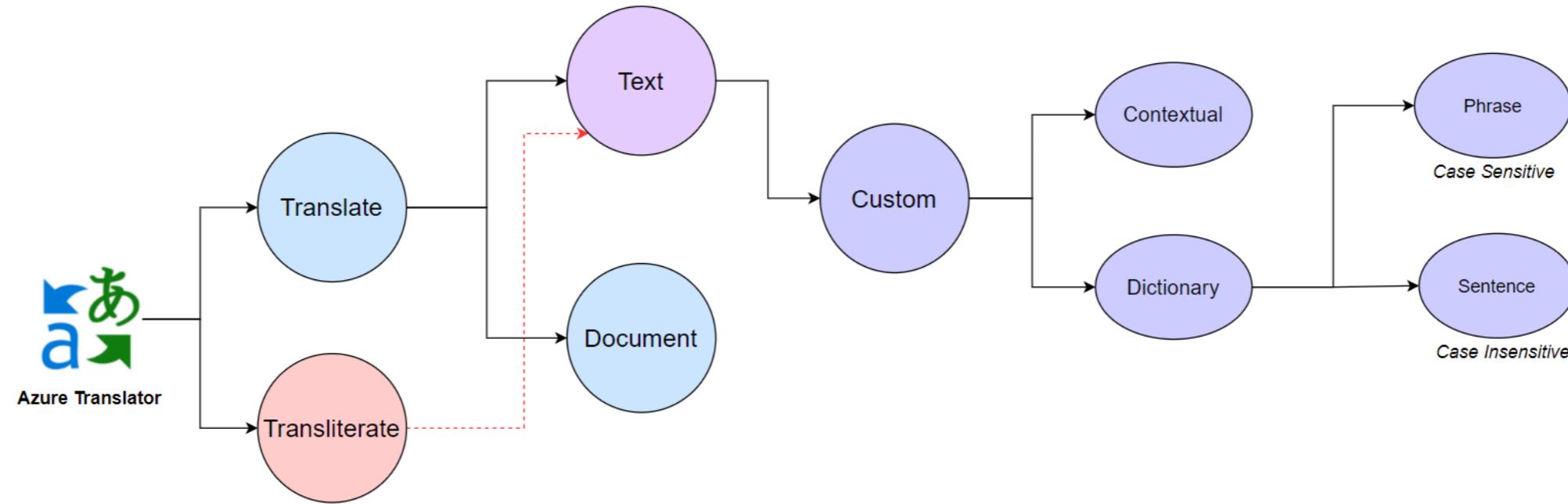


Document Name	HOL – Azure Translator Service v3.2
Author	Shiva S Tomar & Anupreet Kaur
Reviewer	
Executive Summary	Azure Cognitive APIs enable the developers of all skill levels to add human intelligence in their applications. The services are designed for developers interested in pursuing DS/AI/ML skills and people who want to acquire the deep technical knowledge on the Cognitive APIs of Azure, despite not having Machine Learning expertise.
Purpose	This document is created to help you gain level 350 working knowledge on Azure Translator Service. You will be able to explore each functionality offered by the service through the GUI Portal & REST APIs and observe the outcomes. We have also shared a sample dataset to replicate what we have used to create the content of this workshop. Once you complete these labs, you'll go from Zero to Hero on the respective Azure Cognitive service and should be able to Demo, Develop and Deploy your own custom use cases. The important thing to note here is that you don't need to refer any other documents to complete this workshop.
Intent of Guide	This workshop is designed to help you explore all the features of a service offered through their APIs. The diagram shown in the beginning of the document is its functional Architecture; talking about the functionalities offered by the service in a flow. It also covers the Concepts, How-to and best practices about the service. This document is not intended to enable you with scenarios of deployment in production.

Service brief: Azure Translator Service

Azure's Translator Service allows you to translate & transliterate text to the language of your choice.

Diagram: Functional Architecture



The service offers 2 main capabilities :

1. **Translate** : Translate specified source language text into the target language text.
2. **Transliterate** : Map source language script or alphabet to a target language script or alphabet.

Translate works on raw text (Text Translator API) as well as any document containing text (Document Translator API), for Eg PowerPoint, Word Document, PDF etc.

Raw text is used to process real time text by leveraging the Text translator API. Document Translator API opens the document, translates the content and publishes the output translated document while maintaining the format. All of this is carried out in batch mode.

Azure Translator also allows you to create custom models for your use case or business domain to process real time text. Custom models can be created using 2 approaches :

1. **Contextual** : By passing sentence-based training document. This model takes context into consideration
2. **Dictionary** : This is like a 1:1 mapping for source to target translation. Phrase model is used for words & phrases and is Case Sensitive hence will input text to trained outcome only if there is a 100% match. Sentence model is used for sentence level mapping and is case insensitive. Use a dictionary in your training, when you want Microsoft Translator to always translate any instances of the source phrase or sentence, using the translation you've provided in the dictionary.

These APIs are available both as REST APIs and language specific SDKs.

You can deploy the models on the cloud or on the edge.

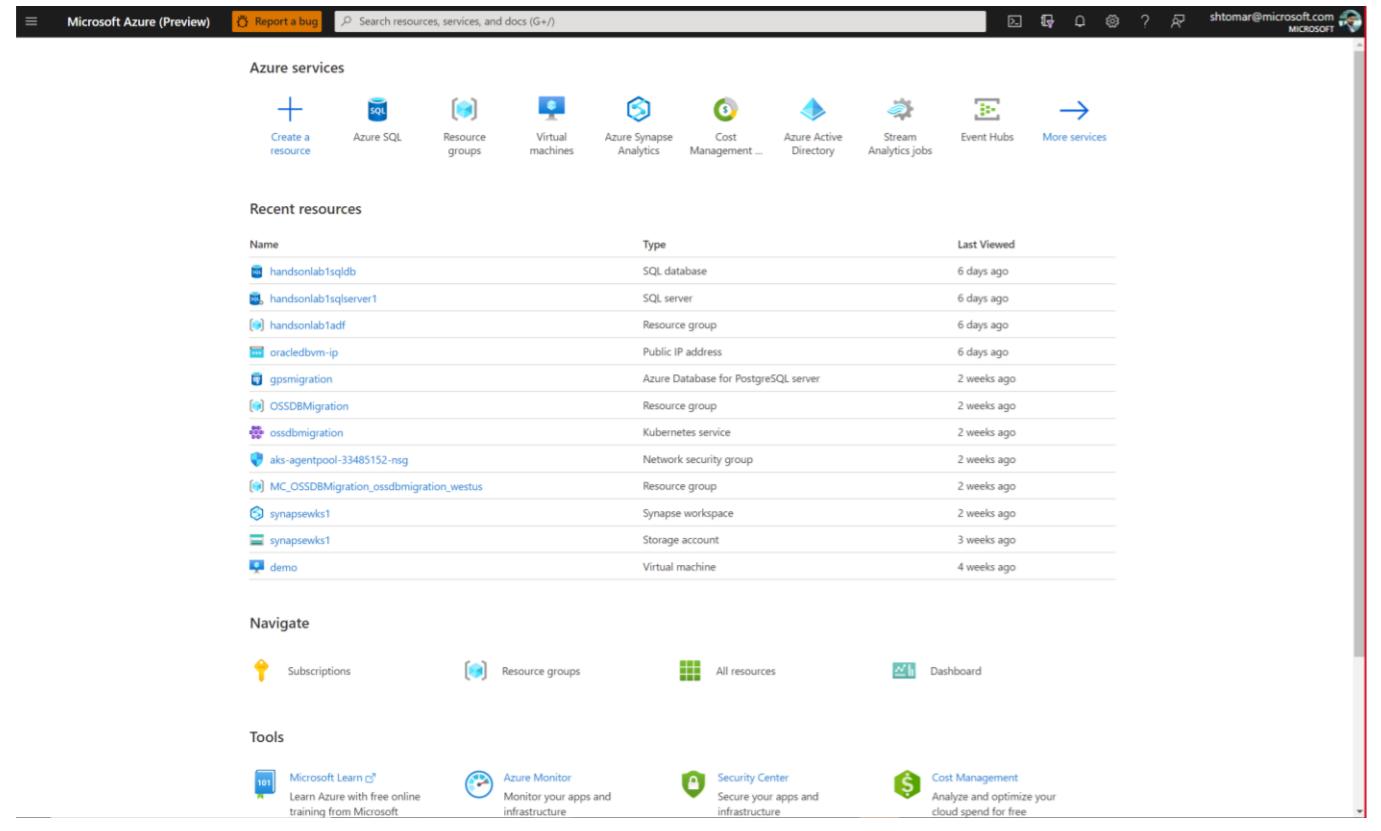
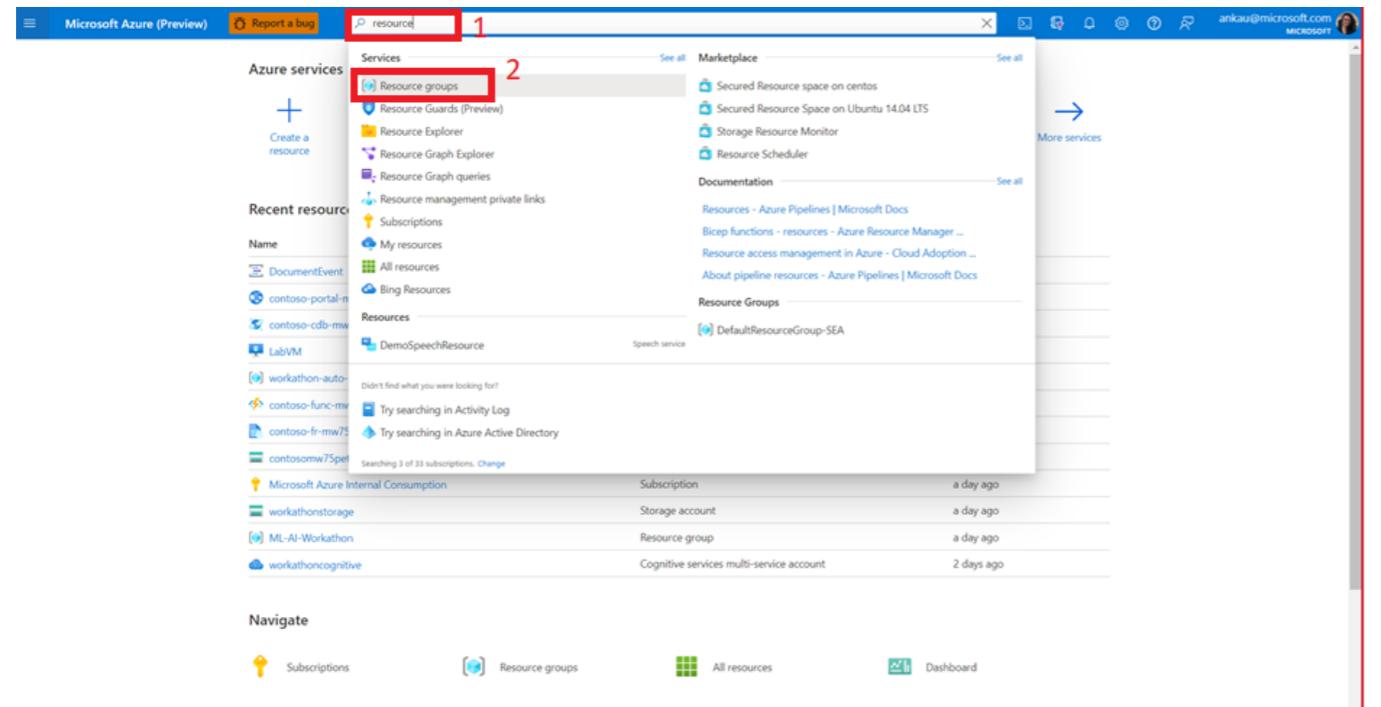
Step by step hands on guide to go from Zero to Hero

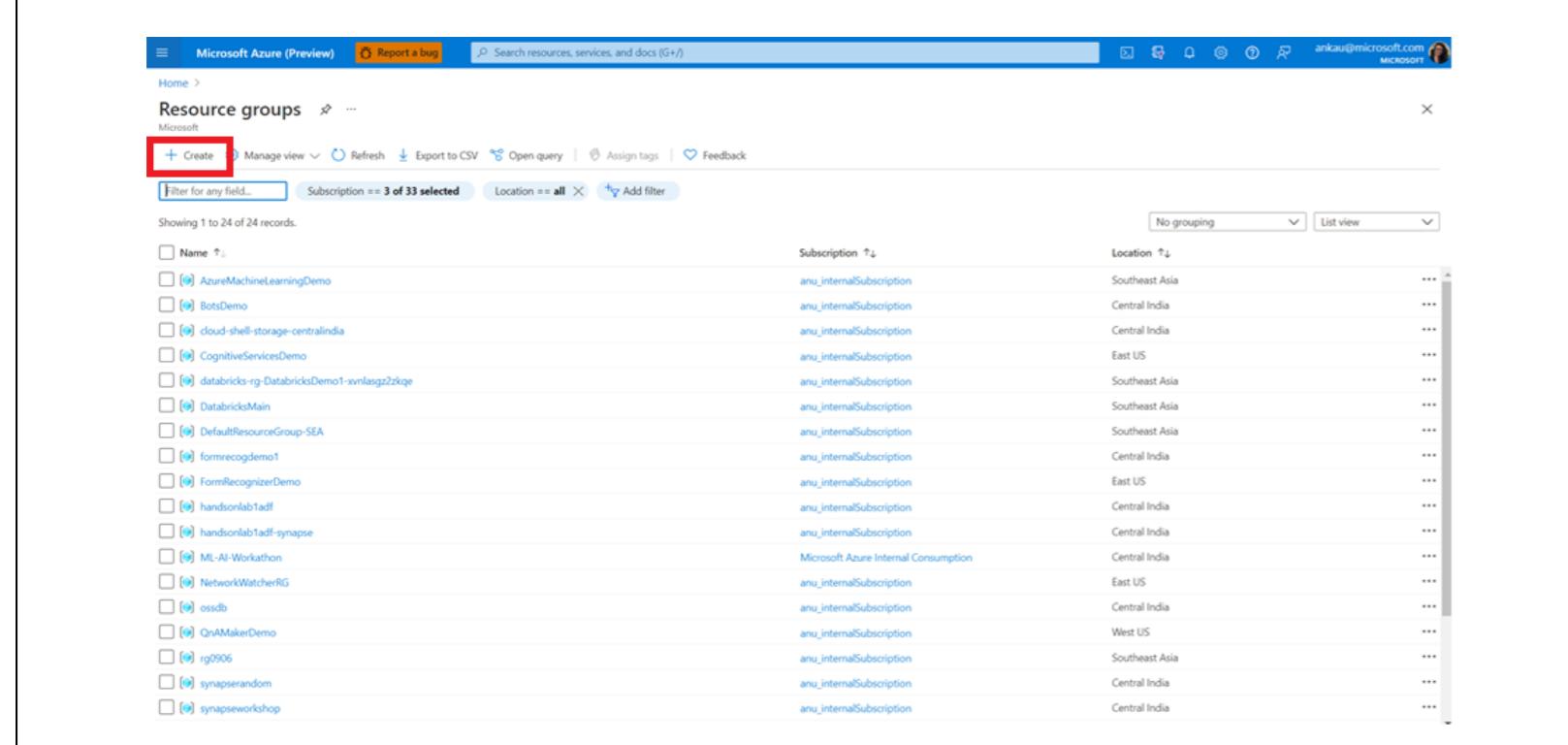
Pre-requisites

- Download & Install Postman
 - o Postman is a free tool which allows you to make API calls
 - o You can download the desktop application or get started using the web version ([Download Postman | Try Postman for Free](#))

- An active Azure Account
 - You can use your current Azure Subscription or get started by creating a free trial account (<https://azure.microsoft.com/en-in/free>)
- Download the data from Data folder, if any.

Let's get started!

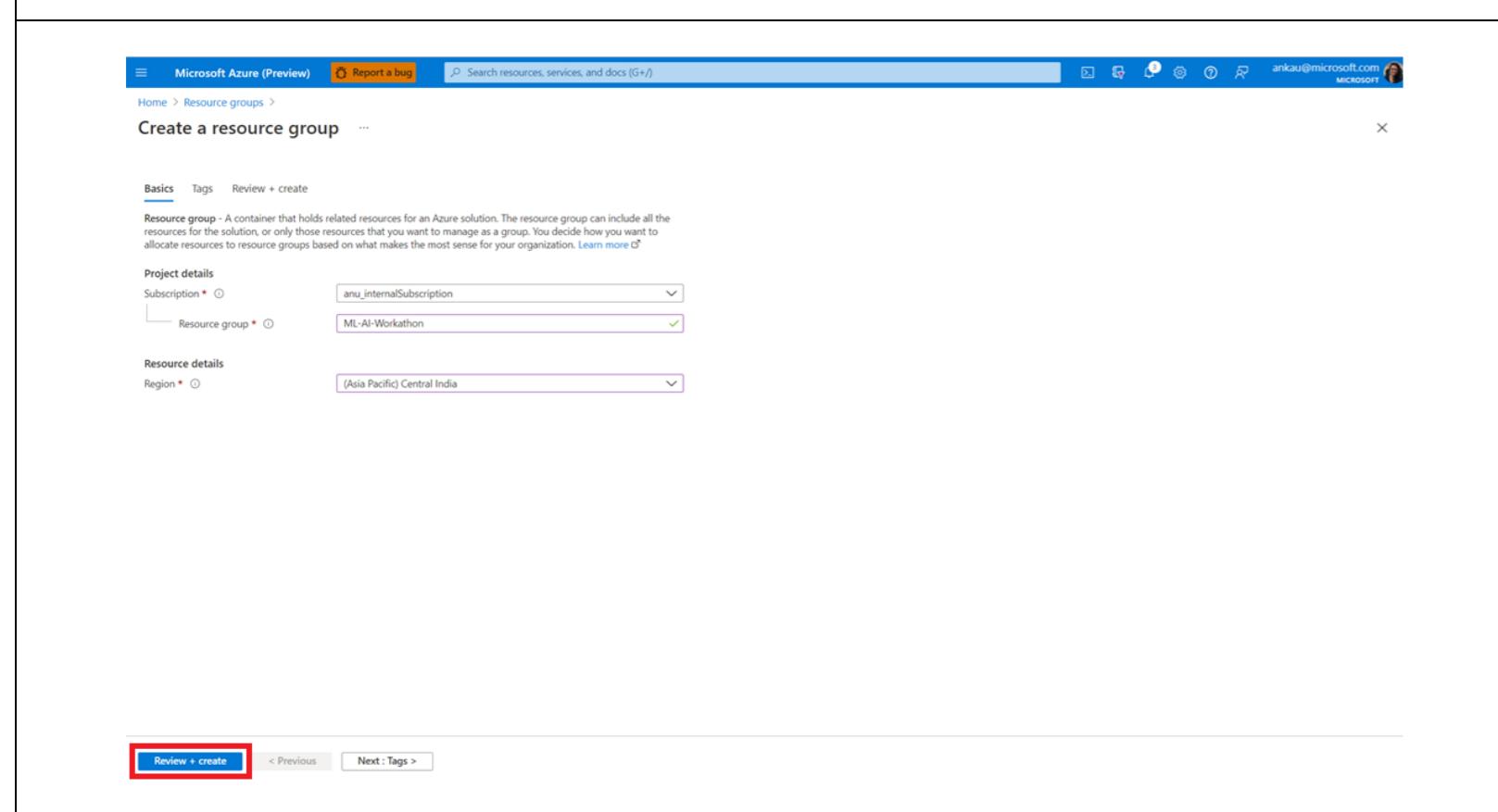
Screenshots	Steps & Significance
	<p>Sign into your Azure Portal.</p>
	<p>Create a Resource Group</p> <p>Follow steps 1 & 2 to create a resource group.</p> <p>You can skip this step if you already have a Resource Group in place.</p>



The screenshot shows the Microsoft Azure Resource Groups page. At the top left, there's a red box around the '+ Create' button. Below it, a table lists 24 resource groups. The columns are 'Name' (with a sort arrow), 'Subscription' (sorted by name), and 'Location'. Some entries in the table are truncated.

Name	Subscription	Location
anu_internalSubscription	anu_internalSubscription	Southeast Asia
anu_internalSubscription	anu_internalSubscription	Central India
anu_internalSubscription	anu_internalSubscription	East US
anu_internalSubscription	anu_internalSubscription	Southeast Asia
anu_internalSubscription	anu_internalSubscription	Southeast Asia
anu_internalSubscription	anu_internalSubscription	South Central US
anu_internalSubscription	anu_internalSubscription	Central India
anu_internalSubscription	anu_internalSubscription	East US
anu_internalSubscription	anu_internalSubscription	West US
anu_internalSubscription	anu_internalSubscription	Southeast Asia
anu_internalSubscription	anu_internalSubscription	Central India
anu_internalSubscription	anu_internalSubscription	Central India

Click create to create a new resource group.



The screenshot shows the 'Create a resource group' wizard in the 'Basics' step. It has tabs for 'Basics', 'Tags', and 'Review + create'. The 'Basics' tab is active. It shows a summary of the project details: Subscription is 'anu_internalSubscription', Resource group is 'ML-AI-Workathon', and Region is '(Asia Pacific) Central India'. At the bottom, the 'Review + create' button is highlighted with a red box.

Enter the details –

1. Subscription : Azure subscription in which you want to deploy the resource group
2. Resource Group : Name of your choice for the resource group
3. Region : Region where you want to deploy the resource group

Click Review + Create.

The screenshot shows the Microsoft Azure portal interface. At the top, there is a search bar with the text 'Trans' highlighted by a red box. Below the search bar, the 'Services' section is expanded, showing 'Translators' selected. The main pane displays a list of resources under the 'Resources' tab, including various App Service plans, Computer vision, Custom vision, Storage accounts, and Logic Apps. A red box labeled '2' points to the 'Translators' entry in the service list.

Create Translator Service

Once the resource group is created, search for Translator in the search bar above and select Translator resource.

The multipurpose Cognitive Service account does not support advanced translator features like Document Translation. Thus, we have created a separate Translator Service account.

The screenshot shows the 'Create Translator' wizard in the Microsoft Azure portal. The current step is 'Basics'. The 'Subscription' dropdown is set to 'Microsoft Azure Internal Consumption' (1). The 'Resource group' dropdown is set to 'ML-AI-Workathon' (1). The 'Region' dropdown is set to 'Central India' (2). The 'Name' input field contains 'workathontranslator' (3). The 'Pricing tier' dropdown is set to 'Standard S1 (Pay as you go)' (4). A red box labeled '5' points to the 'Review + create' button at the bottom left.

Enter the details to create a new cognitive service as follows -

Project details	Description
Subscription	Select one of your available Azure subscriptions.
Resource group	The Azure resource group that will contain your Cognitive Services resource. You can create a new group or add it to a pre-existing group.
Region	The location of your cognitive service instance. Different locations may introduce latency but have no impact on the runtime availability of your resource.
Name	A descriptive name for your cognitive services resource.
Pricing tier	The cost of your Cognitive Services account depends on the options you choose and your usage.

Click Review + Create.

Microsoft Azure (Preview) | Report a bug | Search resources, services, and docs (G+) | shtomar@microsoft.com

Home > Create Translator ...

Validation Passed

Basics Identity Tags Review + create

TERMS

By clicking "Create" I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

Basics

Subscription: Microsoft Azure Internal Consumption
Resource group: ML-AI-Workathon
Region: Central India
Name: workathontranslator
Pricing tier: Standard S1 (Pay as you go)

Identity

Identity type: None

Create < Previous Next Download a template for automation

Verify the details and click Create.

Microsoft Azure (Preview) | Report a bug | Search resources, services, and docs (G+) | shtomar@microsoft.com

Home > workathontranslator | Keys and Endpoint ...

Search (Ctrl+ /) Regenerate Key1 Regenerate Key2

These keys are used to access your Cognitive Service API. Do not share your keys. Store them securely- for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.

Show Keys

KEY 1
KEY 2

Location/Region: centralindia

Web API Containers

Use the below endpoints while using the Web API. To force the request to be handled by a specific geography, see here.

Text Translation: https://api.cognitive.microsofttranslator.com/

Document Translation: https://workathontranslator.cognitiveservices.azure.com/

Copy keys & endpoints

Click Key and Endpoints.

Copy the Key and Endpoint for Text Translation & Document Translation. Paste these in a notepad. You will leverage these in the next step, while setting up the global variables in Postman.

The image contains two screenshots of the Postman application interface. Both screenshots show the 'ML-AI Workathon' workspace.

Screenshot 1: Shows the 'Globals' section under the 'Environment' tab. A red box highlights the 'Edit' button at the top right of the table. The table lists two variables: 'endpoint' with value 'https://workathoncognitive.cognitiveservices.azure.com' and 'key' with value 'bf9fa979abdb4263af5dbae9ac124b5'. A note at the bottom says 'Use variables to reuse values in different places. Work with the current value of a variable to prevent sharing sensitive values with your team.' A red box also highlights the 'Open Overview' button at the bottom left.

Screenshot 2: Shows the 'Globals' section after changes have been made. A red box highlights the 'Save' button at the top right of the table. The table now includes three additional variables: 'text-translator' with value 'https://api.cognitive.microsofttranslator.com/', 'doc-translator' with value 'https://workathontranslator.cognitiveservices.azure.com/', and 'translator-key' with value '86b9b'. A red box highlights the entire list of variables.

We have now switched the interface to Postman to explore the Text Analytics Service. If you haven't downloaded the Postman client, you can use web version.

Configure global variables in Postman

Significance :

The global variables will be created once and leveraged time & again, in each API request that we make through Postman.

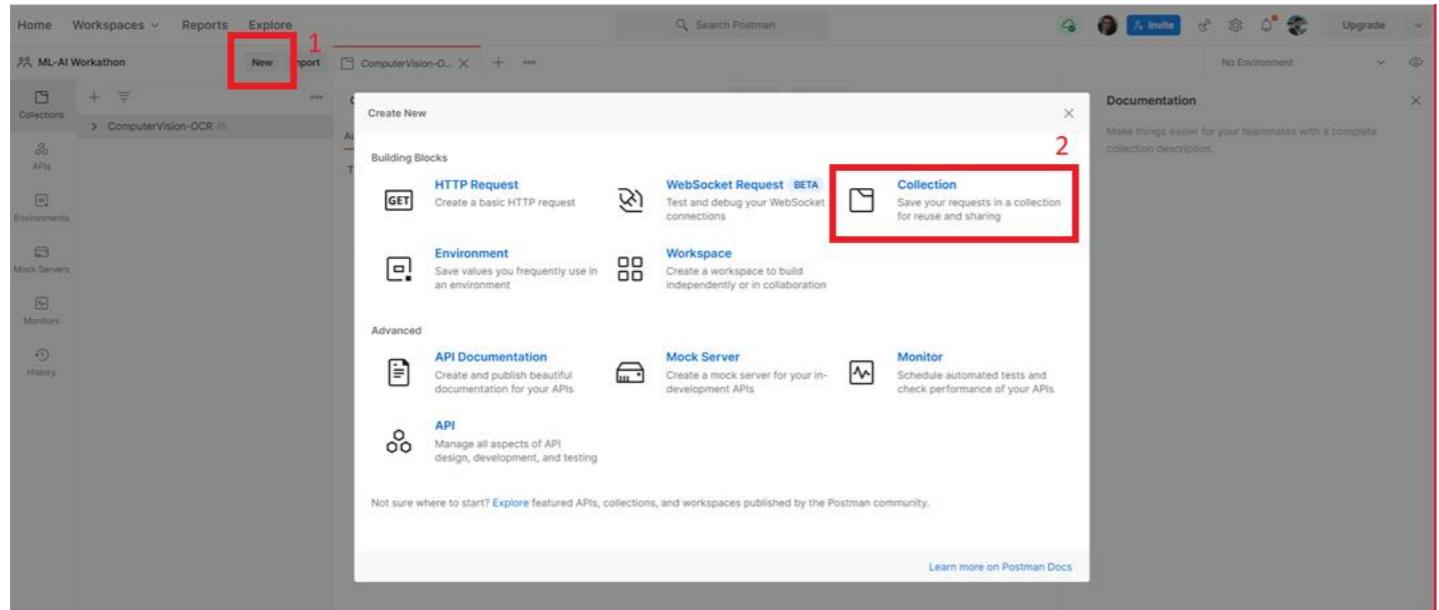
This way, you will not have to hard code the Endpoint & Key for every request you make, thereby, making it more secure. This will also save time and effort.

Follow step 1 & 2 to add global variables for :

1. Text-translator : Paste the endpoint of the Text Translator copied earlier
2. Document-translator : Paste the endpoint of the Document Translator copied earlier
3. Key - Paste the Key of the Translator service account copied earlier

Once you have added the global variables, they will appear in your global variables section, as shown in image 2.

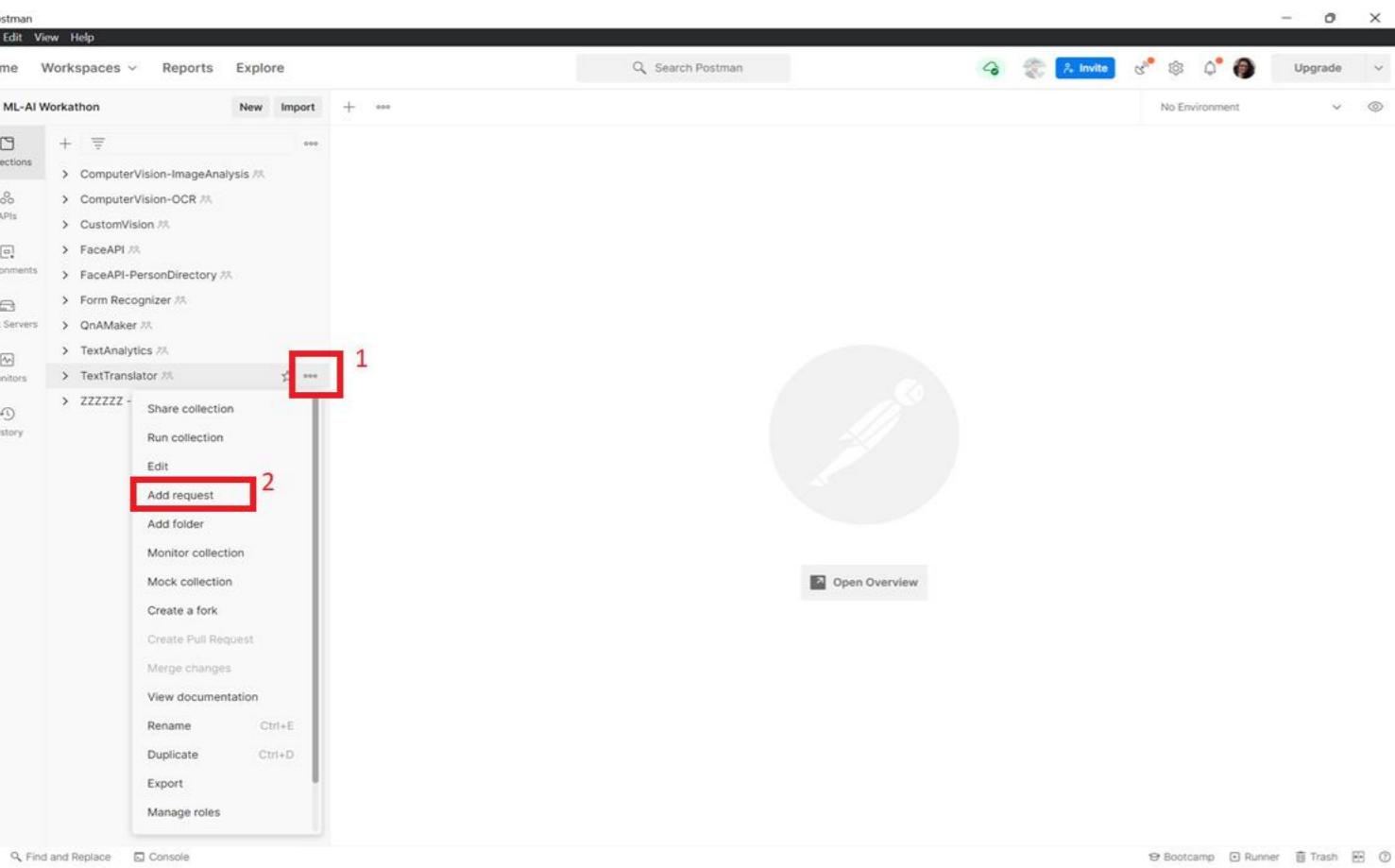
Select Save.



Create new collection in Postman

Open Postman > select New.
On the pop up select Collection.
Name the collection Text Translator.

Collection is like a folder for managing the API call requests.



Translate text to multiple languages

We will be translating text from English to multiple Indian Languages such as Hindi, Kannada, Tamil & Punjabi

This request upon successful execution will return text in each of the chosen languages.

URL : `{{text-translator}}/translate?api-version=3.0&to=hi,kn,ta,pa&includeSentenceLength=true`

Headers :

Ocp-Apim-Subscription-Key : `{{{key}}}`

Content-Type : application/json

Ocp-Apim-Subscription-Region : centralindia (change this if you provisioned your resource in a different region)

Body :

```
[{"Text": "Hello, what is your name?"}]
```

You should also try exploring with different input text. You can add multiple text inputs within the array.

Params :

These will be auto-populated from the URL

Significance of input & output

1. `{{{endpoint}}} , {{{key}}}` : Values being picked from global variables
2. Ocp-Apim-Subscription-Key : This is the Azure Cognitive service key, that will authenticate the request.
Content-Type : This refers to the input type that you provide in the body, for eg application/json allows you to enter body text in JSON format. Change the content-type on the basis of input you provide.
Ocp-Apim-Subscription-Region : region where your resource is deployed.
3. Params :
 - a. To : this takes as input the codes for the target languages, separated by a comma.
 - b. includeSentenceLength : Setting this to true will return the length of the input & translated text
4. After you execute the call, observe the status returned, as shown in step 11. This should reflect 200 OK. Observe the translated sentences.

```

[{"normalizedSource": "watch", "displaySource": "watch", "translations": [{"normalizedTarget": "\u20e3\u20e3\u20e3", "displayTarget": "\u20e3\u20e3\u20e3", "postag": "NOUN", "confidence": 0.3371, "prefixedword": "", "backtranslations": [{"normalizedText": "look", "displayText": "look", "numexamples": 15, "frequencyCount": 4337}, {"normalizedText": "watch", "displayText": "watch", "numexamples": 15, "frequencyCount": 4337}]}]
  
```

Dictionary Lookup API

This returns a dictionary for individual words in the target language and reverse translation from target language to source language along with synonyms and related words.

URL : {{text-translator}}/dictionary/lookup

Headers :

Ocp-Apim-Subscription-Key : {{key}}

Content-Type : application/json

Ocp-Apim-Subscription-Region : centralindia (change this if you provisioned your resource in a different region)

Body :

```
[
  {"Text": "Watch"},
  {"Text": "Play"}
]
```

You should also try exploring with different input text. You can add multiple text inputs within the array. The value for each text should be a single word. This will not return any results upon adding more than 1 word in a line item.

Params :

api-version=3.0

from=en

to=hi

Once you have added the Params, they will be appended to the URL and it will look similar to the one in the screenshot.

Significance of input & output

1. {{endpoint}}, {{key}} : Values being picked from global variables
2. Ocp-Apim-Subscription-Key : This is the Azure Cognitive service key, that will authenticate the request.
Content-Type : This refers to the input type that you provide in the body, for eg application/json allows you to enter body text in JSON format. Change the content-type on the basis of input you provide.
Ocp-Apim-Subscription-Region : region where your resource is deployed.
3. Params :
 - c. from : this takes as input the codes for the source languages
 - d. To : this takes as input the codes for the target languages

After you execute the call, observe the status returned, as shown in step 11. This should reflect 200 OK. Observe the translated sentences.

