

Create speech-enabled apps with Azure AI services

The **Azure AI Speech service** enables you to build **speech-enabled applications**. This module focuses on using the **speech-to-text and text to speech APIs**, which enable you to create apps that are capable of **speech recognition** and **speech synthesis**.

Learning objectives

In this module, you'll learn how to:

- Provision an Azure resource for the Azure AI Speech service
- Implement *speech recognition* with the *Azure AI Speech to text API*
- Use the *Text to speech API* to implement *speech synthesis*
- Configure audio format and voices
- Use **Speech Synthesis Markup Language (SSML)**

API	Function
Speech to text	Speech recognition
Text to speech	Speech synthesis

Introduction

Azure AI Speech provides APIs that you can use to build **speech-enabled applications**. This includes:

- **Speech to text:** An API that enables **speech recognition** in which your application can accept **spoken input**.
- **Text to speech:** An API that enables **speech synthesis** in which your application can provide **spoken output**.
- **Speech Translation:** An API that you can use to **translate spoken input into multiple languages**.

- **Keyword Recognition:** An API that enables your application to **recognize keywords or short phrases**.
- **Intent Recognition:** An API that uses **conversational language understanding to determine the semantic meaning of spoken input**.

This module focuses on speech recognition and speech synthesis, which are core capabilities of any speech-enabled application.

Note: The code examples in this module are provided in Python, but you can use any of the available Azure AI Speech SDK packages to develop speech-enabled applications in your preferred language. Available SDK packages include:

- [azure-cognitiveservices-speech for Python](#)
- [Microsoft.CognitiveServices.Speech for Microsoft .NET](#)
- [microsoft-cognitiveservices-speech-sdk for JavaScript](#)
- [Microsoft Cognitive Services Speech SDK For Java](#)

Provision an Azure resource for speech

Before you can use Azure AI Speech, you need to create an Azure AI Speech resource in your Azure subscription. You can use either **a dedicated Azure AI Speech resource** or **a multi-service Azure AI Services or Azure AI Foundry resource**.

After you create your resource, you'll need the following information to use it from a client application through one of the supported SDKs:

- The location in which the resource is deployed (for example, eastus)
- One of the keys assigned to your resource.

You can view of these values on the **Keys** and **Endpoint** page for your resource in the Azure portal.

While the specific syntax and parameters can vary between language-specific SDKs, most interactions with the Azure AI Speech service start with the creation of a `SpeechConfig` object that encapsulates the connection to your Azure AI Speech resource.

For example, the following Python code instantiates a `SpeechConfig` object based on an Azure AI Speech resource in the East US region:

```
import azure.cognitiveservices.speech as speech_sdk
```

```
speech_config = speech_sdk.SpeechConfig(your_project_key, 'eastus')
```

Note: This example assumes that the Speech SDK package for python has been installed, like this:

```
pip install azure-cognitiveservices-speech
```

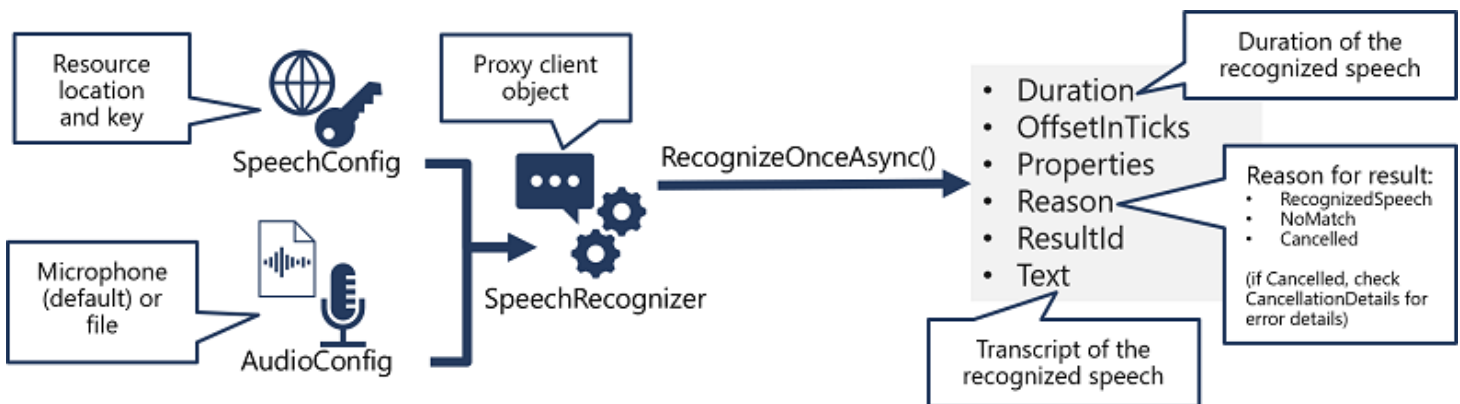
Use the Azure AI Speech to Text API

The Azure AI Speech service supports **speech recognition** through the following features:

- **Real-time transcription:** Instant transcription with intermediate results for live audio inputs.
- **Fast transcription:** Fastest synchronous output for situations with predictable latency.
- **Batch transcription:** Efficient processing for large volumes of prerecorded audio.
- **Custom speech:** Models with **enhanced accuracy for specific domains and conditions**.

