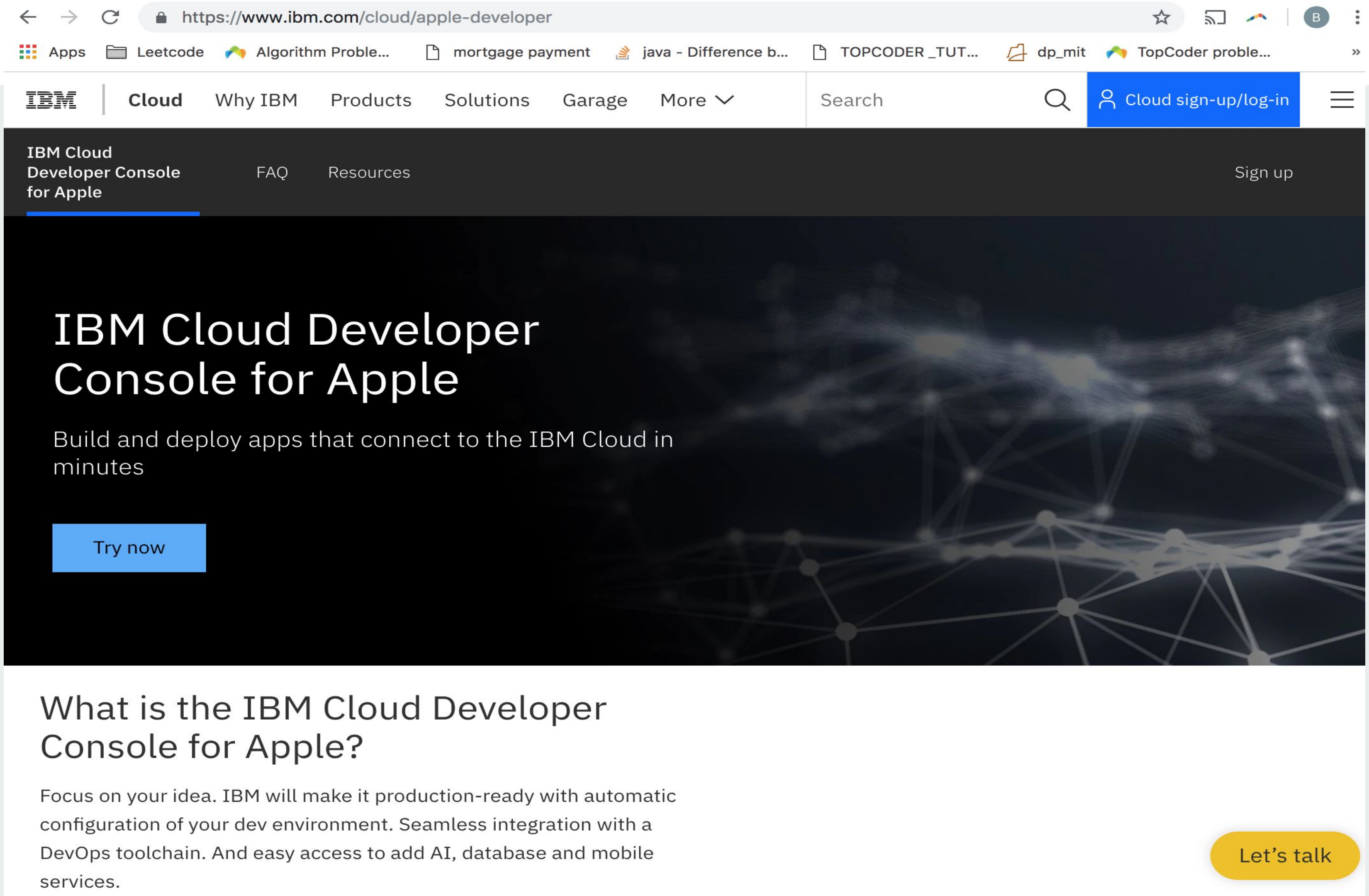
#greetings

Hello!

Let’s build an iOS and Android app

Integrating Watson on iOS

1. Create IBM Cloud account for free
2. Create an iOS app on the cloud and download the source code
3. Inject Watson dependencies with cocoapods
4. Update workspaceID with your newly created Watson Assistant
5. Use Watson chatbot to create a customer service app
6. Create Intents, Entities and Dialogs on IBM Cloud
7. Demo live working app