As part of the output, notice the **translations** & corresponding **backTranslations** for each of them.

The screenshots illustrate the steps to configure and execute a POST request to the Text Translator API.

- Screenshot 1:** Shows the collection navigation pane with the "TextTranslator - Translate + Transliterate - Indian vernacular" item selected. A red box highlights this selection.
- Screenshot 2:** The request configuration screen. The URL is `https://text-translator/text-translator/translate?api-version=3.0&to=pa&toScript=latin&from=hi`. The "Params" tab is selected, showing parameters: `api-version=3.0`, `to=pa`, `toScript=latin`, and `from=hi`. A red box highlights the URL and the "Params" tab.
- Screenshot 3:** The "Headers" tab is selected, showing three headers: `Ocp-Apim-Subscription-Key` (with value `{key}`), `Content-Type` (set to `application/json`), and `Ocp-Apim-Subscription-Region` (set to `centralindia`). A red box highlights the "Headers" tab and the header entries.
- Screenshot 4:** The "Body" tab is selected, showing a JSON array of input texts:

```
[{"Text": "ନମ୍ରକାର"}, {"Text": "ଘଡ଼ୀ"}, {"Text": "ପତି"}]
```

 A red box highlights the "Body" tab and the JSON content.
- Screenshot 5:** The "Send" button is highlighted with a red box.
- Screenshot 6:** The response status is shown as `Status: 200 OK` with a response time of `528 ms` and a size of `769 B`. A red box highlights the status bar.
- Screenshot 7:** The detailed response body is displayed, showing two sets of translations for the input texts. The first set is for "ନମ୍ରକାର" (Namaskar) and the second for "ଘଡ଼ୀ" (Khadi).

Translate & Transliterate

We will use this API to translate between different Indian Vernacular Languages & transliterate the same to English.

URL : `https://text-translator/text-translator/translate`

Headers :

`Ocp-Apim-Subscription-Key : {{key}}`

`Content-Type : application/json`

`Ocp-Apim-Subscription-Region : centralindia` (change this if you provisioned your resource in a different region)

Body :

```
[
  {"Text": "ନମ୍ରକାର"},
  {"Text": "ଘଡ଼ୀ"},
  {"Text": "ପତି"}
]
```

You should also try exploring with different input text. You can add multiple text inputs within the array.

Params :

`api-version=3.0`

`from=hi`

`to=pa`

`toScript=latin`

Once you have added the Params, they will be appended to the URL and it will look similar to the one in the screenshot.

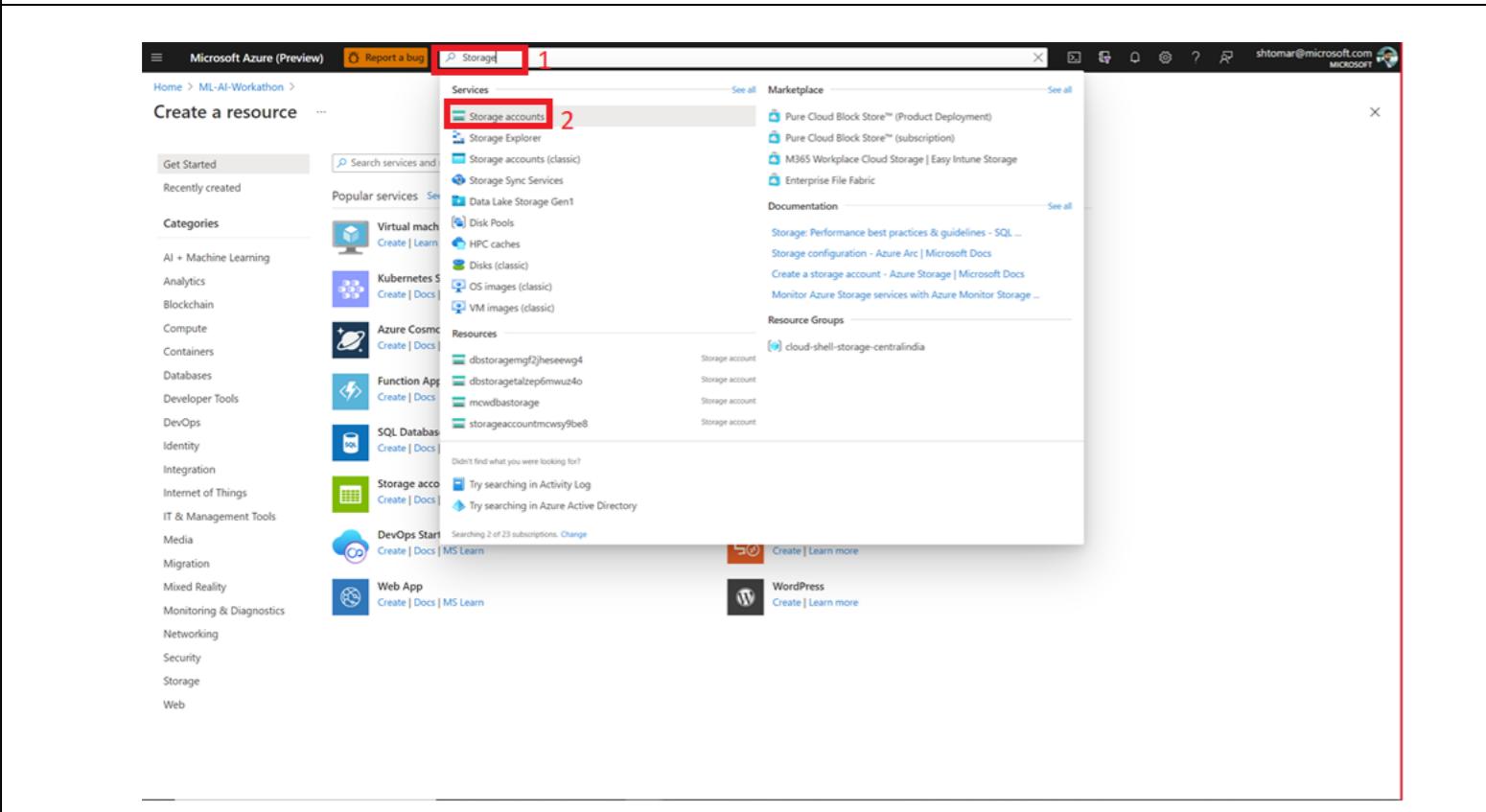
Significance of input & output

1. `Params` : Values being picked from global variables
2. `Ocp-Apim-Subscription-Key` : This is the Azure Cognitive service key, that will authenticate the request.
 - a. `Content-Type` : This refers to the input type that you provide in the body, for eg `application/json` allows you to enter body text in JSON format. Change the content-type on the basis of input you provide.
 - b. `Ocp-Apim-Subscription-Region` : region where your resource is deployed.
3. `Params` :
 - a. `from` : this takes as input the code for the source language
 - b. `To` : this takes as input the codes for the target languages
 - c. `toScript` : this takes as code the script for transliterated language

After you execute the call, observe the status returned, as shown in step 11. This should reflect 200 OK. Observe the translated sentences.

As part of the output, notice the **translations** & corresponding **transliteration** for each of them.

Document Translator



Create Storage Account in Azure Portal

As pre-requisites for Document Translation API, we will be creating Azure blob storage container to store the source documents & a placeholder folder for target documents.

Create a storage account

Project details
Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription Microsoft Azure Internal Consumption

Resource group ML-AI-Workathon Create new

Instance details
If you need to create a legacy storage account type, please click [here](#).

Storage account name workathonstorage

Region (Asia Pacific) Central India

Performance Standard: Recommended for most scenarios (general-purpose v2 account)
Premium: Recommended for scenarios that require low latency.

Redundancy Locally-redundant storage (LRS)

Review + create

Enter the details highlighted in the screenshot.
Click Review + Create.

Verify the details and hit Create.

Validation passed

Basics

Subscription	Microsoft Azure Internal Consumption
Resource Group	ML-AI-Workathon
Location	centralindia
Storage account name	workathonstorage
Deployment model	Resource manager
Performance	Standard
Replication	Locally-redundant storage (LRS)

Advanced

Secure transfer	Enabled
Allow storage account key access	Enabled
Default to Azure Active Directory authorization in the Azure portal	Disabled
Infrastructure encryption	Disabled
Blob public access	Enabled
Minimum TLS version	Version 1.2
Enable hierarchical namespace	Disabled
Enable network file share v3	Hot
Access tier	Large file shares
Large file shares	Disabled

Networking

Network connectivity	Public endpoint (all networks)
Microsoft network routing	disabled

Create

Storage Explorer (preview)

Create blob container

New container

Name * **3**

Public access level **4**

Create **5**

Create blob container

Go to the Storage Explorer (preview) to create a blob container. Follows the steps as numbered in the screenshot. (step 1,2)

Create a container with the following details –

1. Name : **text-translator** [You can choose a different name, however, that will change the path to your images while making API calls from postman. Make sure to change the image paths accordingly]
2. Public access level : Private (no anonymous access)

Click Create.

Microsoft Azure (Preview) | **Report a bug** | **Search resources, services, and docs (G+)**

Home > workathonstorage

workathonstorage | Storage Explorer (preview) | **Storage account** | **Directory: Microsoft**

1. **BLOB CONTAINERS**: *faceapi*, *formadhaar*, *formlicence*, **text-translator** (highlighted).

2. **+ New Folder** (highlighted).

3. **Name:** *source* (highlighted).

4. **OK** (highlighted).

5. **Upload** (highlighted).

6. **Open** (highlighted).

7. **Select a file** (highlighted).

8. **Sample cognitive ppt.pptx** (highlighted).

9. **Open** (highlighted).

10. **Upload** (highlighted).

Create folder

Make sure you are in the **text-translator** container, as shown in steps 1.

Select **+ New Folder** (step 2)

Name the folder – **source**. [You can choose a different name, however, that will change the path to your images while making API calls from postman. Make sure to change the image paths accordingly]

Click Ok.

Once you click Ok, make sure to stay on the same screen and upload files to it.

As this is a virtual folder and will not be accessible if you leave it without adding any files. Once you add a file / folder, it will persist and you will be able to access the path.

Click Upload (step 6)

Select file icon and browse to the file you want to translate (step 7-9)

Click Upload Button (step 10)

1. Right-click on the file 'Sample cognitive.ppt.pptx'.
2. Select 'Get Shared Access Signature...' from the context menu.

3. Set the expiry time.
4. Select permissions (e.g., Add, List).
5. Check the checkbox 'Generate container-level shared access signature URI'.
6. Click the 'Create' button.

7. Copy the generated SAS URL.
8. Click the 'Close' button.

Get SAS URI of the source file

A shared access signature (SAS) provides secure delegated access to resources in your storage account. With a SAS, you have granular control over how a client can access your data. For example:

- What resources the client may access
- What permissions they have to those resources.
- How long the SAS is valid

We require a SAS URI to grant access to the source file.

- In the containers section, right click on file
- Select Generate SAS

- Set up the SAS Expiry (Make sure to set this up to a later date since you will have to generate a new SAS key in case the keys expire)
- For Permissions select all**
- Tick check box 'Generate container level SAS'
- Click Generate SAS URL & Token (step 6)

- Copy the file SAS URL and paste it in a notepad. We will leverage this at a later step

** You should follow least access privilege however, for sake of simplicity, we have selected all for this Workshop.

This workshop is not considering security set up as this is designed to familiarise you with service capabilities.

We will be sharing the security best practices towards the end of the workshop.

Document Translation API

We will use this API to translate a Power Point document to Gujarati.

