Exercise – Convert speech from an audio file to text

100 XP

3 minutes

Choose your development language

C# Python

In this exercise, you will make use of the Speech Service you created earlier. This exercise requires configuration on your local computer for a development environment as we will use Visual Studio Code as the editor. If you running in a hosted environment, most of the software should already be installed for you.

Environment setup

The environment for this exercise will make use of Visual Studio Code as the editor. Depending on the programming language you choose, the setup will differ. Follow the steps outlined here to configure your local computer for completion of the exercises.

1. Install Visual Studio Code for your operating system.

Python

1. If you will be completing your coding with Python, ensure you have a Python environment installed locally.
2. Once you have Python installed, you will need to install the extension for VS Code.

C#

1. If you will be using C# as your code language, start by installing the latest .NET Core package for your platform. You can choose Windows, Linux, or macOS from the drop-down on this page.
2. Once you have .NET Core installed, you will need to add the C# Extension to VS Code. Select the **Extensions** option in the left nav pane, or press **CTRL+SHIFT+X** and enter C# in the search dialog.
3. You will also make use of a recorded .wav file in this exercise. Download the narration.wav file by cloning this repo to your local computer. Once you have cloned the repo, either place the narration.wav file in a location where your code will be executed, or update the path in the code to match the location of the file.

With your environment setup, you are now ready to begin the coding exercise.

Exercise - convert speech input to text

1. Start by opening Visual Studio Code.
2. Create a new folder with a name of your choosing, example **SpeechToTextPython**.
3. Create a new file in the folder called **speechtotext.py**.
4. If you get any messages about installing additional extensions or components, choose to do so.
5. Select the **speechtotext.py** file in the explorer pane.
6. The first item we need to focus on is creating a **speech_config** object. The **speech_config** object is based on the **SpeechConfig** class and will include information related to your subscription such as the key, region, endpoint, etc. For our purpose here, we will only be concerned with the **subscription key** and **region**.
7. In the code window, inside **speechtotext.py**, enter the following code to create two variables and a **speech_config** object. The first line imports the necessary package for the speech sdk.

```python
import azure.cognitiveservices.speech as speechsdk
speech_key, service_region = "YourSubscriptionKey", "YourServiceRegion"
speech_config = speechsdk.SpeechConfig(subscription=speech_key, region=service_region)
```

8. Locate the **YourSubscriptionKey**, **YourServiceRegion** line in the file and paste your key into the proper spot and set the region to where you created your **Speech Service**, such as **westus**.
9. Once you have the **speech_config** object created, the next step is to initialize a **recognizer**. The **recognizer** is based on the **SpeechRecognizer** class. You will pass in the **speech_config** object into the **recognizer** as the credentials required to connect to and validate your request, with the Speech service.
10. The **SpeechRecognizer** can accept input from a microphone or from a file. This exercise will use a file as the input so you will need to add an **AudioConfig** component to the application. Enter the following code to enable reading audio from a file.

> **Note**
>
> The **narration.wav** entry in the filename variable will need to be modified based on where you stored the file.

```python
audio_input = speechsdk.AudioConfig(filename="/narration.wav")
```

11. Enter the following code to initialize the **speech_recognizer** object with the **speech_config** and **audio_config** information.

```python
speech_recognizer = speechsdk.SpeechRecognizer(speech_config=speech_config, audio_config=audio_i
```

12. Enter the following code to output a status message and to call the service, placing the converted text into a variable.

```python
print("Recognizing first result...")

result = speech_recognizer.recognize_once()
```

13. Finally, you will enter code to check the result of the service call. It should succeed but we also include some checks for potential errors that may occur. Enter the following lines of code to check the status and output the results accordingly.

```python
if result.reason == speechsdk.ResultReason.RecognizedSpeech:
    print("Recognized: {}".format(result.text))
elif result.reason == speechsdk.ResultReason.NoMatch:
    print("No speech could be recognized: {}".format(result.no_match_details))
elif result.reason == speechsdk.ResultReason.Canceled:
    cancellation_details = result.cancellation_details
    print("Speech Recognition canceled: {}".format(cancellation_details.reason))
    if cancellation_details.reason == speechsdk.CancellationReason.Error:
        print("Error details: {}".format(cancellation_details.error_details))
```

14. Your completed code should look similar to this.

```python
import azure.cognitiveservices.speech as speechsdk

speech_key, service_region = "YourSubscriptionKey", "YourServiceRegion"
speech_config = speechsdk.SpeechConfig(subscription=speech_key, region=service_region)

audio_filename = "/narration.wav"
audio_input = speechsdk.AudioConfig(filename=audio_filename)

speech_recognizer = speechsdk.SpeechRecognizer(speech_config=speech_config, audio_config=audio_i

print("Recognizing first result...")

result = speech_recognizer.recognize_once()
if result.reason == speechsdk.ResultReason.RecognizedSpeech:
    print("Recognized: {}".format(result.text))
elif result.reason == speechsdk.ResultReason.NoMatch:
    print("No speech could be recognized: {}".format(result.no_match_details))
elif result.reason == speechsdk.ResultReason.Canceled:
    cancellation_details = result.cancellation_details
    print("Speech Recognition canceled: {}".format(cancellation_details.reason))
    if cancellation_details.reason == speechsdk.CancellationReason.Error:
        print("Error details: {}".format(cancellation_details.error_details))
```

15. You will need to add a package for the Cognitive Services speech functions so open a terminal window in Visual Studio Code and paste the following command in the terminal window, and hit Enter to add the package: `pip install --upgrade azure-cognitiveservices-speech`
16. Locate the **Run Python File in Terminal** button in the upper right corner of VS Code Codespaces and select it to run the code.
17. The result of the speech to text transcription is displayed in the terminal window.
18. Feel free to record your own audio file and test with the service.

Next unit: Review supported languages

Continue

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue.