Agentforce Speech Foundations Developer Guide

Salesforce, Spring ’25

@salesforcedocs

Last updated: Feb 6, 2025

Contents

Contents 2
About Agentforce Speech Foundations 3
Considerations and Limitations 4
Prerequisites 4
Get Started 4
    Create a Salesforce App 4
        Option 1: External Client App Instructions 4
        Option 2: Connected App Instructions 5
    Create a Token 6
Call the API 7
    Call the Einstein Transcribe Endpoint 7
    Call the Einstein Speech Endpoint 8
    Call the Einstein Translate Endpoint 9
Supported Languages 10
        For Speech Synthesis (Einstein Speech) 10
        For Translation (Einstein Translate) 10
        For Speech to Text (Einstein Transcribe) 10
Troubleshooting 11

About Agentforce Speech Foundations

Agentforce Speech Foundations is a suite of AI-powered services, including Einstein Transcribe (speech-to-text), Einstein Speech (text-to-speech), and Einstein Translate. These services are designed to provide developers with powerful Speech building blocks, reducing time to market and enabling sophisticated natural language-powered AI Agents and applications.

EDITIONS

Agentforce Speech Foundations is available in Lightning Experience.

Available in:
- Enterprise
- Performance
- Unlimited
- Developer

Prerequisites

Before you can start using Agentforce Speech Foundations features, be sure that you have Einstein Generative AI turned on in your org. See Set Up Einstein Generative AI in Salesforce Help.

Get Started

In order to get started with the Agentforce Speech Foundations API, create a Salesforce app and a token. When you have a valid token, you can call the API.

Create a Salesforce App

To use the Agentforce Speech Foundations API, you must create an External Client App or a Connected App. Both app types allow external services to integrate with Salesforce APIs using well-known authorization protocols, such as OAuth. External Client Apps (ECAs) are the new generation of Salesforce apps, and we suggest that you use an ECA for this purpose. However, these basic guidelines apply to both app types.

Option 1: External Client App Instructions

Before you can use Agentforce Speech Foundations, create an external client app (ECA) and enable OAuth settings. Follow these guidelines when creating the ECA.

1. Check **Enable OAuth** for the ECA.
2. For the callback URL, specify `https://login.salesforce.com`.
3. When enabling OAuth, be sure to include these OAuth scopes:
    - **Access the Salesforce API Platform (sfap_api)**
    - **Manage user data via APIs (api)**
    - **Perform requests at any time (refresh_token, offline_access)**
4. Select **Enable Client Credentials Flow**.
5. Select **Issue JSON Web Token (JWT)-based access tokens for named users**.
6. Deselect **Require Proof Key for Code Exchange (PKCE) Extension for Support Authorization Flows**.
7. Save the ECA.

After you save the external client app, from the ECA Manager page, click **Edit** and then expand the **OAuth Policies** section and follow these steps.

1. Check **Client Credentials Flow** and then set **Run As** to a username that has at least API Only access.
2. Select **Issue JSON Web Token (JWT)-based access tokens**. You can leave the 30-minute timeout value unless you have a reason to change it.
3. Save your changes.

To create a token, you need the consumer key and consumer secret from your ECA.

1. From Setup, select **External Client App Manager**.
2. Select your ECA.
3. Click the **Settings** tab.
4. Expand the **OAuth Settings** section and click **Consumer Key and Secret**. You may be asked to verify your identity before proceeding.
5. Copy **Consumer Key** and **Consumer Secret**.

Option 2: Connected App Instructions

Before you can use Agentforce Speech Foundations, create a connected app and enable OAuth settings. Follow these guidelines when creating the connected app.

1. For the callback URL, specify `https://login.salesforce.com`.
2. When enabling OAuth, be sure to include these OAuth scopes:
    - **Access the Salesforce API Platform (sfap_api)**
    - **Manage user data via APIs (api)**
    - **Perform requests at any time (refresh_token, offline_access)**
3. Deselect **Require Proof Key for Code Exchange (PKCE) Extension for Support Authorization Flows**.
4. Select **Enable Client Credentials Flow**.
5. Select **Issue JSON Web Token (JWT)-based access tokens for named users**.

6. Save the connected app.

After you save the connected app, from the connected app details page, click **Manage** to view the manage-side settings. Then click **Edit Policies** and follow these steps.

1. In the **Client Credentials Flow** section, set **Run As** to a user that has at least API Only access.
2. Select **Issue JSON Web Token (JWT)-based access tokens**. You can leave the 30-minute timeout value unless you have a reason to change it.
3. Save your changes.

To create a token, you need the consumer key and consumer secret from your connected app.

6. From Setup, select **App Manager**.
7. Find your connected app, click the dropdown arrow on the right, and then click **View.**
8. Click **Manage Consumer Details**. You may be asked to verify your identity before proceeding.
9. Copy **Consumer Key** and **Consumer Secret**.

Create a Token

All calls to the Speech Foundations API require a token. Create a token by using the consumer key, consumer secret, and your domain name.

```
None

curl https://{MY_DOMAIN_URL}/services/oauth2/token \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode 'client_id={CONSUMER_KEY}' \
--data-urlencode 'client_secret={CONSUMER_SECRET}'
```

- `{MY_DOMAIN_URL}`: You can get the domain from Setup. Search for **My Domain**. Copy the value shown in the **Current My Domain URL** field.
- `{CONSUMER_KEY`, `{CONSUMER_SECRET}`: You can get the consumer key and secret by following the instructions in Obtain Credentials.