URL : {{doc-translator}}/translator/text/batch/v1.0/batches

Headers :

Ocp-Apim-Subscription-Key : {{key}}

Content-Type : application/json

Body :

```
{
  "inputs": [
    {
      "storageType": "File",
      "source": {
        "sourceUrl": " <>Paste the sas uri copied above within the quotes><>"
      }
    },
    "targets": [
      {
        "targetUrl": " <>copy the sas uri here as well & replace source folder name with target folder name & file name ><>",
        "language": "gu"
      }
    ]
}
```

If you want to place the target file in a different container, fetch the uri for that container as well.

Make sure to match the uri to input with the screenshots in terms of folder & file name compatibility. (step 7,8)

Significance of input & output

1. {{endpoint}}, {{key}} : Values being picked from global variables
2. Ocp-Apim-Subscription-Key : This is the Azure Cognitive service key, that will authenticate the request.

Content-Type : This refers to the input type that you provide in the body, for eg application/json allows you to enter body text in JSON format. Change the content-type on the basis of input you provide.

You will see the translated file in the target location after a while, once complete document is translated.

Optionally, to check the status of the translation, you can option the Operation-location from Headers section in the results (step 11).

Make a GET call to the Operation-location with the headers same as previous call. Observe the status. This will change to succeeded once the operation is completed.

This will also give you status for number of files succeeded & failed.

The screenshot shows the Microsoft Azure Storage Explorer (preview) interface. On the left, there's a sidebar with various navigation options like Overview, Activity log, Tags, etc. The main area shows a blob container named 'text-translator' under 'BLOB CONTAINERS'. Inside this container, there is a single file named 'cognitive-gu.pptx' with a size of 10.1 MB. The file is highlighted with a red box and labeled '3'. The 'FILE SHARES' section is also highlighted with a red box and labeled '1'. The 'QUEUES' and 'TABLES' sections are shown below 'FILE SHARES'. At the bottom, it says 'Showing 1 to 1 of 1 cached items'.

[View the translated file](#)

Navigate to the folder path you mentioned in Target URL to see the translated file. You can download this file to view the contents.

The image contains three screenshots illustrating the translation process:

- Postman Screenshot 1:** Shows a POST request to `/translator/text/batch/v1.0/batches`. The Body tab displays a JSON payload with inputs and targets. Step 1 highlights the 'POST Document Translator - generate...' item in the collection. Step 2 highlights the method and endpoint. Step 3 highlights the JSON body. Step 4 highlights the first target URL. Step 5 highlights the second target URL. Step 6 highlights the language 'pa'. Step 7 highlights the 'Send' button. Step 8 highlights the response headers, including 'Operation-Location'.
- Postman Screenshot 2:** Shows a GET request to `/GET document translation status`. The Headers tab includes 'Ocp-Apim-Subscription-Key' and 'Content-Type'. Step 9 highlights the 'GET Document translation status' item. Step 10 highlights the method and endpoint. Step 11 highlights the Headers section. Step 12 highlights the 'Content-Type' header. Step 13 highlights the 'Send' button. Step 14 highlights the response body, which shows a list of files with their status (e.g., succeeded, failed).
- Microsoft Azure Storage Explorer Screenshot:** Shows the storage account 'workathonstorage'. Step 1 highlights the 'Storage Explorer (preview)' item in the left sidebar. Step 2 highlights a blob container named 'formlendar'. Step 3 highlights a folder named 'text-translator'. Step 4 highlights the contents of the 'target' folder, showing three files: 'cognitive-gu.pptx', 'cognitive-hi.pptx', and 'cognitive-pa.pptx'.

To translate document into multiple languages at once

We will now translate a Power Point document to multiple Indian languages.

URL : `{{doc-translator}}/translator/text/batch/v1.0/batches`

Headers :

Ocp-Apim-Subscription-Key : `{{{key}}}`

Content-Type : application/json

Body :

```
{
  "inputs": [
    {
      "storageType": "File",
      "source": {
        "sourceUrl": "https://workathonstorage.blob.core.windows.net/text-translator/source/Sample cognitive.pptx?sp=racwd1&st=2021-08-16T11:32:37Z&se=2022-03-31T11:32:00Z&sv=2020-08-04&r=c&sig=t4ruvFm000HtY1z42f1pGK02mqhs0NvyUvAvkik30",
        "language": "hi"
      },
      "targets": [
        {
          "targetUrl": "https://workathonstorage.blob.core.windows.net/text-translator/target/cognitive-hi.pptx?sp=racwd1&st=2021-08-16T10:53:45Z&se=2022-03-31T10:53:00Z&sv=2020-08-04&r=c&sig=t4ruvFm000HtY1z42f1pGK02mqhs0NvyUvAvkik30",
          "language": "hi"
        },
        {
          "targetUrl": "https://workathonstorage.blob.core.windows.net/text-translator/target/cognitive-pa.pptx?sp=racwd1&st=2021-08-16T10:53:45Z&se=2022-03-31T10:53:00Z&sv=2020-08-04&r=c&sig=t4ruvFm000HtY1z42f1pGK02mqhs0NvyUvAvkik30",
          "language": "pa"
        }
      ]
    }
  ],
  "targets": [
    {
      "targetUrl": "copy the sas uri here as well & replace source folder name with target folder name & file name",
      "language": "hi"
    },
    {
      "targetUrl": "copy the sas uri here as well & replace source folder name with target folder name & file name",
      "language": "pa"
    }
  ]
}
```

If you want to place the target file in a different container, fetch the uri for that container as well.

Make sure to match the uri to input with the screenshots in terms of folder & file name compatibility. (step 4,5,6)

Significance of input & output

- `{{{endpoint}}} , {{{key}}} : Values being picked from global variables`
- Ocp-Apim-Subscription-Key : This is the Azure Cognitive service key, that will authenticate the request.

Content-Type : This refers to the input type that you provide in the body, for eg application/json allows you to enter body text in JSON format. Change the content-type on the basis of input you provide.

You will see the translated file in the target location after a while once complete document is translated.

Optionally, to check the status of the translation, you can option the Operation-location from Headers section in the results (step 11).

Make a GET call to the Operation-location with the headers same as previous call. Observe the status. This will change to succeeded once the operation is completed. This will also give you status for number of files succeeded & failed.

Navigate to the folder path you mentioned in Target URL to see the translated file. You can download this file to view the contents.

CUSTOM TRANSLATOR

Custom Translator

Learn how to customize Microsoft Translator's neural text and speech translation systems using your own training data to fit your style and terminology

LEARN MORE > SIGN IN > 2

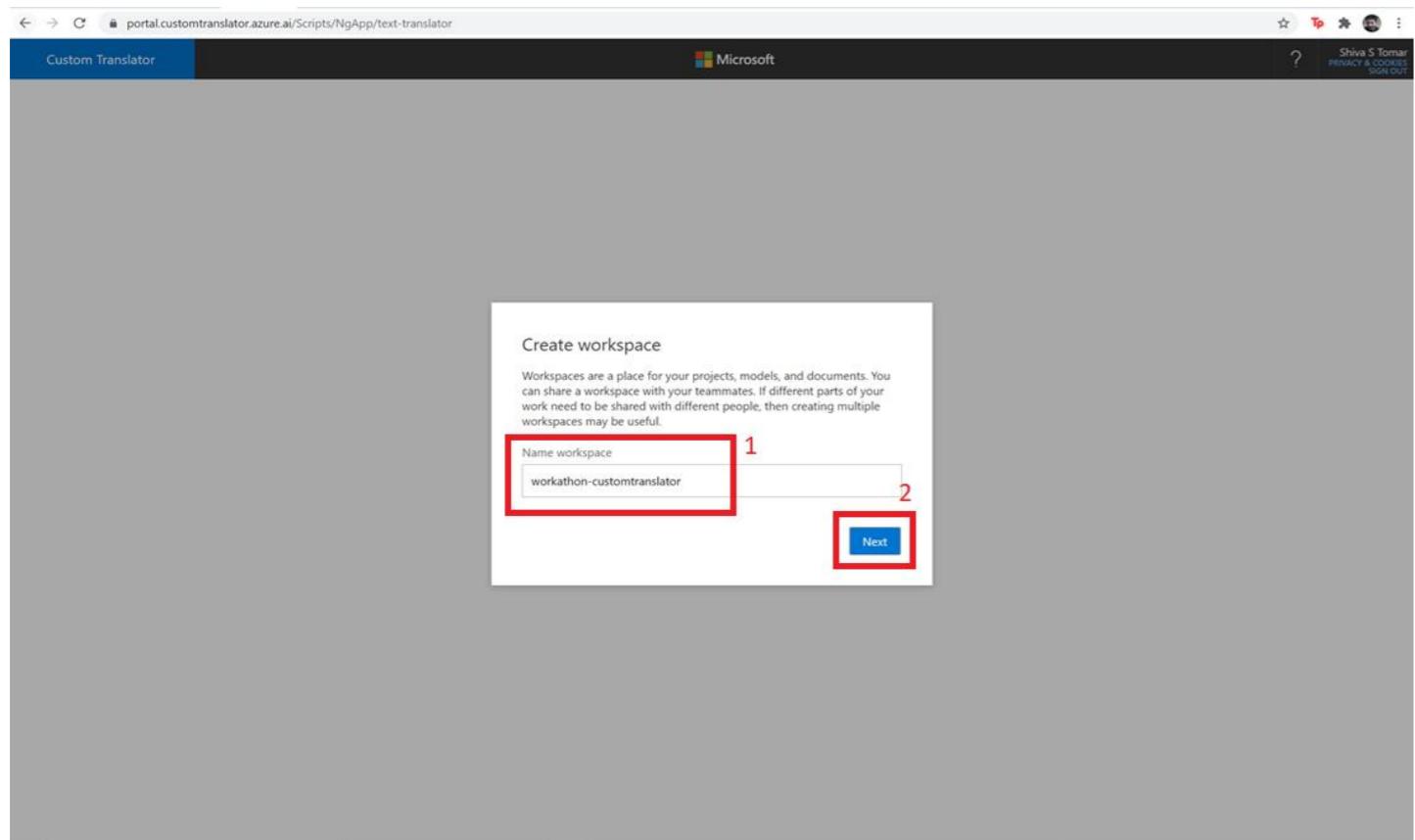
Help
Custom Translator documentation and support.

Custom Speech
Create custom speech models and overcome speech recognition

Custom Voice (preview)
A text-to-speech service that allows you to create a unique,

Go to [Custom Translator Portal](#).