Using the Azure AI Speech SDK

While the specific details vary, depending on the SDK being used (Python, C#, and so on); there's a consistent pattern for using the **Speech to text** API:



- Use a **SpeechConfig** object to encapsulate the information required to connect to your Azure AI Speech resource. Specifically, its **location** and **key**.
- Optionally, use an **AudioConfig** to define the input source for the audio to be transcribed. By default, this is the default system microphone, but you can also specify an audio file.
- Use the **SpeechConfig** and **AudioConfig** to create a **SpeechRecognizer** object. This object is a proxy client for the **Speech to text** API.

- Use the methods of the **SpeechRecognizer** object to call the underlying API functions. For example, the **RecognizeOnceAsync()** method uses the Azure AI Speech service to *asynchronously* transcribe a single spoken utterance.
- Process the response from the Azure AI Speech service. In the case of the **RecognizeOnceAsync()** method, the result is a **SpeechRecognitionResult** object that includes the following properties:
 - Duration
 - OffsetInTicks
 - Properties
 - Reason
 - ResultId
 - Text

If the operation was successful, the **Reason** property has the enumerated value **RecognizedSpeech**, and the Text property contains the transcription. Other possible values for **Result** include **NoMatch** (indicating that the audio was successfully parsed but no speech was recognized) or **Canceled**, indicating that an error occurred (in which case, you can check the Properties collection for the **CancellationReason** property to determine what went wrong).

Use the text to speech API

Similarly to its Speech to text APIs, the Azure AI Speech service offers other **REST APIs for speech synthesis**:

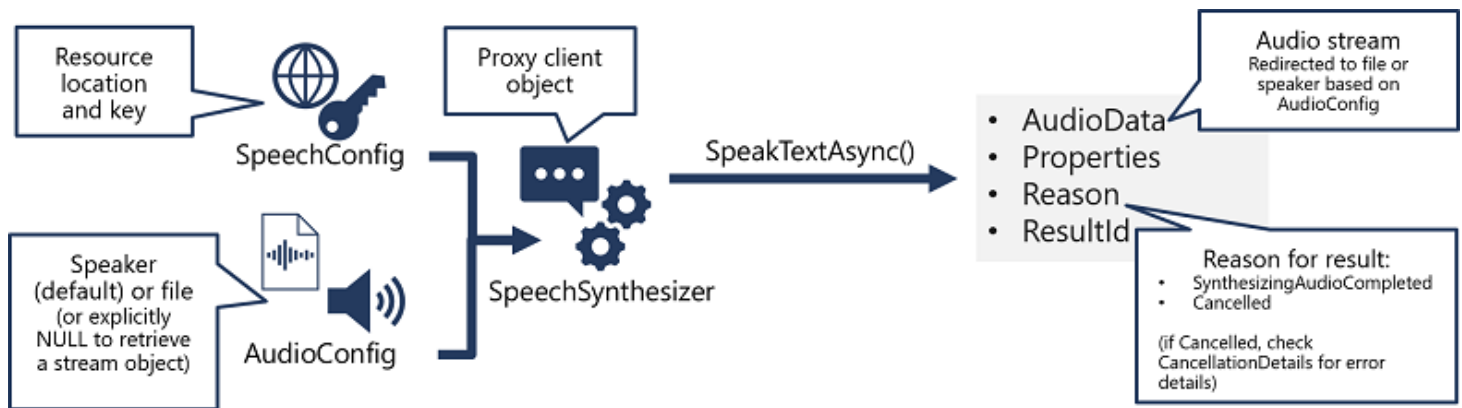
- The **Text to speech API**, which is the primary way to perform speech synthesis.
- The **Batch synthesis API**, which is designed to support batch operations that *convert large volumes of text to audio* - for example to *generate an audio-book from the source text*.

You can learn more about the REST APIs in the [Text to speech REST API documentation](#). In practice, most interactive speech-enabled applications use the Azure AI Speech service through a (programming) language-specific SDK.

Using the Azure AI Speech SDK

As with speech recognition, in practice most interactive speech-enabled applications are built using the Azure AI Speech SDK.