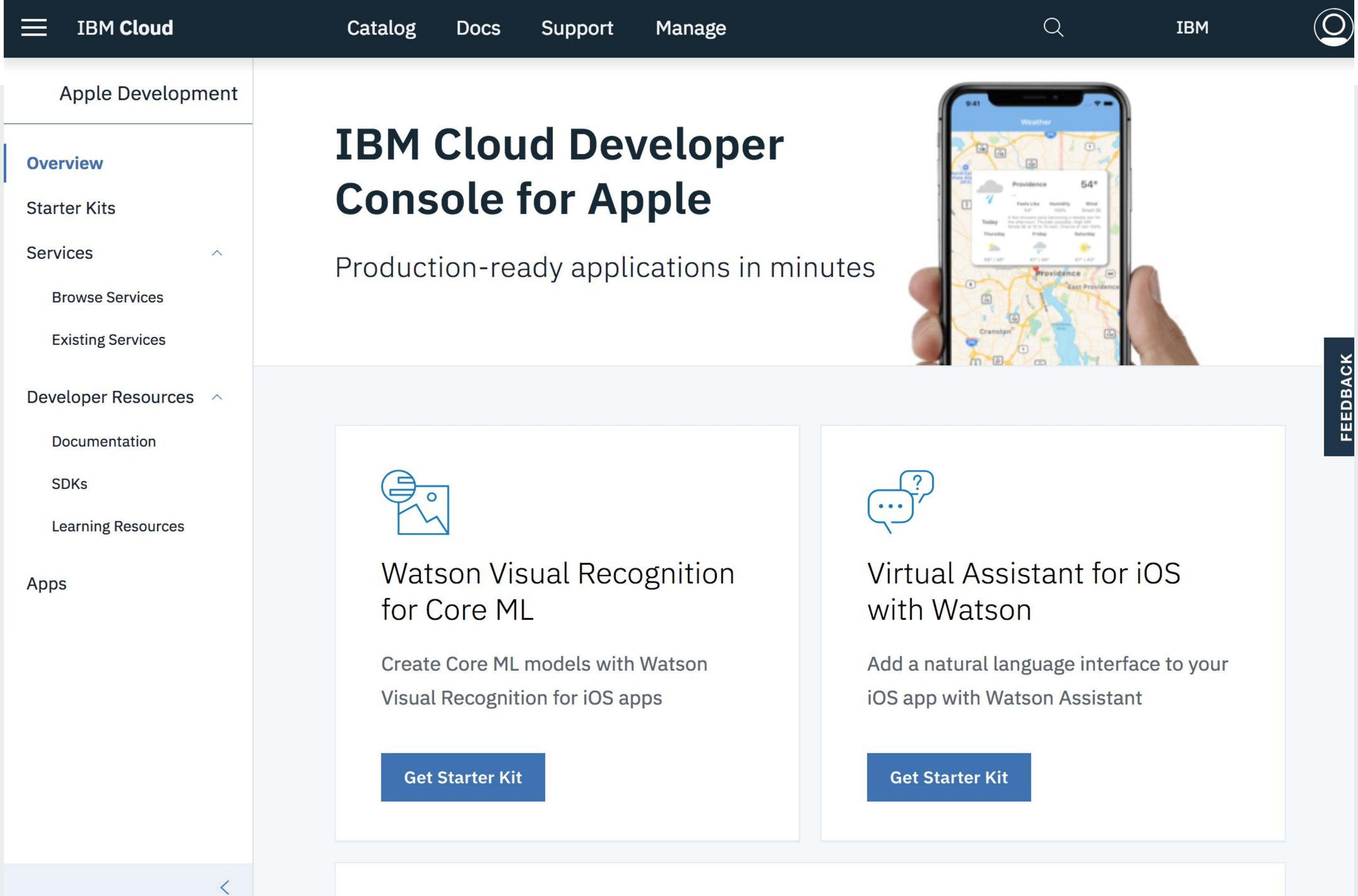


IBM Cloud Developer Console for Apple

Build and deploy apps that connect to the IBM Cloud in minutes

What is the IBM Cloud Developer Console for Apple?

Focus on your idea. IBM will make it production-ready with automatic configuration of your dev environment. Seamless integration with a DevOps toolchain. And easy access to add AI, database and mobile services.


IBM **Cloud** Catalog Docs Support Manage IBM

Apple Development

Overview

Starter Kits

Services

Browse Services

Existing Services

Developer Resources

Documentation

SDKs

Learning Resources

Apps

IBM Cloud Developer Console for Apple

Production-ready applications in minutes

FEEDBACK

Watson Visual Recognition for Core ML

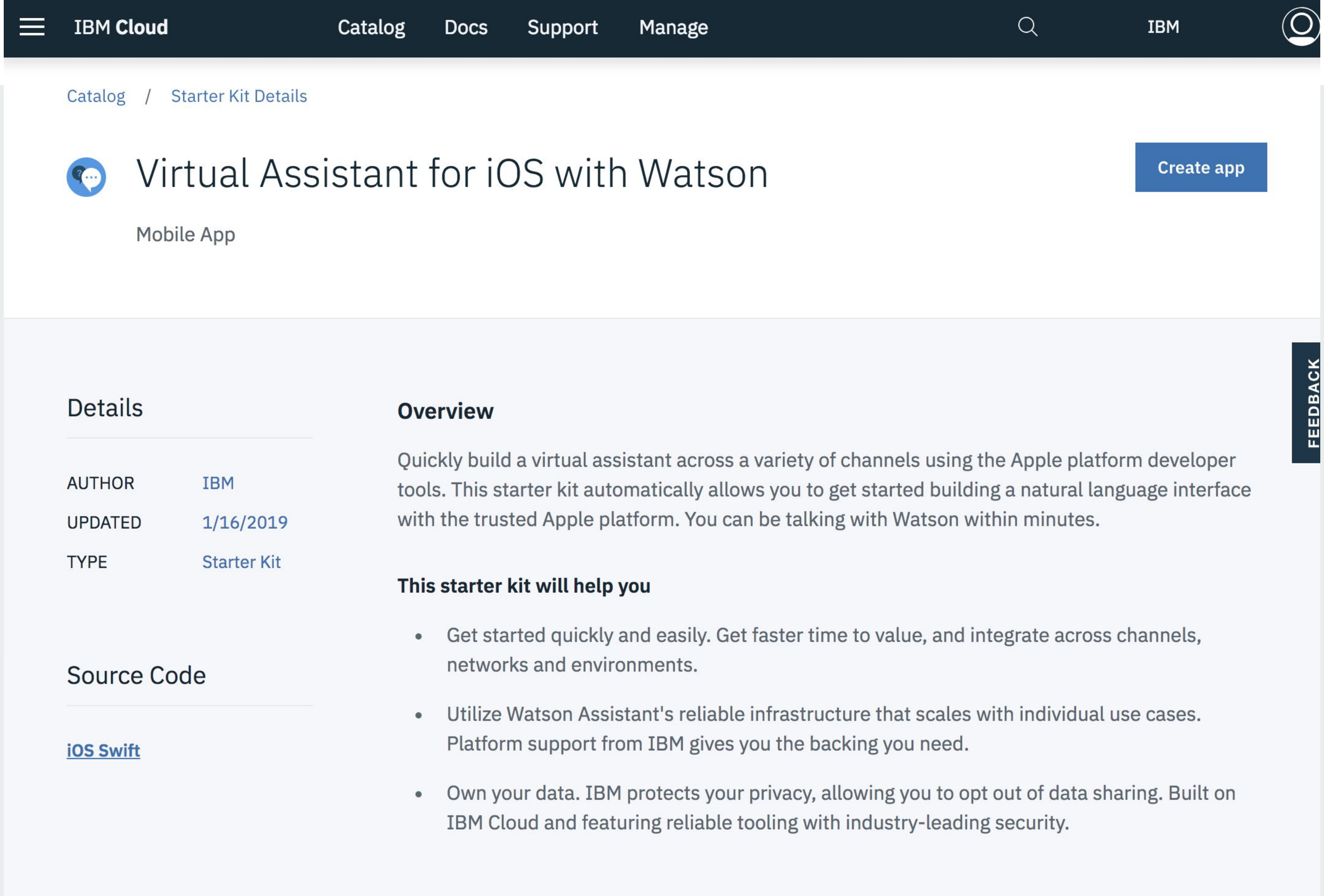
Create Core ML models with Watson Visual Recognition for iOS apps