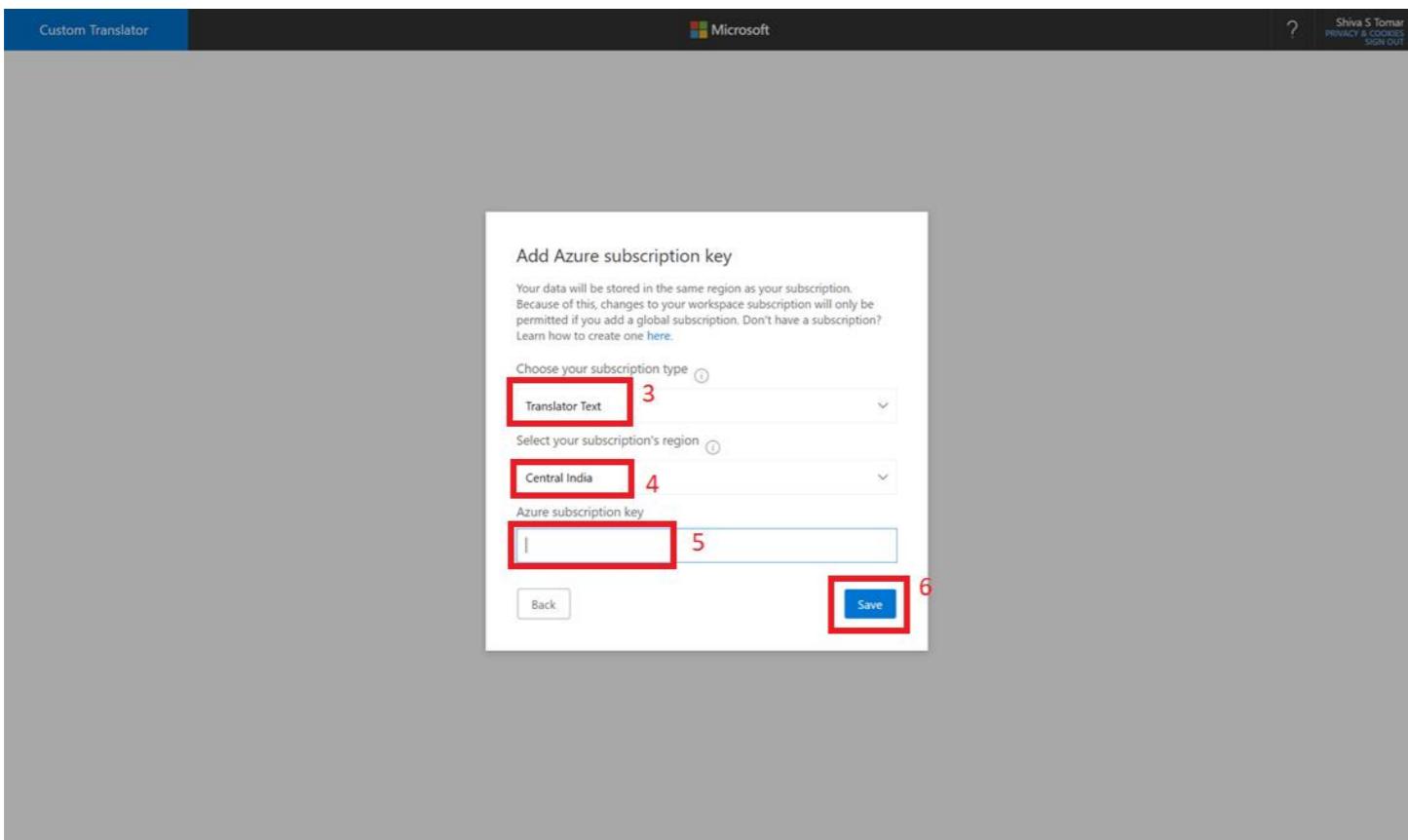
We will be creating a custom model for English to Hindi translation for e-commerce domain, using Phrase Dictionary document type.



Create workspace & set up account

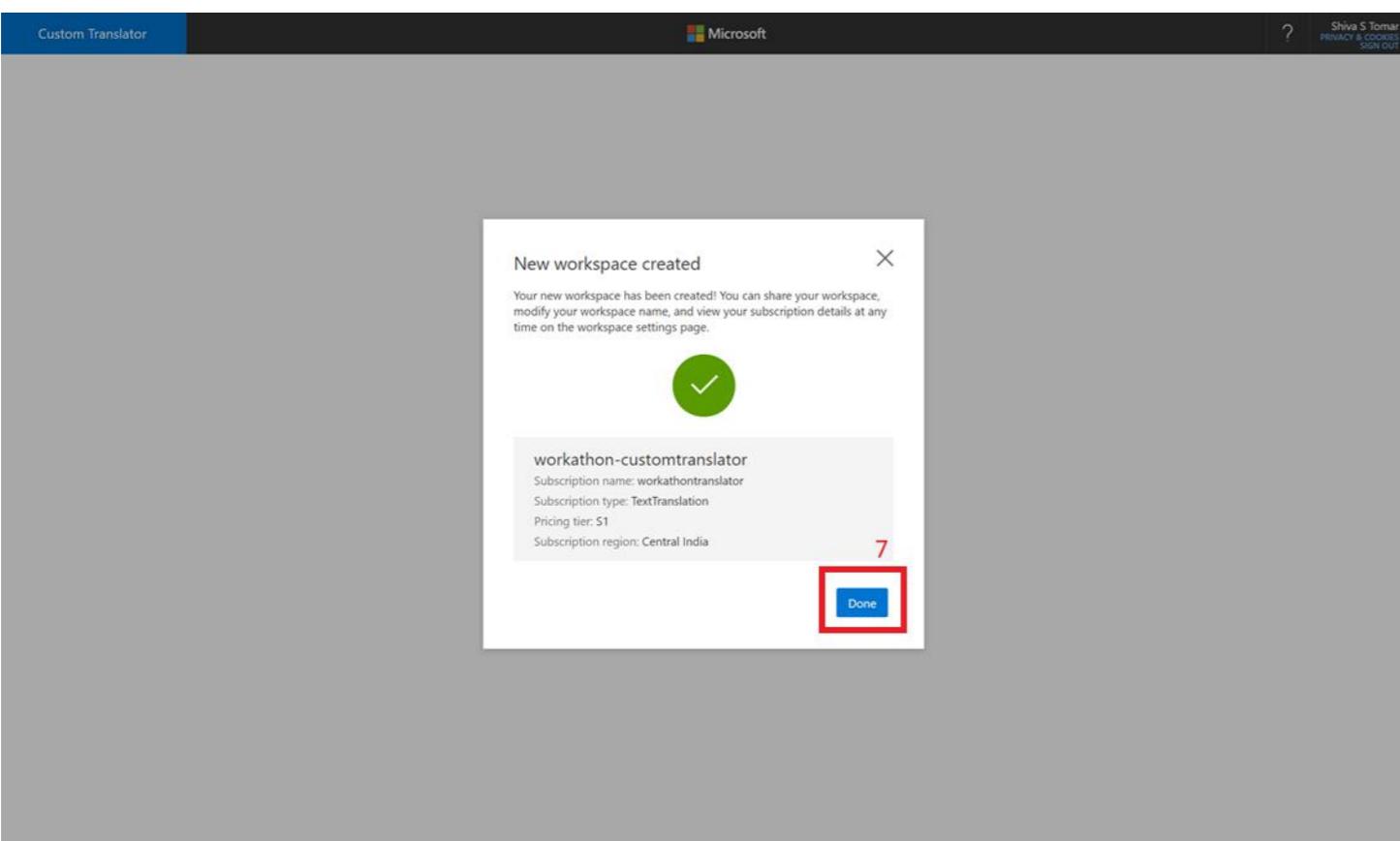
Create a workspace & select next. (step 1,2)

Workspace is like a logical container to hold & manage projects & documents for custom model you will build.

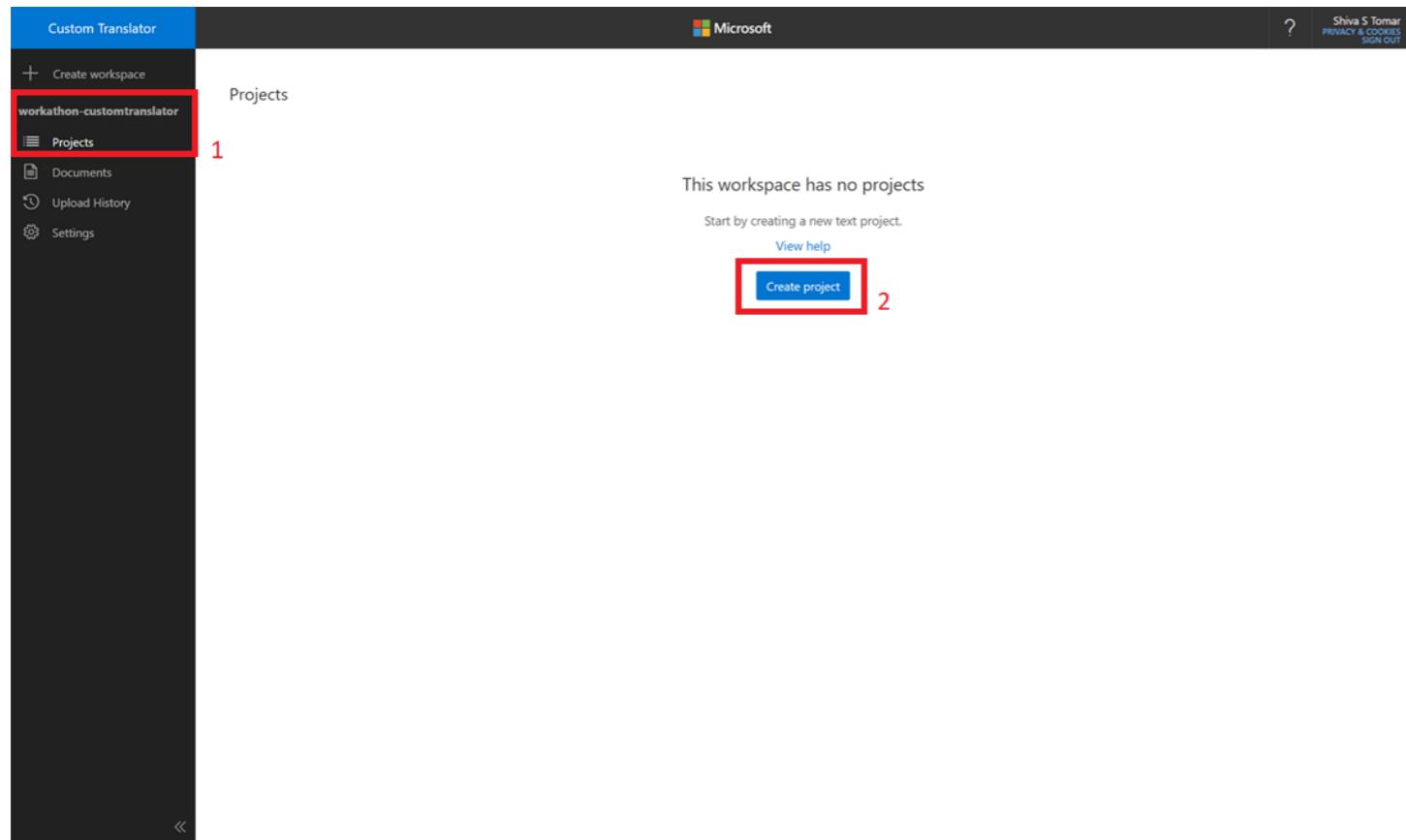


For steps 3 through 5, select the fields as shown and paste the key for the translator service we created in the beginning of the workshop.

Click Save button (step 6)

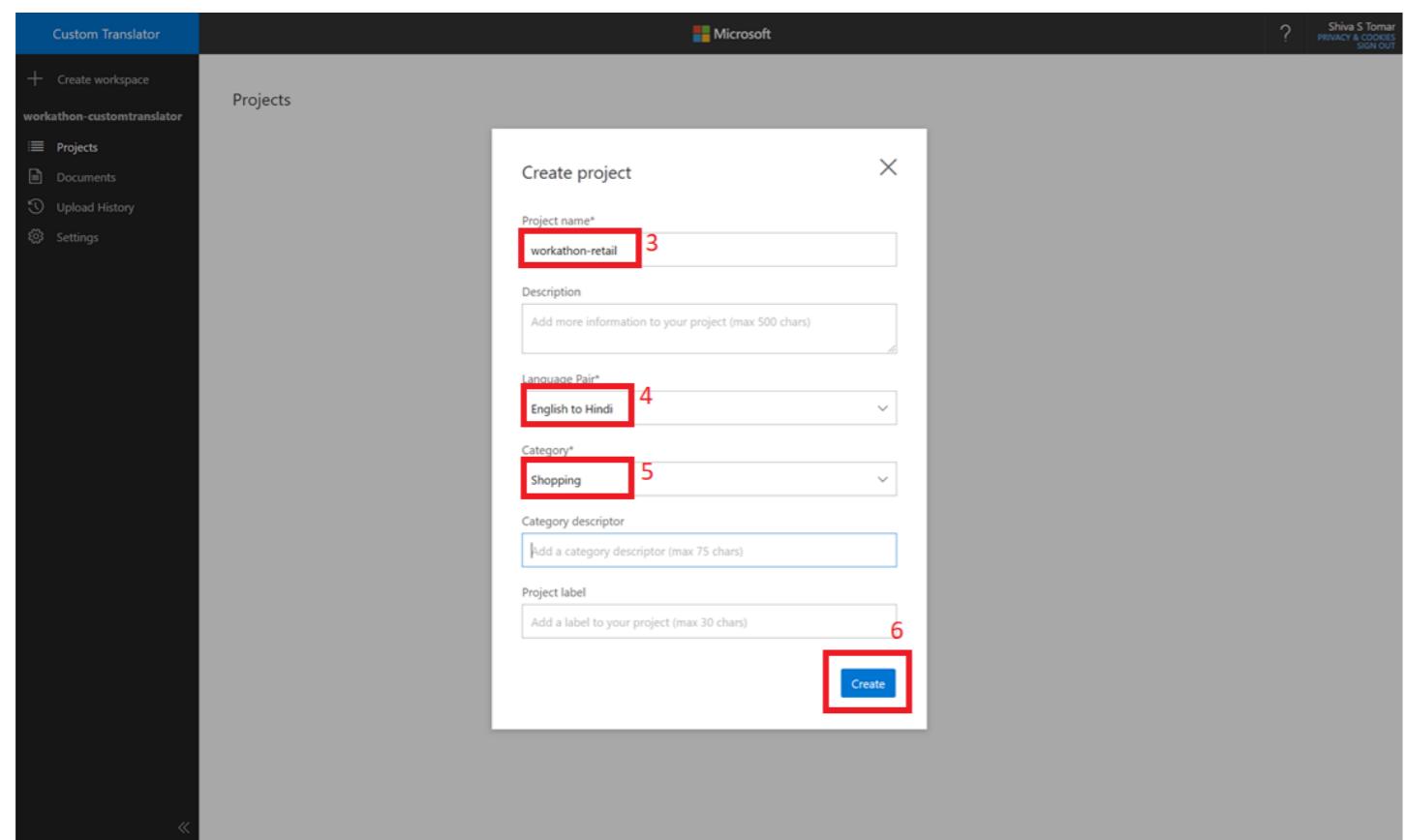


Verify the details and click Done (step 7)