The pattern for implementing speech synthesis is similar to that of speech recognition:



- Use a `SpeechConfig` object to encapsulate the information required to connect to your Azure AI Speech resource. Specifically, its **location** and **key**.
- Optionally, use an `AudioConfig` to define the output device for the speech to be synthesized. By default, this is the default system speaker, but you can also specify an audio file, or by explicitly setting this value to a `null` value, you can process the audio stream object that is returned directly.
- Use the `SpeechConfig` and `AudioConfig` to create a `SpeechSynthesizer` object. This object is a proxy client for the **Text to speech API**.
- Use the methods of the `SpeechSynthesizer` object to call the underlying API functions. For example, the `SpeakTextAsync()` method uses the Azure AI Speech service to convert text to spoken audio.
- Process the response from the Azure AI Speech service. In the case of the `SpeakTextAsync` method, the result is a `SpeechSynthesisResult` object that contains the following properties:
 - `AudioData`
 - `Properties`
 - `Reason`
 - `ResultId`

When speech has been successfully synthesized, the `Reason` property is set to the `SynthesizingAudioCompleted` enumeration and the `AudioData` property contains the audio stream (which, depending on the `AudioConfig` may have been automatically sent to a speaker or file).

Configure audio format and voices

When synthesizing speech, you can use a `SpeechConfig` object to customize the audio that is returned by the Azure AI Speech service.

Audio format

The Azure AI Speech service supports multiple output formats for the audio stream that is generated by speech synthesis. Depending on your specific needs, you can choose a format based on the required:

- Audio file type
- Sample-rate
- Bit-depth

For example, the following Python code sets the speech output format for a previously defined `SpeechConfig` object named `speech_config` :

```
speech_config.set_speech_synthesis_output_format(SpeechSynthesisOutputFormat.Riff24Khz16BitMonoI
```

For a full list of supported formats and their enumeration values, see the [Azure AI Speech SDK documentation](#).

Voices

The Azure AI Speech service provides multiple voices that you can use to personalize your speech-enabled applications. Voices are **identified by names** that indicate a locale and a person's name - for example `en-GB-George` .

The following Python example code sets the voice to be used

```
speech_config.speech_synthesis_voice_name = "en-GB-George"
```

For information about voices, see the [Azure AI Speech SDK documentation](#).

Use Speech Synthesis Markup Language

While the Azure AI Speech SDK enables you to submit plain text to be synthesized into speech, the service also supports an XML-based syntax for describing characteristics of the speech you want to generate. This **Speech Synthesis Markup Language (SSML)** syntax offers greater control over how the spoken output sounds, enabling you to:

- Specify a **speaking style**, such as *"excited"* or *"cheerful"* when using a neural voice.

- Insert **pauses or silence**.
- Specify **phonemes** (phonetic pronunciations), for example to pronounce the text "SQL" as "sequel".
- Adjust the **prosody** of the voice (affecting the pitch, timbre, and speaking rate).
- Use **common "say-as" rules**, for example to specify that a given string should be expressed as a date, time, telephone number, or other form.
- Insert **recorded speech or audio**, for example to include a standard recorded message or simulate background noise.

For example, consider the following SSML:

```
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis"
        xmlns:mstts="https://www.w3.org/2001/mstts" xml:lang="en-US">
  <voice name="en-US-AriaNeural">
    <mstts:express-as style="cheerful">
      I say tomato
    </mstts:express-as>
  </voice>
  <voice name="en-US-GuyNeural">
    I say <phoneme alphabet="sapi" ph="t ao m ae t ow"> tomato </phoneme>.
    <break strength="weak"/> Lets call the whole thing off!
  </voice>
</speak>
```

This SSML specifies a spoken dialog between two different **neural voices**, like this:

- Ariana (cheerfully): "I say tomato:
- Guy: "I say tomato (pronounced tom-ah-toe) ... Let's call the whole thing off!"

To submit an SSML description to the Speech service, you can use an appropriate method of a `SpeechSynthesizer` object, like this:

```
speech_synthesizer.speak_ssml('<speak>...');
```

For more information about SSML, see the [Azure AI Speech SDK documentation](#).

Exercise - Create a speech-enabled app

In this exercise, build a speech enabled app for both speech recognition and synthesis.

Recognize and synthesize speech

Azure AI Speech is a service that provides speech-related functionality, including:

- A *speech-to-text API* that enables you to implement *speech recognition* (converting audible spoken words into text).
- A *text-to-speech API* that enables you to implement *speech synthesis* (converting text into audible speech).

In this exercise, you'll use both of these APIs to implement a speaking clock application.

Module assessment

1. What information do you need from your Azure AI Speech service resource to consume it using the Azure AI Speech SDK? **The location and one of the keys.**
NOTE: Location is not endpoint!! An example of location: eastus
2. Which object should you use to specify that the speech input to be transcribed to text is in an audio file? **AudioConfig** [Use the Azure AI Speech to Text API](#)
3. How can you change the voice used in speech synthesis? Set the `SpeechSynthesisVoiceName` property of the `speechConfig` object to the desired voice name.

Summary

In this module, you learned how to:

- Provision an Azure resource for the Azure AI Speech service
- Use the **Speech to text API** to implement **speech recognition**
- Use the **Text to speech API** to implement **speech synthesis**
- Configure audio format and voices
- Use **Speech Synthesis Markup Language (SSML)**

To learn more about the Azure AI Speech, refer to the [Azure AI Speech service documentation](#).