Get Starter Kit

Virtual Assistant for iOS with Watson

Add a natural language interface to your iOS app with Watson Assistant

Get Starter Kit



IBM Cloud Catalog Docs Support Manage IBM

Catalog / Starter Kit Details

Virtual Assistant for iOS with Watson

Create app

Mobile App

FEEDBACK

Details

AUTHOR	IBM
UPDATED	1/16/2019
TYPE	Starter Kit

Source Code

iOS Swift

Overview

Quickly build a virtual assistant across a variety of channels using the Apple platform developer tools. This starter kit automatically allows you to get started building a natural language interface with the trusted Apple platform. You can be talking with Watson within minutes.

This starter kit will help you

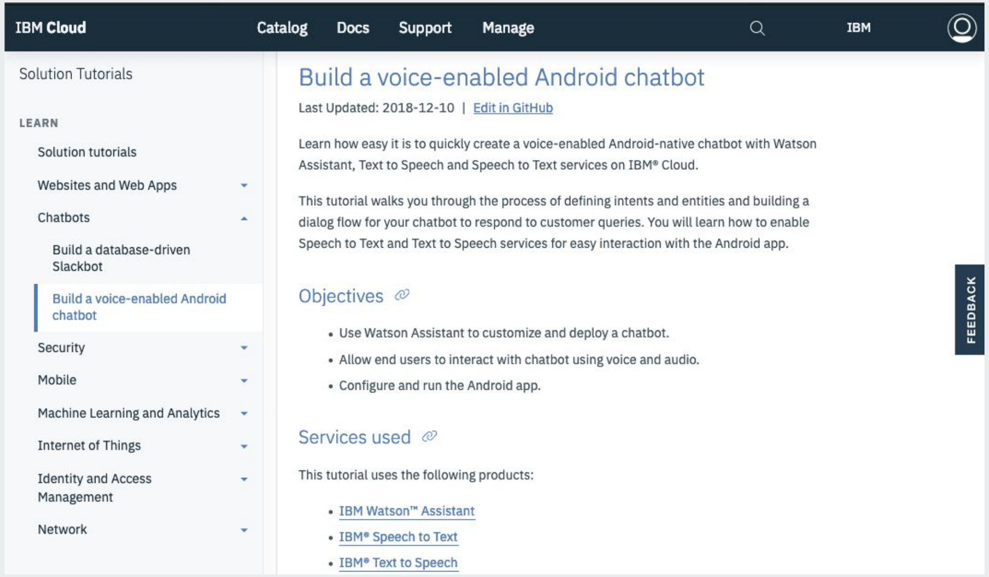
- Get started quickly and easily. Get faster time to value, and integrate across channels, networks and environments.
- Utilize Watson Assistant's reliable infrastructure that scales with individual use cases. Platform support from IBM gives you the backing you need.
- Own your data. IBM protects your privacy, allowing you to opt out of data sharing. Built on IBM Cloud and featuring reliable tooling with industry-leading security.

Integrating Watson on Android

1. (Optional) Use the Sample!
2. Import Watson dependencies.
3. Request manifest permissions.
4. Link to IBM Cloud Services
5. Use watson speech to text to record and transcribe audio.
6. Use watson text to speech to talk to the user.
7. Use watson assistant to interact with the user.

Watson Android Sample App

- Premade example using Assistant, TTS and STT
- Detailed step by step instructions available at: bluemix.net/docs/tutorials/android-watson-chatbot.html#build-a-voice-enabled-android-chatbot



Build a voice-enabled Android chatbot

Last Updated: 2018-12-10 | Edit in GitHub

Learn how easy it is to quickly create a voice-enabled Android-native chatbot with Watson Assistant, Text to Speech and Speech to Text services on IBM® Cloud.

This tutorial walks you through the process of defining intents and entities and building a dialog flow for your chatbot to respond to customer queries. You will learn how to enable Speech to Text and Text to Speech services for easy interaction with the Android app.

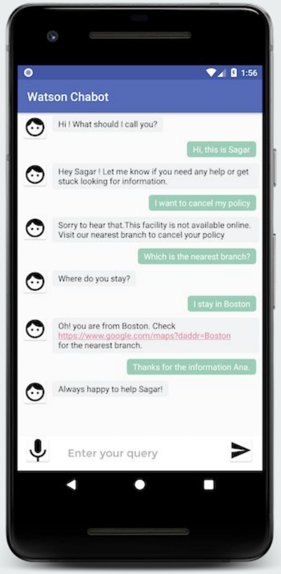
Objectives

- Use Watson Assistant to customize and deploy a chatbot.
- Allow end users to interact with chatbot using voice and audio.
- Configure and run the Android app.