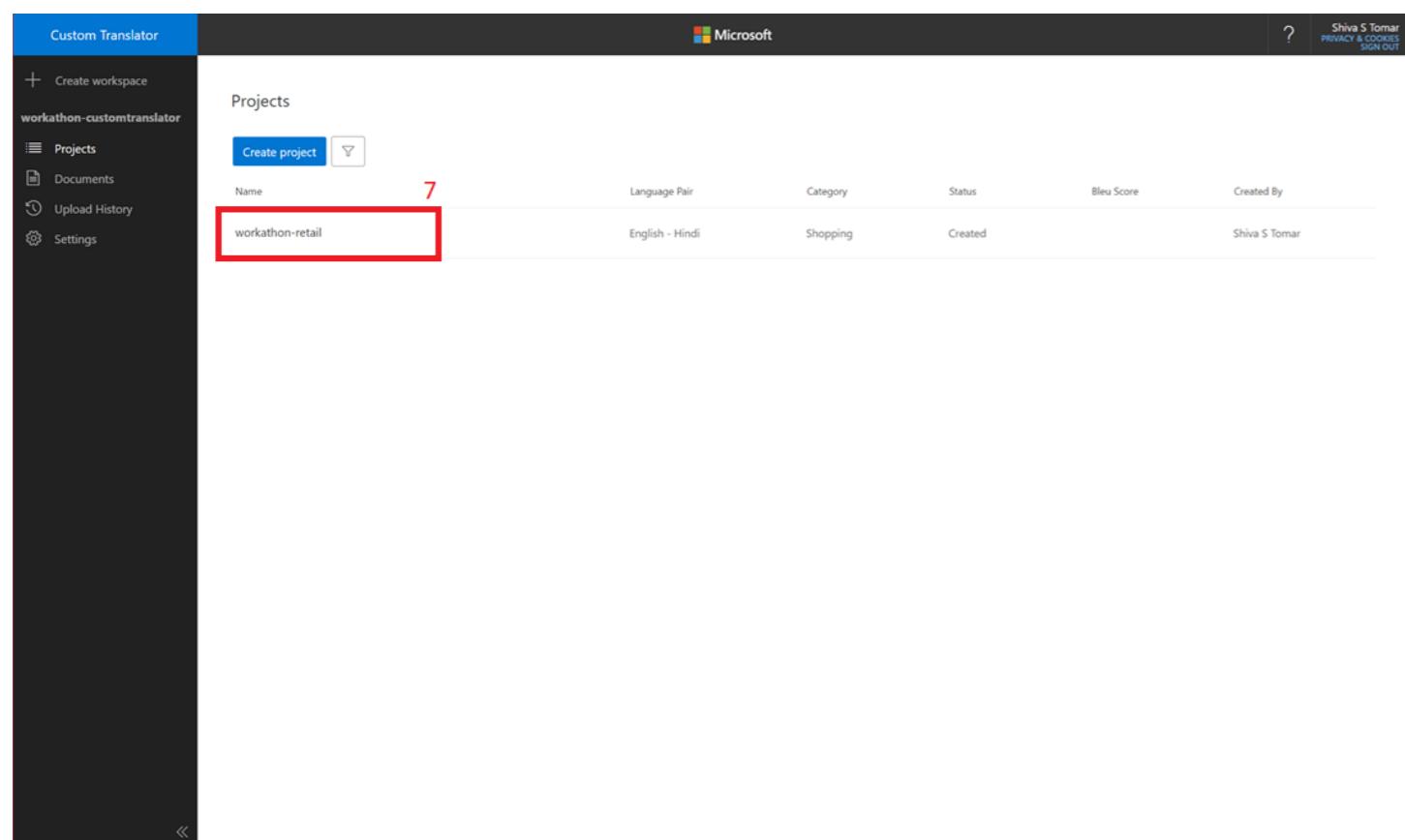
Create Project

Go to Projects & select Create Project (step 1,2)

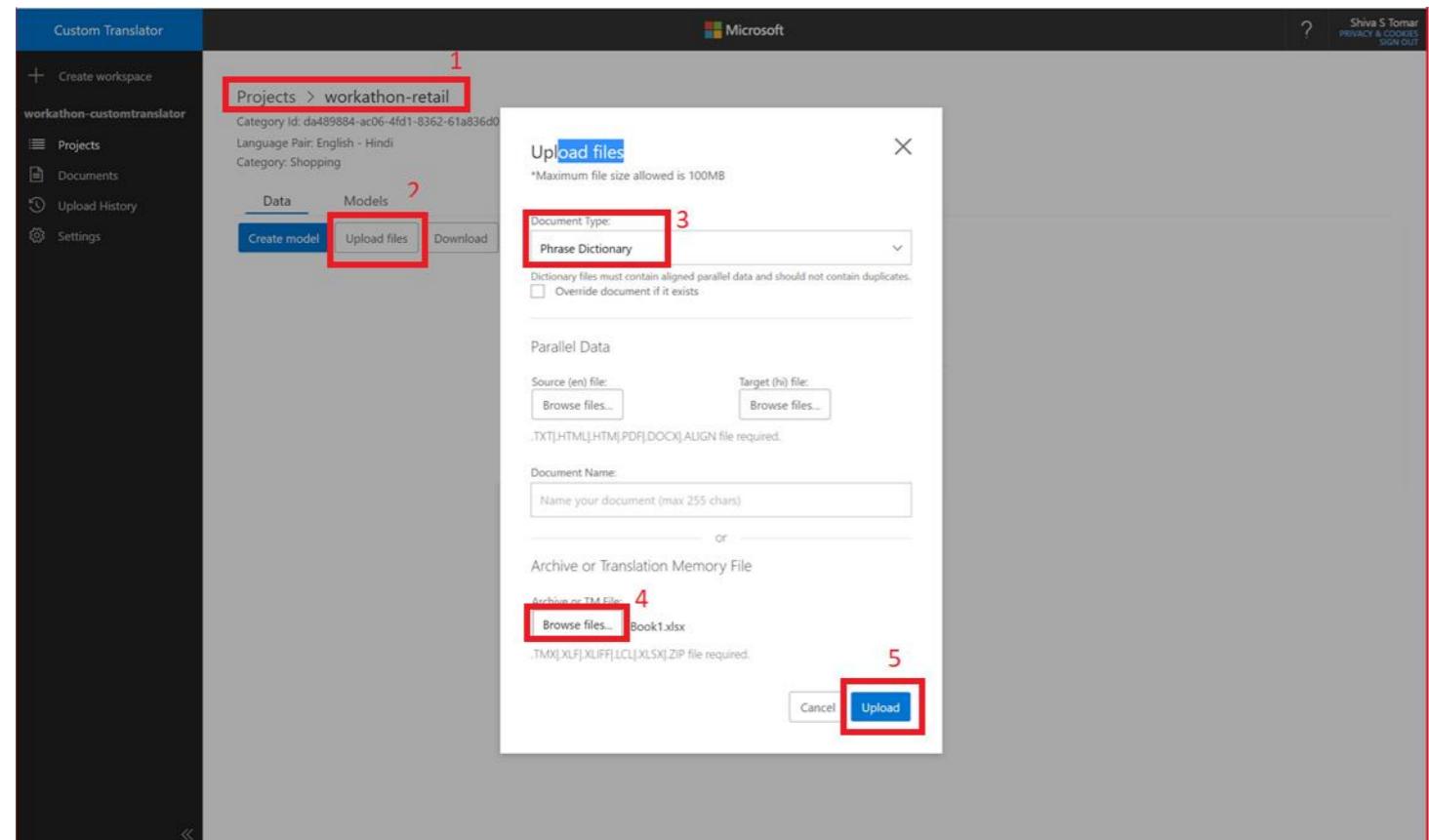


For steps 3 through 5, enter the details pertaining to the project as shown. We will be creating a custom model for English to Hindi translation.

Click Create (step 6)



Once created, click on the Project Name (step 7)



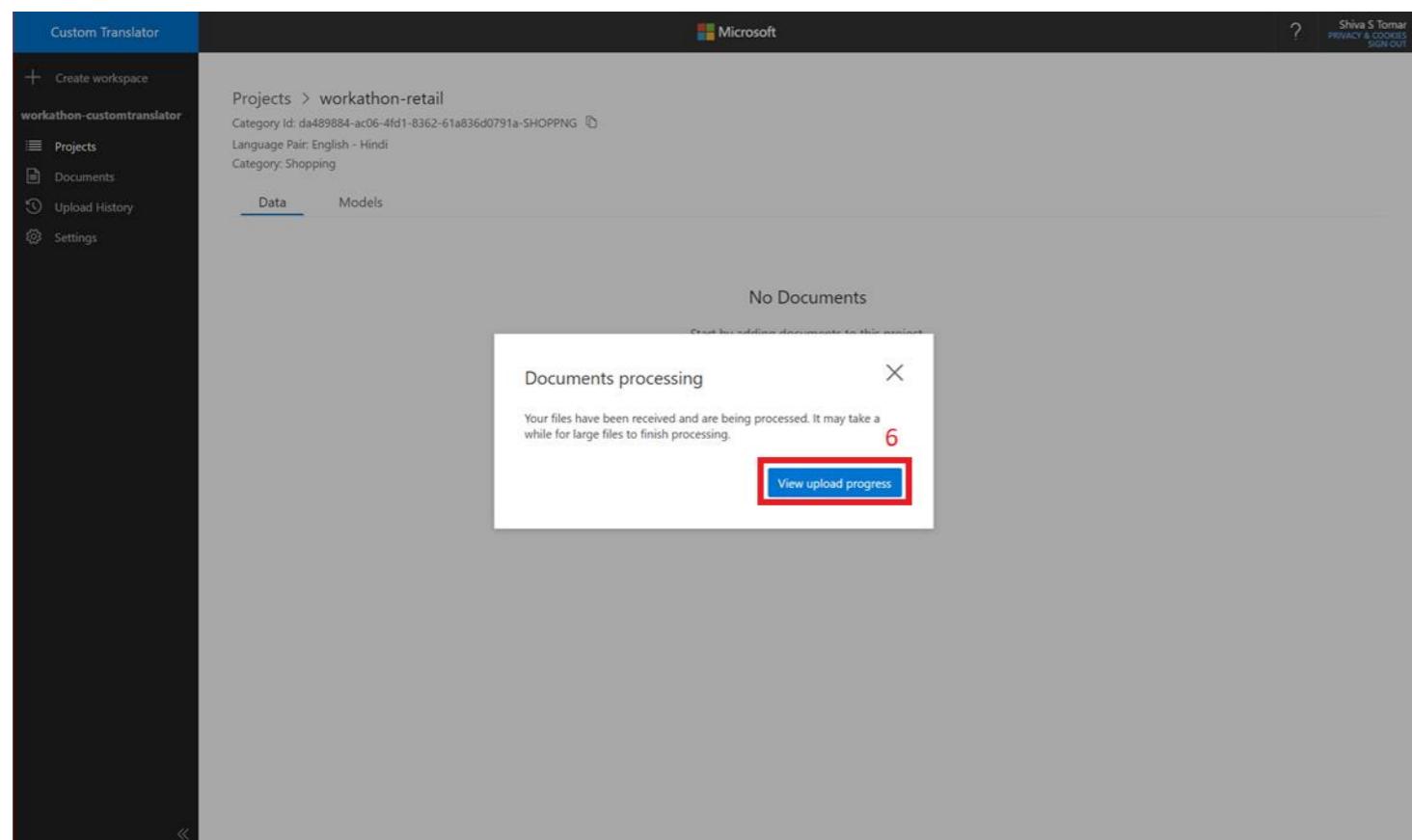
Upload training document

While in the project, click Upload Files (step 2)

