Quickstart: Recognize speech from a microphone

06/25/2020 • 54 minutes to read • +3

Choose a programming language or CLI

CLI	C#	C++	Go	Java	Python	JavaScript	More languages...

In this article

Prerequisites
Create a new Website folder
Unpack the Speech SDK for JavaScript into that folder
Create an index.html page
Create the token source (optional)
Build and run the sample locally
Build and run the sample via a web server
Next steps

In this quickstart, you use the Speech SDK to interactively recognize speech from a microphone input, and get the text transcription from captured audio. It's easy to integrate this feature into your apps or devices for common recognition tasks, such as transcribing conversations. It can also be used for more complex integrations, like using the Bot Framework with the Speech SDK to build voice assistants.

After satisfying a few prerequisites, recognizing speech from a microphone only takes four steps:

- ✓ Create a `SpeechConfig` object from your subscription key and region.
- ✓ Create a `SpeechRecognizer` object using the `SpeechConfig` object from above.
- ✓ Using the `SpeechRecognizer` object, start the recognition process for a single utterance.
- ✓ Inspect the `SpeechRecognitionResult` returned.

You can view or download all Speech SDK JavaScript Samples on GitHub.

Prerequisites

Before you get started:

- ✓ Create an Azure Speech resource
- ✓ Setup your development environment and create an empty project
- ✓ Make sure that you have access to a microphone for audio capture

Create a new Website folder

Create a new, empty folder. In case you want to host the sample on a web server, make sure that the web server can access the folder.

Unpack the Speech SDK for JavaScript into that folder

Download the Speech SDK as a .zip package and unpack it into the newly created folder. This results in two files being unpacked, `microsoft.cognitiveservices.speech.sdk.bundle.js` and `microsoft.cognitiveservices.speech.sdk.bundle.js.map`. The latter file is optional, and is useful for debugging into the SDK code.

Create an index.html page

Create a new file in the folder, named `index.html` and open this file with a text editor.