Services used

This tutorial uses the following products:

- IBM Watson™ Assistant
- IBM® Speech to Text
- IBM® Text to Speech



Hi ! What should I call you?

Hi, this is Sagar

Hey Sagar ! Let me know if you need any help or get stuck looking for information.

I want to cancel my policy

Sorry to hear that.This facility is not available online. Visit our nearest branch to cancel your policy

Which is the nearest branch?

Where do you stay?

I stay in Boston

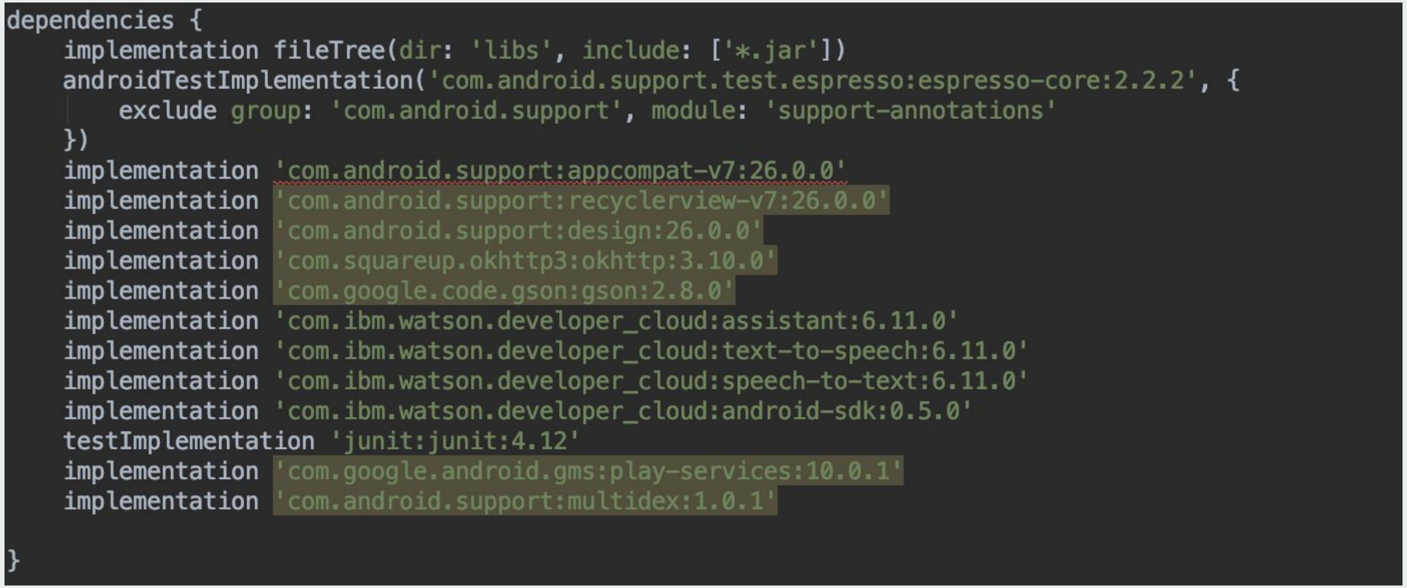
Oh! you are from Boston. Check https://www.google.com/maps?daddr=Boston for the nearest branch.

Thanks for the information Ana.

Always happy to help Sagar!

Importing Watson

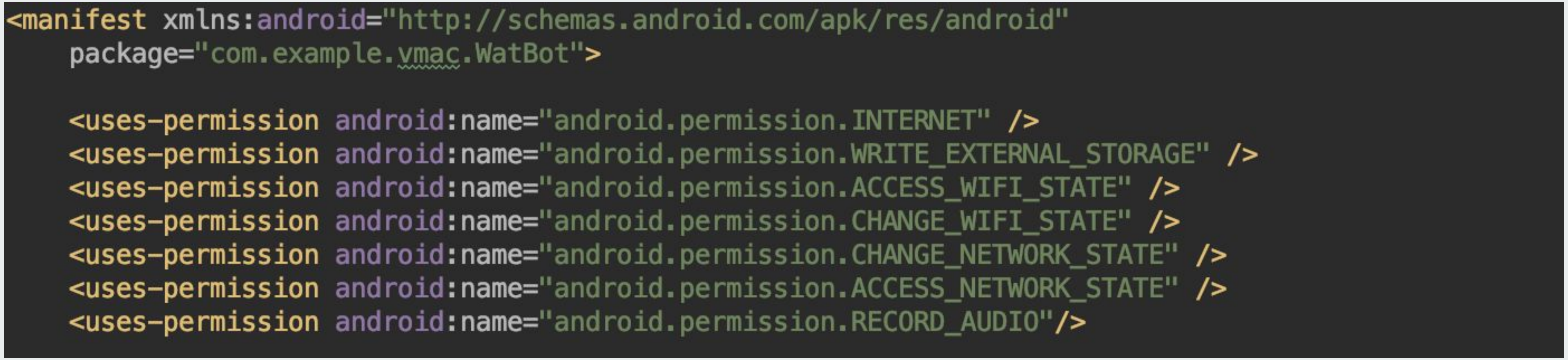
- Allows use of the watson developer sdk, and useful libraries for Watson Assistant, SST and TTS.



```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    androidTestImplementation('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    implementation 'com.android.support:appcompat-v7:26.0.0'
    implementation 'com.android.support:recyclerview-v7:26.0.0'
    implementation 'com.android.support:design:26.0.0'
    implementation 'com.squareup.okhttp3:okhttp:3.10.0'
    implementation 'com.google.code.gson:gson:2.8.0'
    implementation 'com.ibm.watson.developer_cloud:assistant:6.11.0'
    implementation 'com.ibm.watson.developer_cloud:text-to-speech:6.11.0'
    implementation 'com.ibm.watson.developer_cloud:speech-to-text:6.11.0'
    implementation 'com.ibm.watson.developer_cloud:android-sdk:0.5.0'
    testImplementation 'junit:junit:4.12'
    implementation 'com.google.android.gms:play-services:10.0.1'
    implementation 'com.android.support:multidex:1.0.1'

}
```

