

[Provision an Azure AI Translator resource](#)

[Prepare to develop an app in Cloud Shell](#)

[Configure your application](#)

[Add code to translate text](#)

[Clean up resources](#)

[More information](#)

# Translate Text

**Azure AI Translator** is a service that enables you to translate text between languages. In this exercise, you'll use it to create a simple app that translates input in any supported language to the target language of your choice.

While this exercise is based on Python, you can develop text translation applications using multiple language-specific SDKs; including:

- [Azure AI Translation client library for Python](#)
- [Azure AI Translation client library for .NET](#)
- [Azure AI Translation client library for JavaScript](#)

This exercise takes approximately **30** minutes.

## Provision an *Azure AI Translator* resource


If you don't already have one in your subscription, you'll need to provision an **Azure AI Translator** resource.

1. Open the Azure portal at `https://portal.azure.com`, and sign in using the Microsoft account associated with your Azure subscription.
2. In the search field at the top, search for **Translators** then select **Translators** in the results.
3. Create a resource with the following settings:
  - **Subscription:** *Your Azure subscription*
  - **Resource group:** *Choose or create a resource group*
  - **Region:** *Choose any available region*
  - **Name:** *Enter a unique name*
  - **Pricing tier:** Select **F0** (*free*), or **S** (*standard*) if F is not available.
4. Select **Review + create**, then select **Create** to provision the resource.
5. Wait for deployment to complete, and then go to the deployed resource.
6. View the **Keys and Endpoint** page. You will need the information on this page later in the exercise.

## Prepare to develop an app in Cloud Shell

To test the text translation capabilities of Azure AI Translator, you'll develop a simple console application in the Azure Cloud Shell.

1. In the Azure Portal, use the **[>]** button to the right of the search bar at the top of the page to create a new Cloud Shell in the Azure portal, selecting a **PowerShell** environment. The cloud shell provides a command line interface in a pane at the bottom of the Azure portal.

 **Note:** If you have previously created a cloud shell that uses a *Bash* environment, switch it to **PowerShell**.

2. In the cloud shell toolbar, in the **Settings** menu, select **Go to Classic version** (this is required to use the code editor).

**Ensure you've switched to the classic version of the cloud shell before continuing.**

3. In the PowerShell pane, enter the following commands to clone the GitHub repo for this exercise:

CodeCopy

```
rm -r mslearn-ai-language -f
git clone https://github.com/microsoftlearning/mslearn-ai-language
```

!

**Tip:** As you enter commands into the cloudshell, the output may take up a large amount of the screen buffer. You can clear the screen by entering the `cls` command to make it easier to focus on each task.

4. After the repo has been cloned, navigate to the folder containing the application code files:

CodeCopy

```
cd mslearn-ai-language/Labfiles/06-translator-sdk/Python/translate-text
```

## Configure your application

1. In the command line pane, run the following command to view the code files in the **translate-text** folder:

CodeCopy

```
ls -a -l
```

The files include a configuration file (**.env**) and a code file (**translate.py**).

2. Create a Python virtual environment and install the Azure AI Translation SDK package and other required packages by running the following command:

CodeCopy

```
python -m venv labenv
./labenv/bin/Activate.ps1
pip install -r requirements.txt azure-ai-translation-text==1.0.1
```

3. Enter the following command to edit the application configuration file:

CodeCopy

```
code .env
```

The file is opened in a code editor.

4. Update the configuration values to include the **region** and a **key** from the Azure AI Translator resource you created (available on the **Keys and Endpoint** page for your Azure AI Translator resource in the Azure portal).

!

**NOTE:** Be sure to add the *region* for your resource, not the endpoint!

5. After you've replaced the placeholders, within the code editor, use the **CTRL+S** command or **Right-click > Save** to save your changes and then use the **CTRL+Q** command or **Right-click > Quit** to close the code editor while keeping the cloud shell command line open.


## Add code to translate text

1. Enter the following command to edit the application code file:

CodeCopy

```
code translate.py
```

2. Review the existing code. You will add code to work with the Azure AI Translation SDK.

 **Tip:** As you add code to the code file, be sure to maintain the correct indentation.

3. At the top of the code file, under the existing namespace references, find the comment **Import namespaces** and add the following code to import the namespaces you will need to use the Translation SDK:

CodeCopy

```
# import namespaces
from azure.core.credentials import AzureKeyCredential
from azure.ai.translation.text import *
from azure.ai.translation.text.models import InputTextItem
```

4. In the **main** function, note that the existing code reads the configuration settings.

5. Find the comment **Create client using endpoint and key** and add the following code:

CodeCopy

```
# Create client using endpoint and key
credential = AzureKeyCredential(translatorKey)
client = TextTranslationClient(credential=credential, region=translatorRegion)
```

6. Find the comment **Choose target language** and add the following code, which uses the Text Translator service to return list of supported languages for translation, and prompts the user to select a language code for the target language:

CodeCopy


```
# Choose target language
languagesResponse = client.get_supported_languages(scope="translation")
print("{} languages supported.".format(len(languagesResponse.translation)))
print("(See https://learn.microsoft.com/azure/ai-services/translator/language-support#translation)")
print("Enter a target language code for translation (for example, 'en'):")
targetLanguage = "xx"
supportedLanguage = False
while supportedLanguage == False:
    targetLanguage = input()
    if targetLanguage in languagesResponse.translation.keys():
        supportedLanguage = True
    else:
        print("{} is not a supported language.".format(targetLanguage))
```

7. Find the comment **Translate text** and add the following code, which repeatedly prompts the user for text to be translated, uses the Azure AI Translator service to translate it to the target language (detecting the source language automatically), and displays the results until the user enters *quit*:

CodeCopy

```
# Translate text
inputText = ""
while inputText.lower() != "quit":
    inputText = input("Enter text to translate ('quit' to exit):")
    if inputText != "quit":
        input_text_elements = [InputTextItem(text=inputText)]
        translationResponse = client.translate(body=input_text_elements, to_language=[targetLanguage])
        translation = translationResponse[0] if translationResponse else None
        if translation:
            sourceLanguage = translation.detected_language
            for translated_text in translation.translations:
                print(f"'{inputText}' was translated from {sourceLanguage.language} to {translated_text.to} as '{translated_text.text}'.")
```

8. Save your changes (CTRL+S), then enter the following command to run the program (you maximize the cloud shell pane and resize the panels to see more text in the command line pane):

Code  Copy

```
python translate.py
```

9. When prompted, enter a valid target language from the list displayed.
10. Enter a phrase to be translated (for example `This is a test` or `C'est un test` ) and view the results, which should detect the source language and translate the text to the target language.
11. When you're done, enter `quit` . You can run the application again and choose a different target language.

## Clean up resources

If you're finished exploring the Azure AI Translator service, you can delete the resources you created in this exercise. Here's how:

1. Close the Azure cloud shell pane
2. In the Azure portal, browse to the Azure AI Translator resource you created in this lab.
3. On the resource page, select **Delete** and follow the instructions to delete the resource.

## More information

For more information about using **Azure AI Translator**, see the [Azure AI Translator documentation](#).