AI & Machine Learning Products (https://cloud.google.com/products/machine-learning/)
Cloud Text-to-Speech (https://cloud.google.com/text-to-speech/)
Documentation (https://cloud.google.com/text-to-speech/docs/) Guides

Speaking addresses with SSML

This tutorial demonstrates how to use Speech Synthesis Markup Language (SSML) (https://cloud.google.com/text-to-speech/docs/ssml) to speak a text file of addresses. You can mark up a string of text with SSML tags to personalize synthetic audio from Text-to-Speech.

Plaintext	SSML rendering of plaintext
`123 Street Ln`	`<speak>123 Street Ln</speak>`
`1 Number St`	`<speak>1 Number St</speak>`
`1 Piazza del Fibonacci`	`<speak>1 Piazza del Fibonacci</speak>`

Objective

Send a synthetic speech request to Text-to-Speech using SSML and Text-to-Speech client libraries (https://cloud.google.com/text-to-speech/docs/reference/libraries).

Costs

Refer to the Text-to-Speech pricing page (https://cloud.google.com/text-to-speech/pricing/) for cost information.

Before you begin

- Make sure that you have a Text-to-Speech project (https://cloud.google.com/text-to-speech/docs/reference/libraries) in Google Cloud Console (https://console.cloud.google.com/).
- This tutorial allows you to use Java, Node.js, or Python. If you plan to use Java, download (https://maven.apache.org/download.cgi) and install (https://maven.apache.org/install.html)

Maven. If you plan to use Node.js, download npm (https://www.npmjs.com/get-npm).

Downloading the code samples

To download the code samples, clone the Google Cloud GitHub samples for the programming language that you intend to use.

JAVA NODE.JS PYTHON

This tutorial uses code in the `texttospeech/cloud-client/src/main/java/com/example/texttospeech/` directory of the Google Cloud Platform Java samples (https://github.com/GoogleCloudPlatform/java-docs-samples).

To download and navigate to the code for this tutorial, run the following commands from the terminal.

```
git clone https://github.com/GoogleCloudPlatform/java-docs-samples.git
cd java-docs-samples/texttospeech/cloud-client/src/main/java/com/example/texttospee
```

Installing the client library

This tutorial uses the Text-to-Speech client library (https://cloud.google.com/text-to-speech/docs/reference/libraries).

JAVA NODE.JS PYTHON

This tutorial uses the following dependencies.

texttospeech/cloud-client/pom.xml (https://github.com/GoogleCloudPlatform/java-docs-samples/blob/master/texttospeech/cloud-client/pom.xml)

ECLOUDPLATFORM/JAVA-DOCS-SAMPLES/BLOB/MASTER/TEXTTOSPEECH/CLOUD-CLIENT/POM.XML)

```
<dependency>
  <groupId>com.google.cloud</groupId>
  <artifactId>google-cloud-texttospeech</artifactId>
  <version>0.117.0-beta</version>
</dependency>
```

Setting up your Google Cloud Platform credentials

Provide authentication credentials to your application code by setting the environment variable ***GOOGLE_APPLICATION_CREDENTIALS***. Replace ***[PATH]*** with the file path of the JSON file that contains your service account key, and ***[FILE_NAME]*** with the filename. This variable only applies to your current shell session, so if you open a new session, set the variable again.

LINUX OR MACOS WINDOWS

```
export GOOGLE_APPLICATION_CREDENTIALS="[PATH]"
```

For example:

```
export GOOGLE_APPLICATION_CREDENTIALS="/home/user/Downloads/[FILE_NAME].json"
```

Importing libraries

This tutorial uses the following system and client libraries.

JAVA NODE.JS PYTHON

TOSPEECH/CLOUD-CLIENT/SRC/MAIN/JAVA/COM/EXAMPLE/TEXTTOSPEECH/SSMLADDRESSES.JAVA)

FEEDBACK (#)

```
// Imports the Google Cloud client library
import com.google.cloud.texttospeech.v1.AudioConfig;
import com.google.cloud.texttospeech.v1.AudioEncoding;
import com.google.cloud.texttospeech.v1.SsmlVoiceGender;
import com.google.cloud.texttospeech.v1.SynthesisInput;
import com.google.cloud.texttospeech.v1.SynthesizeSpeechResponse;
import com.google.cloud.texttospeech.v1.TextToSpeechClient;
import com.google.cloud.texttospeech.v1.VoiceSelectionParams;
import com.google.common.html.HtmlEscapers;
import com.google.protobuf.ByteString;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.nio.file.Files;
import java.nio.file.Paths;
```

Using Text-to-Speech API

The following function takes a string of text tagged with SSML and the name of an MP3 file. The function uses the text tagged with SSML to generate synthetic audio. The function saves the synthetic audio to the MP3 filename designated as a parameter.

Note: This function overrides any pre-existing files with the same name as the ***outfile*** parameter. Ensure that you do not lose any pre-existing local files by using a unique filename as your ***outfile***.