Request Manifest Permissions

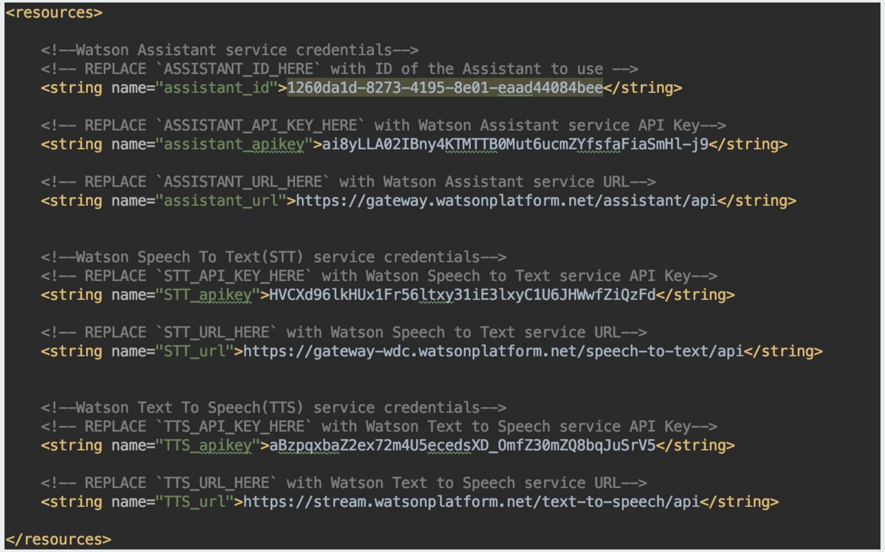
- The app needs access to Internet and Network to talk to the IBM Cloud.
- External Storage is needed to hold recordings.
- Record Audio is needed for Speech to Text to record the user.



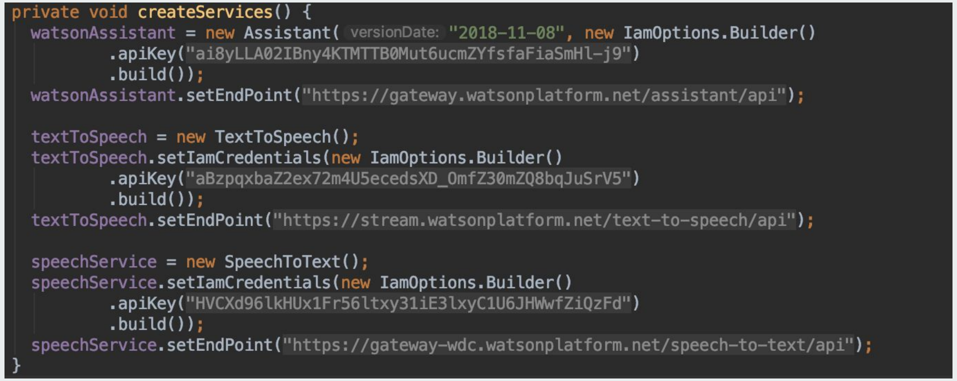
```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.vmac.WatBot">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
    <uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.RECORD_AUDIO"/>
```

Link to IBM Cloud

- Create a config.xml file.
- Run createServices() in onCreate()



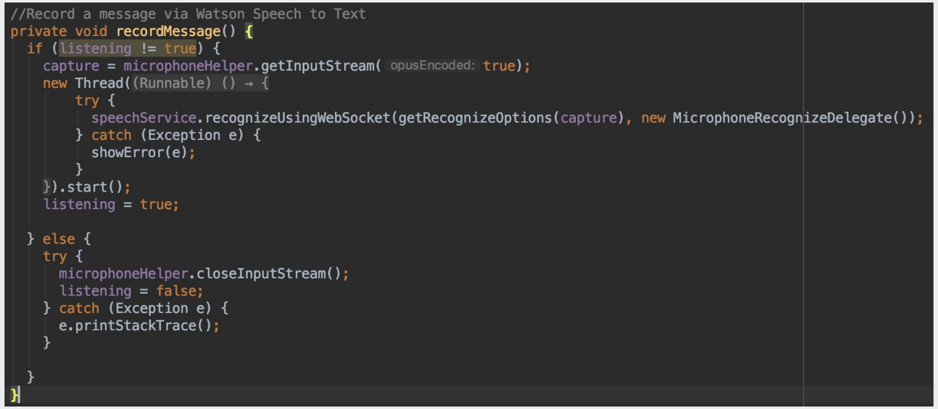
```xml
<resources>

    <!--Watson Assistant service credentials-->
    <!-- REPLACE `ASSISTANT_ID_HERE` with ID of the Assistant to use -->
    <string name="assistant_id">1260da1d-8273-4195-8e01-eaad44084bee</string>

    <!-- REPLACE `ASSISTANT_API_KEY_HERE` with Watson Assistant service API Key-->
    <string name="assistant_apikey">ai8yLLA02IBny4KTMTTB0Mut6ucmZYfsfaFiaSmHl-j9</string>

    <!-- REPLACE `ASSISTANT_URL_HERE` with Watson Assistant service URL-->
    <string name="assistant_url">https://gateway.watsonplatform.net/assistant/api</string>


    <!--Watson Speech To Text(STT) service credentials-->
    <!-- REPLACE `STT_API_KEY_HERE` with Watson Speech to Text service API Key-->
    <string name="STT_apikey">HVCXd96lkHUx1Fr56ltxy31iE3lxyC1U6JHWwfZiQzFd</string>

    <!-- REPLACE `STT_URL_HERE` with Watson Speech to Text service URL-->
    <string name="STT_url">https://gateway-wdc.watsonplatform.net/speech-to-text/api</string>


    <!--Watson Text To Speech(TTS) service credentials-->
    <!-- REPLACE `TTS_API_KEY_HERE` with Watson Text to Speech service API Key-->
    <string name="TTS_apikey">aBzpqxbaZ2ex72m4U5ecedsXD_OmfZ30mZQ8bqJuSrV5</string>

    <!-- REPLACE `TTS_URL_HERE` with Watson Text to Speech service URL-->
    <string name="TTS_url">https://stream.watsonplatform.net/text-to-speech/api</string>

</resources>
```