The previous command returns a JSON payload similar to this response.

```
None

{
  "access_token":"eyJ0bmsiOiJj…(shortened)",
  "signature":"NpybSGypelvA…(shortened)",
  "token_format":"jwt",
  "scope":"sfap_api api",
  "instance_url":"https://mydomain.salesforce.com",
  "id":"https://login.mydomain.salesforce.com/id",
  "token_type":"Bearer",
  "issued_at":"1740482931553",
  "api_instance_url":"https://api.salesforce.com"
}
```

Verify that the `token_format` is “jwt” and that the `scope` includes “sfap_api”. Copy the access token specified in the `access_token` property. This token is required when making requests to the API.

Call the API

After you've set up your connected app and have a JWT, you’re ready to call the Agentforce Speech Foundations APIs. This section describes how to call the endpoints for each of the main Speech Foundations features.

Call the Einstein Transcribe Endpoint

This endpoint is used to transcribe audio into text. In order to transcribe audio, call the **`transcriptions`** endpoint.

- Specify the path to the audio file (instead of `path/to/audio/file`). To use a base-64-encoded string instead of a file, remove the `@` symbol before the input text.
- Use `EinsteinGPT` for the `x-sfdc-app-context` header.
- Use `external-edc` for the `x-client-feature-id` header.
- Replace `{YOUR_JWT}` with the token that you created in Create a Token.
- Default language (if parameter is not provided) is English, supported languages are detailed in the table below
- If you are interested in Streaming Transcribe, please contact your Solution Engineer.

```
None
curl --location
'https://api.salesforce.com/einstein/platform/v1/models/transcribeInternalV1/tr
anscriptions' \
--header 'Authorization: Bearer {YOUR_JWT}' \
--header 'x-sfdc-app-context: EinsteinGPT' \
--header 'x-client-feature-id: external-edc' \
--form 'input=@"path/to/audio/file"' \
--form 'engine="internal"' \
--form 'language="english"'
```

Sample response:

```
None

{
  "transcription" :
    [ "telephone Hello hello is Mary there. I'm sorry.
       You have the wrong number. Oh Is this 99999 no,
       it's not I'm sorry. That's okay" ]
}
```

Call the Einstein Speech Endpoint

This endpoint is used to convert text to speech. In order to convert text to speech, call the **`speech-synthesis`** endpoint.

- Specify the path to the text file (instead of `path/to/text/file`). To use a string instead of a file, remove the `@` symbol before the input text.
- Use `EinsteinGPT` for the x-sfdc-app-context header.
- Use `external-edc` for the x-client-feature-id header.
- Replace `{YOUR_JWT}` with the token that you created in Create a Token.
- This example uses the `Joanna` voice in the `voiceId` field. For other supported voices and their IDs, see https://docs.aws.amazon.com/polly/latest/dg/available-voices.html
- Please see the table below for supported languages

```
None
curl --location
'https://api.salesforce.com/einstein/platform/v1/models/transcribeInternalV1/sp
eech-synthesis' \
--header 'x-sfdc-app-context: EinsteinGPT' \
--header 'x-client-feature-id: external-edc' \
--header 'Authorization: Bearer {YOUR_JWT}' \
--form 'input=@"path/to/text/file"' \
--form 'engine="aws"' \
--form 'voiceId="Joanna"'
```

The audio stream returns a base-64-encoded result. You can convert this text to a file.

Sample response:

```
None

{
  "contentType" : "audio/mpeg",
  "requestCharacters" : 3,
  "audioStream" : "SUQzBAAAA…(shortened)"
}
```

Call the Einstein Translate Endpoint

This endpoint is used to translate text into another language. In order to translate text, call the **`translations`** endpoint.

- Specify the text you want to translate in the form `input` field. To specify a file instead of text, add the `@` symbol before the file path: `input=@"path/to/file"`.
- Use the form format to specify the `sourceLanguage` and `targetLanguage`.
- Use `EinsteinGPT` for the `x-sfdc-app-context` header.
- Use `external-edc` for the `x-client-feature-id` header.
- Replace `{YOUR_JWT}` with the token that you created in Create a Token.
- For more details about Languages please see https://docs.aws.amazon.com/translate/latest/dg/what-is-languages.html

```
None
curl --location
'https://api.salesforce.com/einstein/platform/v1/models/transcribeInternalV1/tr
anslations' \
--header 'x-sfdc-app-context: EinsteinGPT' \
--header 'x-client-feature-id: external-edc' \
--header 'Authorization: Bearer {YOUR_JWT}' \
--form 'input="Hello everyone. This is the text I would like to translate."' \
--form 'engine="aws"' \
--form 'sourceLanguage="en"' \
--form 'targetLanguage="es"'
```

Sample response:

```
None

{
  "translation" : "Hola a todos. Este es el texto que me gustaría traducir."
}
```

Supported Languages

For Speech Synthesis (Einstein Speech)

Please refer to Amazon Polly and specify `voiceId` in order to determine the language. Language codes currently aren't supported.

For Translation (Einstein Translate)

Please refer to Amazon Translate and note that you can specify `sourceLanguage` to be "auto" for language detection.

For Speech to Text (Einstein Transcribe)

Language name	"language" parameter
English	english
French	french
Spanish	spanish
Japanese	japanese

Chinese (Mandarin)	chinese
German	german
Italian	italian
Portuguese	portuguese
Arabic	arabic
Catalan	catalan
Danish	danish
Dutch	dutch
Hebrew	hebrew
Korean	korean
Norwegian	norwegian
Swedish	swedish
Finnish	finnish
Language detection	Not supported

Troubleshooting

If you run into any issues, refer to these troubleshooting tips.

404 Response: Not Found	
Error	You receive a 404 response when you attempt to call any of the endpoints.
Solution	Verify that you're using a valid token and that you're using the correct endpoint. See Create a Token.
500 Response: Internal Server Error	
Error	You receive a 500 response when you attempt to call any of the endpoints.
Solution	Verify that you've set up the external client app (or connected app)

	correctly. See Get Started.