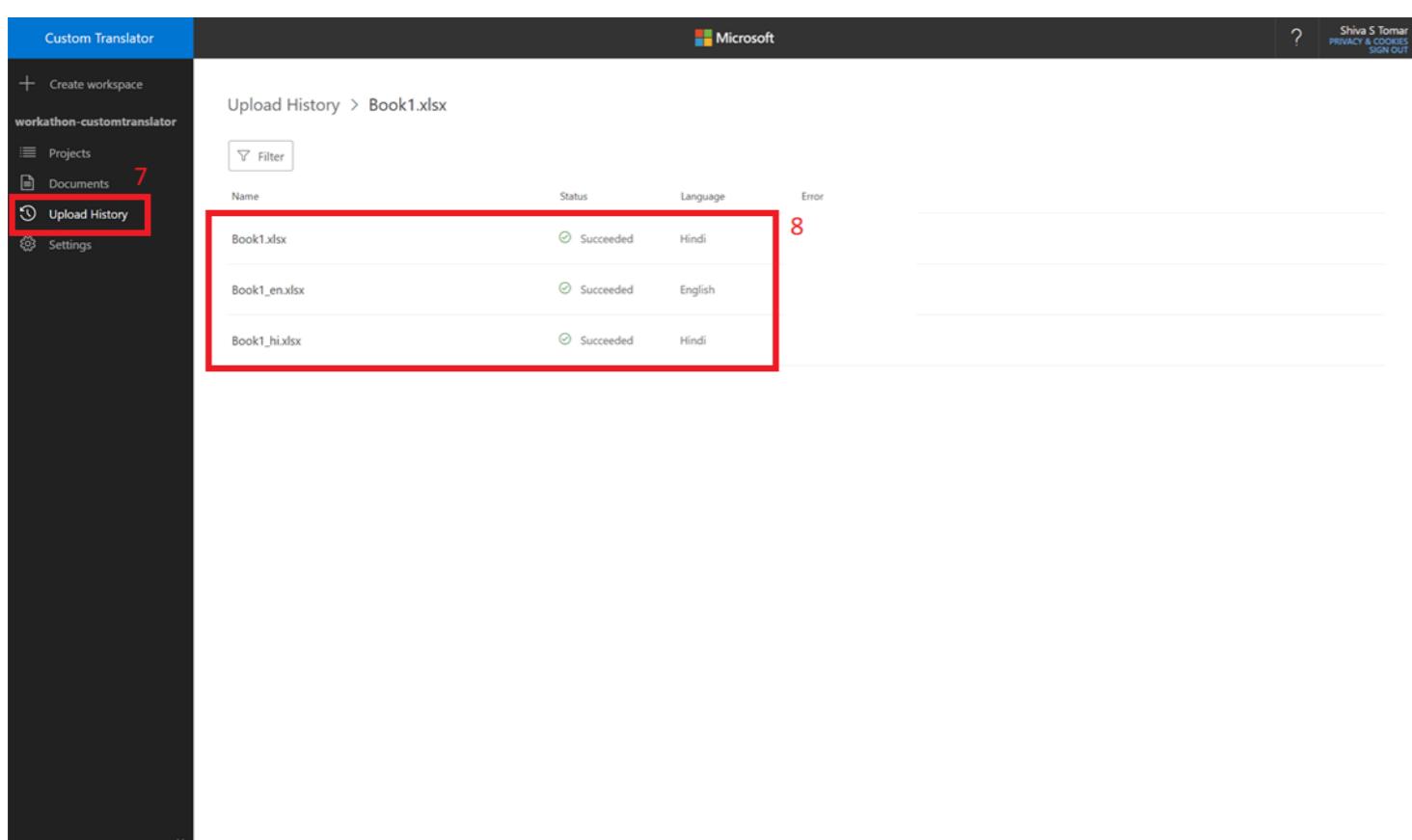
Select Phrase Dictionary from Document Type dropdown.

Browse files as shown in step 4 and select the mapped Dictionary file you downloaded in the beginning of the workshop.

Click Upload (step 5)



Click View upload progress (step 6)



Once uploaded, you can view the documents in Upload History, as shown in steps 7 & 8.

This screenshot shows the Microsoft Custom Translator interface. On the left, a sidebar lists 'Create workspace', 'workathon-customtranslator', and 'Projects'. The 'Projects' item is selected and highlighted with a red box, labeled '1'. The main content area displays a table titled 'Projects' with one entry: 'workathon-retail'. A red box highlights the project name 'workathon-retail' in the table, labeled '2'.

Train Project

Come back to the Project home page and select the project (step 1,2)

This screenshot shows the 'Models' tab within the project 'workathon-retail'. The 'Create model' button is highlighted with a red box and labeled '3'. The background shows a table with one document entry: 'Book1'.

Click create model (step 3)

This screenshot shows the 'Create model' dialog box. It contains fields for 'Model name' (set to 'workathon-custom-retail') and 'Training options' (set to 'Train immediately'). The 'Create model' button at the bottom is highlighted with a red box and labeled '6'.

Provide the Model name and select train immediately. (Step 4,5)

Click Create Model (step 6)

Custom Translator

Projects > workathon-retail

Category Id: da489884-ac06-4fd1-8362-61a836d0791a-SHOPPING

Language Pair: English - Hindi

Category: Shopping

Data Models

Name	Status	Modified Date	Bleu Score	Baseline Bleu Score	Training	Dictionary	Tuning	Test	Model Action
workathon-custom-retail	Training succeeded	2021-08-17	0	478	0	0			Deploy

Deploy Custom Model

Once training is trained, you can see the status as shown in step 2.

Now, let's deploy the model to be able to leverage it to make real time translation calls. Click Deploy (Step 3)

Custom Translator

Projects > workathon-retail

Category Id: da489884-ac06-4fd1-8362-61a836d0791a-SHOPPING

Language Pair: English - Hindi

Category: Shopping

Data Models

Name	Status	Modified Date	Bleu Score	Baseline Bleu Score	Training	Dictionary	Tuning	Test	Model Action
workathon-custom-retail	Training succeeded	2021-08-17	0	478	0	0			Deploy

Deploy or undeploy model

North America: Undeployed

Europe: Undeployed

Asia Pacific: Deployed

Cancel Save

Change the drop downs to Deployed for Regions where you want to deploy your model in (step 4)

Select save (step 5)

Custom Translator

Projects > workathon-retail

Category Id: da489884-ac06-4fd1-8362-61a836d0791a-SHOPPING

Language Pair: English - Hindi

Category: Shopping

Data Models

Name	Status	Modified Date	Bleu Score	Baseline Bleu Score	Training	Dictionary	Tuning	Test	Model Action
workathon-custom-retail	Deploying	2021-08-17	0	478	0	0			

Go to the Models tab and view the status. This will be set to deploying for a while till model deployment isn't completed (Step 6)

The screenshot shows the Microsoft Custom Translator interface. On the left, a sidebar lists 'Create workspace', 'workathon-customtranslator' (selected), 'Projects' (highlighted with a red box and number 1), 'Documents', 'Upload History', and 'Settings'. The main area is titled 'Projects' and shows a table with one row:

Name	Language Pair	Category	Status	Bleu Score	Created By
workathon-retail	English - Hindi	Shopping	Deployed		Shiva S Tomar

On the right, there is explanatory text: 'Once the Model is deployed, you can view the change in status as shown in step 3.' Below it, another screenshot shows the project page for 'workathon-retail' with the category ID highlighted with a red box and number 2.

The screenshot shows the 'workathon-retail' project page. At the top, the category ID 'Category Id: da489884-ac06-4fd1-8362-61a836d0791a-SHOPPING' is highlighted with a red box and number 3. Below this, the page displays project details: Language Pair: English - Hindi; Category: Shopping; Data tab selected; Document Type: Phrase Dictionary; Created Date: 2021-08-17; English Sentences: 478; Hindi Sentences: 478.

Once the Model is deployed, you can view the change in status as shown in step 3.

Go to the project page and copy the Category ID as shown in the below image. We will require this to make real time API calls to our custom model.

To translate text using Custom Model

We will now translate text using the Custom Model just created. Note that custom models just work with Text Translation API & not Document Translation API.

URL : `{{text-translator}}/translate`

Params :

`api-version=3.0`

`to=hi`

`includeSentenceLength=true`

`Category= <> paste the CategoryID we copied above <>`

Once you add the params, these will be appended to the URL & you will see a URL similar to the one in the screenshot.

Headers :

`Ocp-Apim-Subscription-Key : {{key}}`

`Content-Type : application/json`

`Ocp-Apim-Subscription-Region : centralindia (change this if you provisioned your resource in a different region)`

Body :

```
[{"Text": "United States, watch, computer, Fossil, CO, overseas"}]
```

Significance of input & output

- `{} : Values being picked from global variables`
- `Ocp-Apim-Subscription-Key : This is the Azure Cognitive service key, that will authenticate the request.`

`Content-Type : This refers to the input type that you provide in the body, for eg application/json allows you to enter body text in JSON format. Change the content-type on the basis of input you provide.`

`Ocp-Apim-Subscription-Region : region where your resource is deployed.`

Observe the obtained translations.

Now, try executing the call by removing the category parameter & executing the pre-built translator model and observe the difference in outcome.

Homework

1. Try the pre-built models for other languages that you're familiar with.
2. Try creating the custom model using other document types.

Additional recommended resources

Service Name	Category	Links
Translator Service	Programming Language	C#, Java, Python, Node.js, Go, PHP, Ruby
	Tiers	Free, S1, S2, S3, S4, C2, C3, C4, D3 (Difference between Tiers)
	Pricing	https://azure.microsoft.com/en-in/pricing/details/cognitive-services/translator/
	Limits	https://docs.microsoft.com/en-us/azure/cognitive-services/translator/request-limits
	Language Support	https://docs.microsoft.com/en-us/azure/cognitive-services/translator/language-support
	Sample Apps	Sample Applications
	Regional Availability & Support	https://azure.microsoft.com/en-us/global-infrastructure/services/?products=cognitive-services&regions=all
	SLAs for Cognitive Services	https://azure.microsoft.com/en-in/support/legal/sla/cognitive-services/v1_1/
	Compliance & Certificates	https://azure.microsoft.com/en-us/support/legal/cognitive-services-compliance-and-privacy/
	Cognitive Services Updates	https://azure.microsoft.com/en-us/updates/?product=cognitive-services

Security best practices

1. [Azure Cognitive Services security](#)
2. [Networking](#)
3. [Authentication](#)
4. [Key Management](#)
5. [Data loss prevention](#)
6. [Azure security baseline](#)
7. [Regulatory Compliance controls](#)

Responsible AI being a part of best practices, we encourage you to read [this](#).

[Translator Documentation](#)

[Text Translator APIs](#)

[Custom Translator APIs](#)