JAVA NODE.JS PYTHON

TOSPEECH/CLOUD-CLIENT/SRC/MAIN/JAVA/COM/EXAMPLE/TEXTTOSPEECH/SSMLADDRESSES.JAVA)

FEEDBACK (#)

```
/**
 * Generates synthetic audio from a String of SSML text.
 *
 * <p>Given a string of SSML text and an output file name, this function calls the
 * API. The API returns a synthetic audio version of the text, formatted according
 * commands. This function saves the synthetic audio to the designated output file
 *
 * @param ssmlText String of tagged SSML text
 * @param outFile String name of file under which to save audio output
 * @throws Exception on errors while closing the client
 */
public static void ssmlToAudio(String ssmlText, String outFile) throws Exception {
  // Instantiates a client
  try (TextToSpeechClient textToSpeechClient = TextToSpeechClient.create()) {
    // Set the ssml text input to synthesize
    SynthesisInput input = SynthesisInput.newBuilder().setSsml(ssmlText).build();

    // Build the voice request, select the language code ("en-US") and
    // the ssml voice gender ("male")
    VoiceSelectionParams voice =
        VoiceSelectionParams.newBuilder()
            .setLanguageCode("en-US")
            .setSsmlGender(SsmlVoiceGender.MALE)
            .build();

    // Select the audio file type
    AudioConfig audioConfig =
        AudioConfig.newBuilder().setAudioEncoding(AudioEncoding.MP3).build();

    // Perform the text-to-speech request on the text input with the selected voic
    // audio file type
    SynthesizeSpeechResponse response =
        textToSpeechClient.synthesizeSpeech(input, voice, audioConfig);

    // Get the audio contents from the response
    ByteString audioContents = response.getAudioContent();

    // Write the response to the output file
    try (OutputStream out = new FileOutputStream(outFile)) {
      out.write(audioContents.toByteArray());
      System.out.println("Audio content written to file " + outFile);
    }
  }
}
```

Personalizing synthetic audio

The following function takes in the name of a text file and converts the contents of the file into a string of text tagged with SSML.

JAVA NODE.JS PYTHON

TOSPEECH/CLOUD-CLIENT/SRC/MAIN/JAVA/COM/EXAMPLE/TEXTTOSPEECH/SSMLADDRESSES.JAVA)

FEEDBACK (#)

```
/**
 * Generates SSML text from plaintext.
 *
 * <p>Given an input filename, this function converts the contents of the input te
 * String of tagged SSML text. This function formats the SSML String so that, when
 * the synthetic audio will pause for two seconds between each line of the text fi
 * function also handles special text characters which might interfere with SSML c
 *
 * @param inputFile String name of plaintext file
 * @return a String of SSML text based on plaintext input.
 * @throws IOException on files that don't exist
 */
public static String textToSsml(String inputFile) throws Exception {

  // Read lines of input file
  String rawLines = new String(Files.readAllBytes(Paths.get(inputFile)));

  // Replace special characters with HTML Ampersand Character Codes
  // These codes prevent the API from confusing text with SSML tags
  // For example, '<' --> '&lt;' and '&' --> '&amp;'
  String escapedLines = HtmlEscapers.htmlEscaper().escape(rawLines);

  // Convert plaintext to SSML
  // Tag SSML so that there is a 2 second pause between each address
  String expandedNewline = escapedLines.replaceAll("\\n", "\n<break time='2s'/>");
  String ssml = "<speak>" + expandedNewline + "</speak>";

  // Return the concatenated String of SSML
  return ssml;
}
```

Putting it all together

This program uses the following input.

```
123 Street Ln, Small Town, IL 12345 USA
1 Jenny St & Number St, Tutone City, CA 86753
1 Piazza del Fibonacci, 12358 Pisa, Italy
```

Passing the above text to `text_to_ssml()` generates the following tagged text.

```
<speak>123 Street Ln, Small Town, IL 12345 USA
<break time="2s"/>1 Jenny St &amp; Number St, Tutone City, CA 86753
<break time="2s"/>1 Piazza del Fibonacci, 12358 Pisa, Italy
<break time="2s"/></speak>
```

Running the code

To generate an audio file of synthetic speech, run the following code from the command line.

JAVA NODE.JS PYTHON

LINUX OR MACOS WINDOWS

From the `java-docs-samples/texttospeech/cloud-client/` directory, execute the following command on the command line.

```
$ mvn clean package
```

Checking your output

This program outputs an ***example.mp3*** audio file of synthetic speech.

JAVA NODE.JS PYTHON

Navigate into `java-docs-samples/texttospeech/cloud-client/resources/` directory.

Check the `resources` directory for an ***example.mp3*** file.

Listen to the following audio clip to check that your ***example.mp3*** file sounds the same.

▶ 0:00 / 0:21

Troubleshooting

- Forgetting to set the ***GOOGLE_APPLICATION_CREDENTIALS*** (#setting_up_your_google_cloud_platform_credentials) environment variable on the command line generates the error message:

  ```
  The Application Default Credentials are not available.
  ```

- Passing `text_to_ssml()` the name of a non-existent file generates the error message:

  ```
  IOError: [Errno 2] No such file or directory
  ```

- Passing `ssml_to_audio()` a ***ssml_text*** parameter which contains ***None*** generates the error message:

  ```
  InvalidArgument: 400 Invalid input type. Type has to be text or SSML
  ```

- Make sure that you are running the code from the correct directory (#running_the_code).

What's next

- Explore other SSML tags (https://cloud.google.com/text-to-speech/docs/ssml).
- Learn how to provide speech translation features to your Android app (https://cloud.google.com/solutions/mobile/speech-translation-android-microservice)
- Learn how to use SSML with Translation and Vision (https://cloud.google.com/translate/docs/hybrid-glossaries-tutorial)

Cleaning up

To avoid incurring charges to your Google Cloud Platform account for the resources used in this tutorial, use the Google Cloud Console (https://console.cloud.google.com/) to delete your project if you do not need it.

Deleting your project

1. In the Cloud Console (https://console.cloud.google.com/), go to the Projects page.

2. In the project list, select the project you want to delete and click **Delete**.
3. In the dialog box, type the project ID, and click **Shut down** to delete the project.

Last updated December 4, 2019.