

Table of Contents

Quick Start 2

Quick Start

Please note that the code provided in this page is *purely* for learning purposes and is far from perfect. Remember to null-check all responses!

Breaking Changes Notice

If you've just updated the package, it is recommended to check the [changelogs](#) for information on breaking changes.

Setup

Add an instance of [TextToSpeechManager](#) to your scene, and set it up with your Google Cloud TTS API key. Check [this](#) guide on how to create an API key.

There are only two methods in `TextToSpeechManager`:

Method	What it does
SetApiKey	Sets the Text To Speech API key
Request ¹ ²	Computes a request on the TTS API

In this page, the fields, properties and methods of each type will not be explained. Every type has been fully documented in code, so please check the code docstrings or [reference documentation](#) to learn more about each type.

Beta API

`TextToSpeechManager` supports both the v1 and v1beta TTS API versions. To use the Beta API, you can set the `useBetaApi` boolean parameter in the request object's constructor.

Synthesis

This is a simple script which synthesizes some text:

```
using Uralstech.UCloud.TextToSpeech;
using Uralstech.UCloud.TextToSpeech.Synthesis;

private AudioSource _audioSource;

protected void Start()
{
    if (!TryGetComponent(out _audioSource))
```

```

        _audioSource = gameObject.AddComponent<AudioSource>();

        Speak("Hello, World!");
    }

    private async void Speak(string text)
    {
        const TextToSpeechSynthesisAudioEncoding encoding =
            TextToSpeechSynthesisAudioEncoding.WavLinear16;

        Debug.Log("Sending TTS request.");
        TextToSpeechSynthesisResponse response = await
            TextToSpeechManager.Instance.Request<TextToSpeechSynthesisResponse>
            (new TextToSpeechSynthesisRequest()
            {
                Input = new TextToSpeechSynthesisInput(text),
                Voice = new TextToSpeechSynthesisVoiceSelection("en-US"),
                AudioConfiguration = new TextToSpeechSynthesisAudioConfiguration(encoding)
            });

        Debug.Log("TTS response received, playing audio.");
        AudioClip clip = await response.ToAudioClip(encoding);

        _audioSource.PlayOneShot(clip);
    }

```

Here, we just create a [TextToSpeechSynthesisRequest](#), pass it to [TextToSpeechManager](#), await the result and convert it to an [AudioClip](#). That's all!

Now, let's go over the parameters of [TextToSpeechSynthesisRequest](#):

- [TextToSpeechSynthesisInput](#)
 - Contains text input to be synthesized.
 - It has two fields, [Text](#) and [Ssml](#). One of them **must** be provided. See [SSML](#) for more details.
 - The [constructor](#) has a boolean, [isSsml](#), for setting the [Text](#) or [Ssml](#) field. It is [false](#) by default.
- [TextToSpeechSynthesisVoiceSelection](#)
 - Description of which voice to use for a synthesis request.
 - It has fields for all the parameters needed for the desired voice, like [Gender](#), [Name](#), [CustomVoiceParameters](#), etc., but the main required field is [LanguageCode](#).

- For example, you can create a request that uses the [Journey](#) voice with the following `TextToSpeechSynthesisVoiceSelection`:

```
new TextToSpeechSynthesisVoiceSelection("en-US")
{
    Name = "en-US-Journey-F"
},
```

- [TextToSpeechSynthesisAudioConfiguration](#)

- Description of audio data to be synthesized.
- Contains fields for configuring the response audio from the TTS API, mainly `Encoding`. Not all encodings are supported by the `ToAudioClip` method. Unsupported encodings will have to be converted manually. These are the supported encodings:
 - `WavLinear16`
 - `Mp3`
 - `Mp3_64Kbps` (Requires Beta API)

The response from the synthesis request, [TextToSpeechSynthesisResponse](#), only contains the raw audio data, as a base64 encoded string. There are some other fields for the Beta API, but you can check the reference docs for that.

Listing Voices

You can also request a list of available voices through the API:

```
using Uralstech.UCloud.TextToSpeech;
using Uralstech.UCloud.TextToSpeech.Voices;

private async void ListAllVoices()
{
    Debug.Log("Getting all voices for en-US.");
    TextToSpeechVoiceListResponse voices = await
    TextToSpeechManager.Instance.Request<TextToSpeechVoiceListResponse>(
        new TextToSpeechVoiceListRequest("en-US"));

    Debug.Log($"Got the
    voices:\n{Newtonsoft.Json.JsonConvert.SerializeObject(voices.Voices)}");
}
```

It's just one line of code! You can also list **all** voices, for **every** language, by using the empty constructor for [TextToSpeechVoiceListRequest](#). To filter list of voices that have been returned by the API in

[TextToSpeechVoiceListResponse](#), check out the many extension methods that the plugin provides in [IEnumerableExtensions](#)!

Operation Endpoints

To use the operation endpoint methods, check out [UCloud.Operations](#)[↗], which is included as a dependency when you install this package.