1. Create the following HTML skeleton:

HTML Copy

```html
<!DOCTYPE html>
<html>
<head>
  <title>Microsoft Cognitive Services Speech SDK JavaScript Quickstart</title>
  <meta charset="utf-8" />
</head>
<body style="font-family:'Helvetica Neue',Helvetica,Arial,sans-serif; font-size:13px;">
  <!-- <uidiv> -->
  <div id="warning">
    <h1 style="font-weight:500;">Speech Recognition Speech SDK not found (microsoft.cognitiveservices.speech.sdk.bundle.js missing).</h1>
  </div>

  <div id="content" style="display:none">
    <table width="100%">
      <tr>
```

```
        <td></td>
        <td><h1 style="font-weight:500;">Microsoft Cognitive Services Speech
SDK JavaScript Quickstart</h1></td>
      </tr>
      <tr>
        <td align="right"><a
href="https://docs.microsoft.com/azure/cognitive-services/speech-
service/get-started" target="_blank">Subscription</a>:</td>
        <td><input id="subscriptionKey" type="text" size="40"
value="subscription"></td>
      </tr>
      <tr>
        <td align="right">Region</td>
        <td><input id="serviceRegion" type="text" size="40"
value="YourServiceRegion"></td>
      </tr>
      <tr>
        <td></td>
        <td><button id="startRecognizeOnceAsyncButton">Start
recognition</button></td>
      </tr>
      <tr>
        <td align="right" valign="top">Results</td>
        <td><textarea id="phraseDiv" style="display: inline-
block;width:500px;height:200px"></textarea></td>
      </tr>
    </table>
  </div>
  <!-- </uidiv> -->

  <!-- <speechsdkref> -->
  <!-- Speech SDK reference sdk. -->
  <script src="microsoft.cognitiveservices.speech.sdk.bundle.js"></script>
  <!-- </speechsdkref> -->

  <!-- <authorizationfunction> -->
  <!-- Speech SDK Authorization token -->
  <script>
  // Note: Replace the URL with a valid endpoint to retrieve
  //       authorization tokens for your subscription.
  var authorizationEndpoint = "token.php";

  function RequestAuthorizationToken() {
    if (authorizationEndpoint) {
      var a = new XMLHttpRequest();
      a.open("GET", authorizationEndpoint);
      a.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");
      a.send("");
      a.onload = function() {
          var token = JSON.parse(atob(this.responseText.split(".")[1]));
          serviceRegion.value = token.region;
          authorizationToken = this.responseText;
          subscriptionKey.disabled = true;
          subscriptionKey.value = "using authorization token (hit F5 to
refresh)";
          console.log("Got an authorization token: " + token);
      }
    }
  }
  </script>
  <!-- </authorizationfunction> -->

  <!-- <quickstartcode> -->
  <!-- Speech SDK USAGE -->
  <script>
    // status fields and start button in UI
    var phraseDiv;
    var startRecognizeOnceAsyncButton;

    // subscription key and region for speech services.
    var subscriptionKey, serviceRegion;
    var authorizationToken;
    var SpeechSDK;
    var recognizer;

    document.addEventListener("DOMContentLoaded", function () {
      startRecognizeOnceAsyncButton =
document.getElementById("startRecognizeOnceAsyncButton");
      subscriptionKey = document.getElementById("subscriptionKey");
      serviceRegion = document.getElementById("serviceRegion");
      phraseDiv = document.getElementById("phraseDiv");

      startRecognizeOnceAsyncButton.addEventListener("click", function () {
        startRecognizeOnceAsyncButton.disabled = true;
        phraseDiv.innerHTML = "";

        // if we got an authorization token, use the token. Otherwise use
the provided subscription key
        var speechConfig;
        if (authorizationToken) {
          speechConfig =
SpeechSDK.SpeechConfig.fromAuthorizationToken(authorizationToken,
serviceRegion.value);
        } else {
          if (subscriptionKey.value === "" || subscriptionKey.value ===
"subscription") {
            alert("Please enter your Microsoft Cognitive Services Speech
subscription key!");
            return;
          }
          speechConfig =
SpeechSDK.SpeechConfig.fromSubscription(subscriptionKey.value,
serviceRegion.value);
        }

        speechConfig.speechRecognitionLanguage = "en-US";
        var audioConfig  =
SpeechSDK.AudioConfig.fromDefaultMicrophoneInput();
        recognizer = new SpeechSDK.SpeechRecognizer(speechConfig,
audioConfig);

        recognizer.recognizeOnceAsync(
          function (result) {
            startRecognizeOnceAsyncButton.disabled = false;
            phraseDiv.innerHTML += result.text;
            window.console.log(result);

            recognizer.close();
            recognizer = undefined;
          },
          function (err) {
            startRecognizeOnceAsyncButton.disabled = false;
            phraseDiv.innerHTML += err;
            window.console.log(err);

            recognizer.close();
            recognizer = undefined;
          });
      });

      if (!!window.SpeechSDK) {
        SpeechSDK = window.SpeechSDK;
        startRecognizeOnceAsyncButton.disabled = false;

        document.getElementById('content').style.display = 'block';
        document.getElementById('warning').style.display = 'none';

        // in case we have a function for getting an authorization token,
call it.
        if (typeof RequestAuthorizationToken === "function") {
            RequestAuthorizationToken();
        }
      }
    });
  </script>
  <!-- </quickstartcode> -->
</body>
</html>
```

Create the token source (optional)

In case you want to host the web page on a web server, you can optionally provide a token source for your demo application. That way, your subscription key will never leave your server while allowing users to use speech capabilities without entering any authorization code themselves.

Create a new file named `token.php`. In this example we assume your web server supports the PHP scripting language with curl enabled. Enter the following code:

PHP Copy

```php
<?php
header('Access-Control-Allow-Origin: ' . $_SERVER['SERVER_NAME']);

// Replace with your own subscription key and service region (e.g.,
"westus").
$subscriptionKey = 'YourSubscriptionKey';
$region = 'YourServiceRegion';

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, 'https://' . $region .
'.api.cognitive.microsoft.com/sts/v1.0/issueToken');
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, '{}');
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: application/json',
'Ocp-Apim-Subscription-Key: ' . $subscriptionKey));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
echo curl_exec($ch);
?>
```

> ⓘ Note
>
> Authorization tokens only have a limited lifetime. This simplified example does not show how to refresh authorization tokens automatically. As a user, you can manually reload the page or hit F5 to refresh.

Build and run the sample locally

To launch the app, double-click on the index.html file or open index.html with your favorite web browser. It will present a simple GUI allowing you to enter your subscription key and region and trigger a recognition using the microphone.

> ⓘ Note
>
> This method doesn't work on the Safari browser. On Safari, the sample web page needs to be hosted on a web server; Safari doesn't allow websites loaded from a local file to use the microphone.

Build and run the sample via a web server

To launch your app, open your favorite web browser and point it to the public URL that you host the folder on, enter your region, and trigger a recognition using the microphone. If configured, it will acquire a token from your token source.

Next steps

With this base knowledge of speech recognition, continue exploring the basics to learn about common functionality and tasks within the Speech SDK.

Explore speech recognition basics

Is this page helpful?

👍 Yes 👎 No