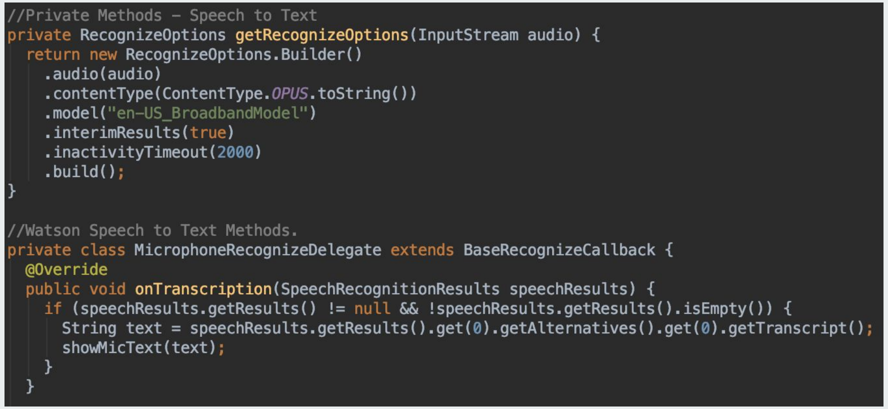
```java
private void createServices() {
  watsonAssistant = new Assistant( versionDate: "2018-11-08", new IamOptions.Builder()
          .apiKey("ai8yLLA02IBny4KTMTTB0Mut6ucmZYfsfaFiaSmHl-j9")
          .build());
  watsonAssistant.setEndPoint("https://gateway.watsonplatform.net/assistant/api");

  textToSpeech = new TextToSpeech();
  textToSpeech.setIamCredentials(new IamOptions.Builder()
          .apiKey("aBzpqxbaZ2ex72m4U5ecedsXD_OmfZ30mZQ8bqJuSrV5")
          .build());
  textToSpeech.setEndPoint("https://stream.watsonplatform.net/text-to-speech/api");

  speechService = new SpeechToText();
  speechService.setIamCredentials(new IamOptions.Builder()
          .apiKey("HVCXd96lkHUx1Fr56ltxy31iE3lxyC1U6JHWwfZiQzFd")
          .build());
  speechService.setEndPoint("https://gateway-wdc.watsonplatform.net/speech-to-text/api");
}
```

Using Watson STT

- Record Audio using the microphoneHelper.
- Send the audio to the IBM Cloud using speechService.
- Use the onTranscription hook to parse the results.
- Use RecognizeOptions to change any settings.



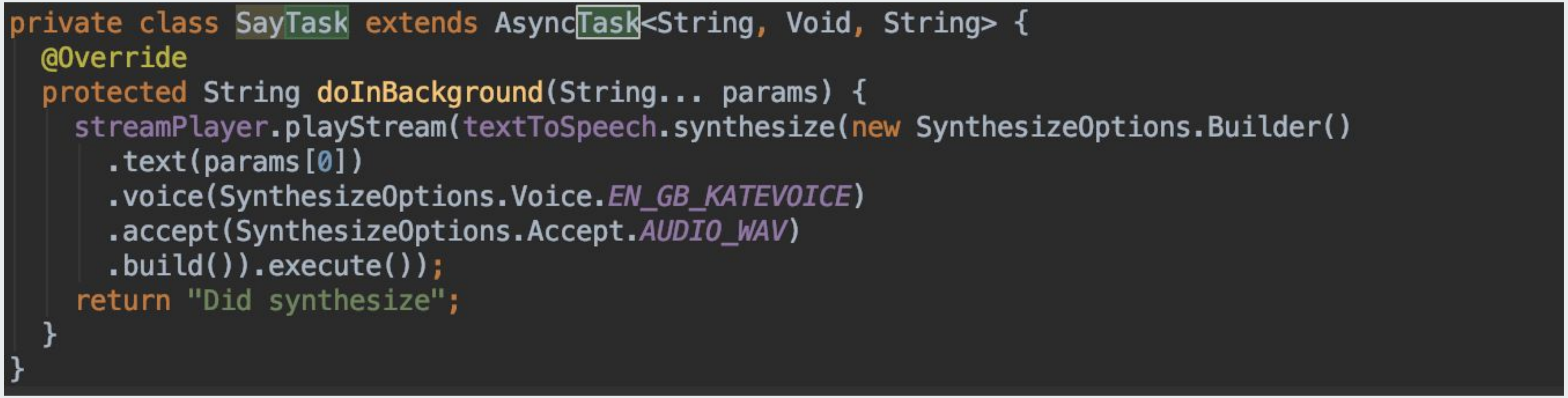
```
//Record a message via Watson Speech to Text
private void recordMessage() {
  if (listening != true) {
    capture = microphoneHelper.getInputStream( opusEncoded: true);
    new Thread((Runnable) () → {
        try {
          speechService.recognizeUsingWebSocket(getRecognizeOptions(capture), new MicrophoneRecognizeDelegate());
        } catch (Exception e) {
          showError(e);
        }
    }).start();
    listening = true;

  } else {
    try {
      microphoneHelper.closeInputStream();
      listening = false;
    } catch (Exception e) {
      e.printStackTrace();
    }

  }
}
```



```
//Private Methods - Speech to Text
private RecognizeOptions getRecognizeOptions(InputStream audio) {
  return new RecognizeOptions.Builder()
    .audio(audio)
    .contentType(ContentType.OPUS.toString())
    .model("en-US_BroadbandModel")
    .interimResults(true)
    .inactivityTimeout(2000)
    .build();
}

//Watson Speech to Text Methods.
private class MicrophoneRecognizeDelegate extends BaseRecognizeCallback {
  @Override
  public void onTranscription(SpeechRecognitionResults speechResults) {
    if (speechResults.getResults() != null && !speechResults.getResults().isEmpty()) {
      String text = speechResults.getResults().get(0).getAlternatives().get(0).getTranscript();
      showMicText(text);
    }
  }
```

Using Watson TTS

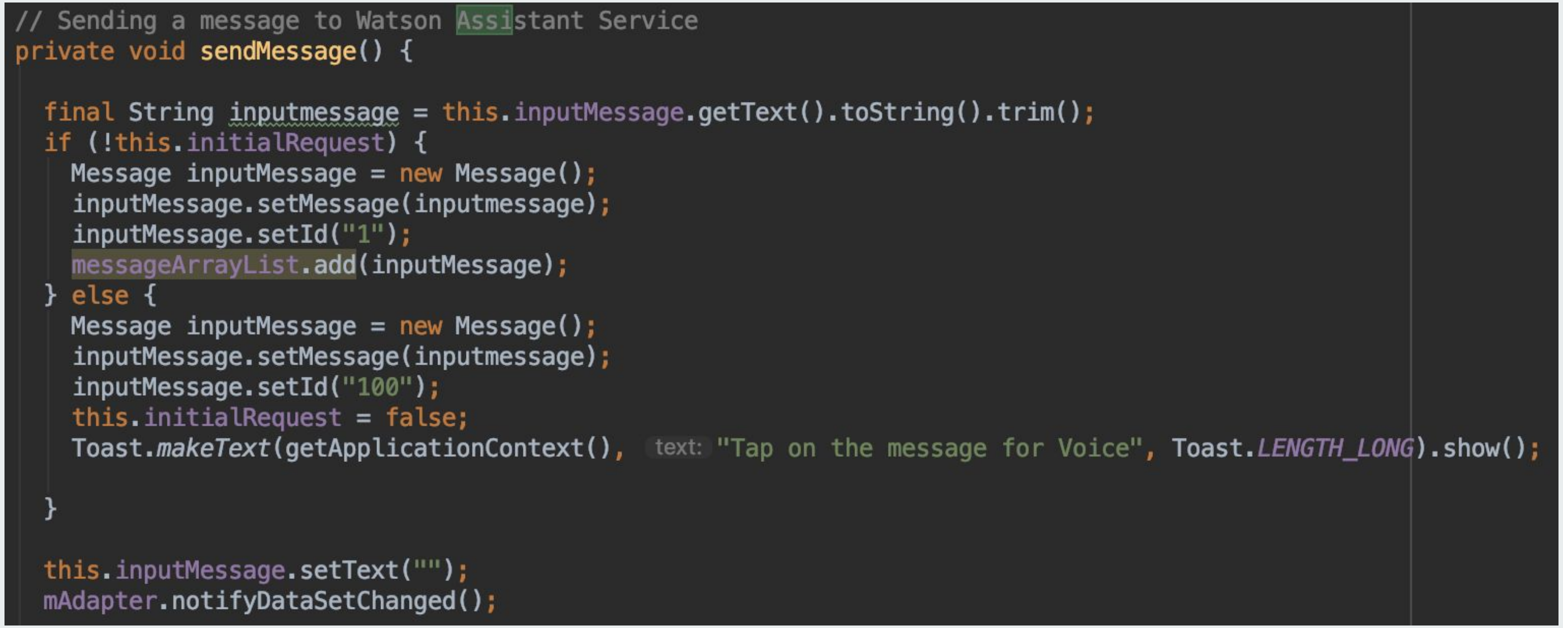
- Use textToSpeech.synthesize on your string.
- Use SynthesizeOptions.Builder() to change the voice or output.



```
private class SayTask extends AsyncTask<String, Void, String> {
  @Override
  protected String doInBackground(String... params) {
    streamPlayer.playStream(textToSpeech.synthesize(new SynthesizeOptions.Builder()
      .text(params[0])
      .voice(SynthesizeOptions.Voice.EN_GB_KATEVOICE)
      .accept(SynthesizeOptions.Accept.AUDIO_WAV)
      .build()).execute());
    return "Did synthesize";
  }
}
```

Using Watson Assistant 1

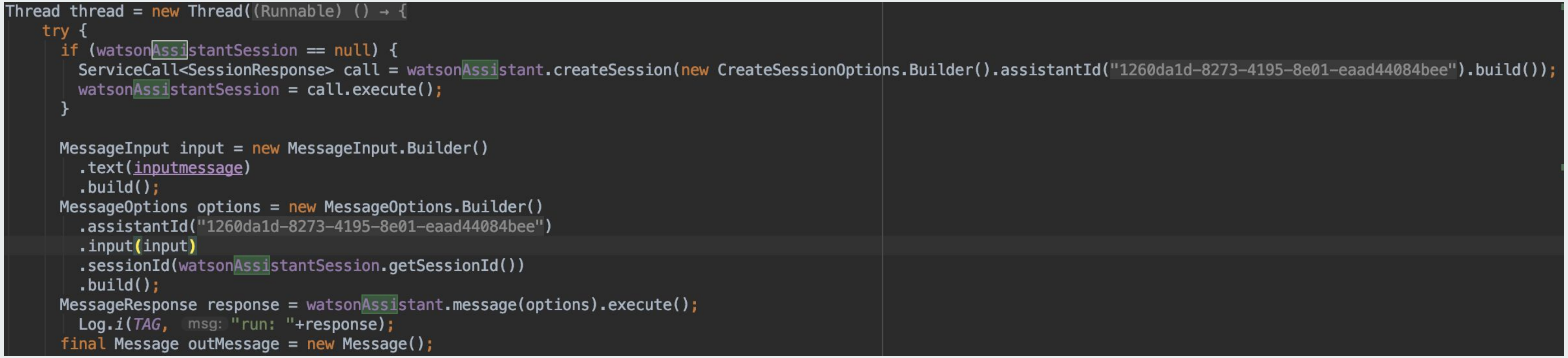
- Use the message class to format the user input.



```java
// Sending a message to Watson Assistant Service
private void sendMessage() {

  final String inputmessage = this.inputMessage.getText().toString().trim();
  if (!this.initialRequest) {
    Message inputMessage = new Message();
    inputMessage.setMessage(inputmessage);
    inputMessage.setId("1");
    messageArrayList.add(inputMessage);
  } else {
    Message inputMessage = new Message();
    inputMessage.setMessage(inputmessage);
    inputMessage.setId("100");
    this.initialRequest = false;
    Toast.makeText(getApplicationContext(),  text: "Tap on the message for Voice", Toast.LENGTH_LONG).show();

  }

  this.inputMessage.setText("");
  mAdapter.notifyDataSetChanged();
```

Using Watson Assistant 2

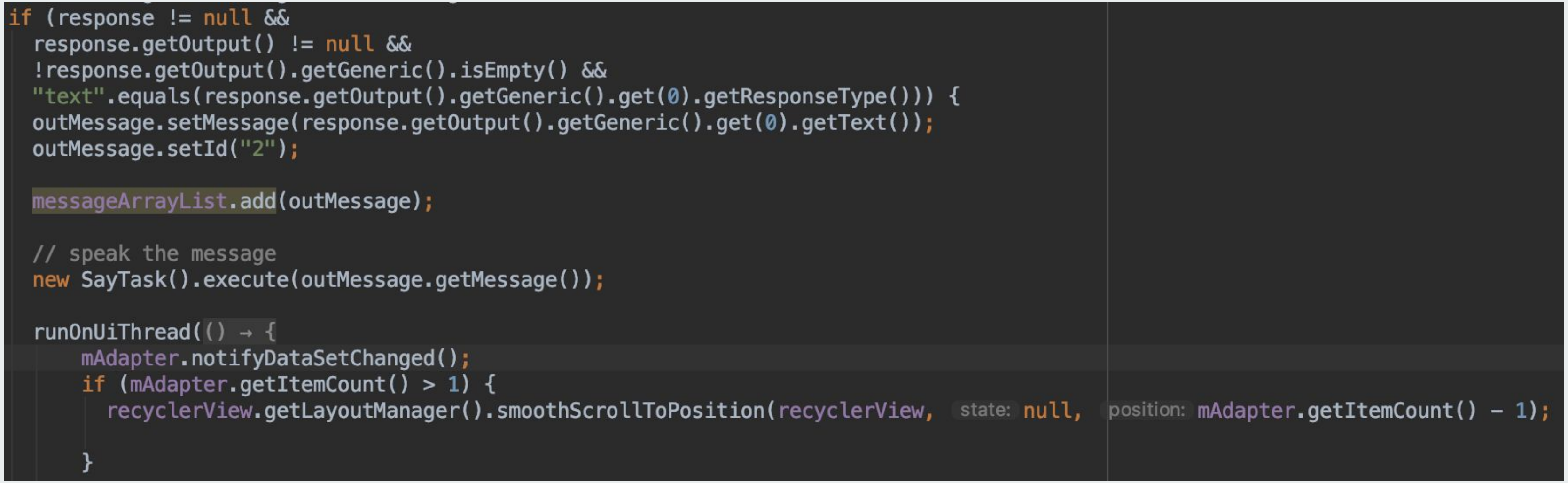
- Create a thread to handle requests.
- Create a new Assistant session.
- Send the inputMessage to be processed in the cloud.



```java
Thread thread = new Thread((Runnable) () → {
    try {
      if (watsonAssistantSession == null) {
        ServiceCall<SessionResponse> call = watsonAssistant.createSession(new CreateSessionOptions.Builder().assistantId("1260da1d-8273-4195-8e01-eaad44084bee").build());
        watsonAssistantSession = call.execute();
      }

      MessageInput input = new MessageInput.Builder()
        .text(inputmessage)
        .build();
      MessageOptions options = new MessageOptions.Builder()
        .assistantId("1260da1d-8273-4195-8e01-eaad44084bee")
        .input(input)
        .sessionId(watsonAssistantSession.getSessionId())
        .build();
      MessageResponse response = watsonAssistant.message(options).execute();
        Log.i(TAG,  msg: "run: "+response);
      final Message outMessage = new Message();
```

Using Watson Assistant 3

- If the response isn't null or empty we want to print it to the user.
- Send our response message to TTS.
- Make sure our app scrolls if there are too many messages.



```
if (response != null &&
  response.getOutput() != null &&
  !response.getOutput().getGeneric().isEmpty() &&
  "text".equals(response.getOutput().getGeneric().get(0).getResponseType())) {
  outMessage.setMessage(response.getOutput().getGeneric().get(0).getText());
  outMessage.setId("2");

  messageArrayList.add(outMessage);

  // speak the message
  new SayTask().execute(outMessage.getMessage());

  runOnUiThread(() → {
      mAdapter.notifyDataSetChanged();
      if (mAdapter.getItemCount() > 1) {
        recyclerView.getLayoutManager().smoothScrollToPosition(recyclerView, state: null, position: mAdapter.getItemCount() - 1);

      }
```