

Back up and recover your conversational language understanding models

When you create a Language resource in the Azure portal, you specify a region for it to be created in. From then on, your resource and all of the operations related to it take place in the specified Azure server region. It's rare, but not impossible, to encounter a network issue that hits an entire region. If your solution needs to always be available, then you should design it to either fail-over into another region. This process requires two Azure Language in Foundry Tools resources in different regions and the ability to sync your CLU models across regions.

If your app or business depends on the use of a CLU model, we recommend that you create a replica of your project into another supported region. So that if a regional outage occurs, you can then access your model in the other fail-over region where you replicated your project.

Replicating a project means that you export your project metadata and assets and import them into a new project. This action only makes a copy of your project settings, intents, entities, and utterances. You still need to [train](#) and [deploy](#) the models to be available for use with [runtime APIs](#) .

In this article, you learn to use the export and import APIs to replicate your project from one resource to another existing in different supported geographical regions. We also provide guidance for keeping your projects in sync and the changes needed to your runtime consumption.

Prerequisites

- Two Language resources in different Azure regions, each of them in a different region.

Get your resource keys endpoint

Use the following steps to get the keys and endpoint for your primary and secondary resources.

Go to your resource overview page in the [Azure portal](#) . From the menu on the left side, select **Keys and Endpoint**. You use the endpoint and key for the API requests

💡 Tip

Keep a note of keys and endpoints for both primary and secondary resources. Use these values to replace the following placeholders: {PRIMARY-ENDPOINT}, {PRIMARY-RESOURCE-KEY}, {SECONDARY-ENDPOINT}, and {SECONDARY-RESOURCE-KEY}. Also take note of your project name, your model name, and your deployment name. Use these values to replace the following placeholders: {PROJECT-NAME}, {MODEL-NAME}, and {DEPLOYMENT-NAME}.

Export your primary project assets

Start by exporting the project assets from the project in your primary resource.

Submit export job

Replace the placeholders in the following request with your {PRIMARY-ENDPOINT} and {PRIMARY-RESOURCE-KEY} that you obtained in the first step.

Create a POST request by using the following URL, headers, and JSON body to export your project.

Request URL

Use the following URL when you create your API request. Replace the placeholder values with your own values.

rest

```
{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/:export?stringIndexType=Utf16CodeUnit&api-version={API-VERSION}
```

[Expand table](#)

Placeholder	Value	Example
{ENDPOINT}	The endpoint for authenticating your API request.	https://<your-custom-subdomain>.cognitiveservices.azure.com
{PROJECT-NAME}	The name for your project. This value is case sensitive.	EmailApp
{API-VERSION}	The version of the API that you're calling.	2023-04-01

Headers

Use the following header to authenticate your request.

[Expand table](#)

Key	Value
Ocp-Apim-Subscription-Key	The key to your resource. Used for authenticating your API requests.

After you send your API request, you receive a 202 response that indicates success. In the response headers, extract the operation-location value. The value is formatted like this example:

rest

```
{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/jobs/{JOB-ID}?api-version={API-VERSION}
```

JOB-ID is used to identify your request because this operation is asynchronous. Use this URL to get the exported project JSON by using the same authentication method.


Get export job status

Replace the placeholders in the following request with your {PRIMARY-ENDPOINT} and {PRIMARY-RESOURCE-KEY} that you obtained in the first step.

Use the following **GET** request to query the status of your export job. You can use the URL you received from the previous step, or replace the placeholder values with your own values.

rest


{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/export/jobs/{JOB-ID}?api-version={API-VERSION}

 Expand table

Placeholder	Value	Example
{ENDPOINT}	The endpoint for authenticating your API request.	https://<your-custom-subdomain>.cognitiveservices.azure.com
{PROJECT-NAME}	The name for your project. This value is case-sensitive.	myProject
{JOB-ID}	The ID for locating your export job status. It's in the <code>location</code> header value you received in the previous step.	xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx
{API-VERSION}	The version of the API you're calling.	2023-04-01

Headers

Use the following header to authenticate your request.

 Expand table

Key	Description	Value
Ocp-Apim-Subscription-Key	The key to your resource. Used for authenticating your API requests.	{YOUR-PRIMARY-RESOURCE-KEY}

Response body

JSON

```
{
  "resultUrl": "{Endpoint}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/export/jobs/xxxxxx-xxxxx-xxxxx-xx/result?api-version={API-VERSION}",
  "jobId": "xxxx-xxxxx-xxxxx-xxx",
  "createdDateTime": "2022-04-18T15:23:07Z",
  "lastUpdatedDateTime": "2022-04-18T15:23:08Z",
  "expirationDateTime": "2022-04-25T15:23:07Z",
  "status": "succeeded"
}
```

Use the url from the `resultUrl` key in the body and view the exported assets from this job.

Get export results

Submit a **GET** request using the `{RESULT-URL}` you received from the previous step to view the results of the export job.

Headers

Use the following header to authenticate your request.

 Expand table

Key	Description	Value
Ocp-Apim-Subscription-Key	The key to your resource. Used for authenticating your API requests.	{PRIMARY-RESOURCE-KEY}

Copy the response body to use as the body for the next import job.

Import to a new project

Now go ahead and import the exported project assets in your new project in the secondary region so you can replicate it.

Submit import job

Replace the placeholders in the following request with your {SECONDARY-ENDPOINT}, and {SECONDARY-RESOURCE-KEY} that you obtained in the first step.

Submit a `POST` request by using the following URL, headers, and JSON body to import your project.

Request URL

Use the following URL when you create your API request. Replace the placeholder values with your own values.

```
rest
```

```
{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/:import?  
api-version={API-VERSION}
```

[Expand table](#)

Placeholder	Value	Example
{ENDPOINT}	The endpoint for authenticating your API request.	https://<your-custom-subdomain>.cognitiveservices.azure.com
{PROJECT-NAME}	The name for your project. This value is case sensitive and must match the project name in the JSON file that you're importing.	EmailAppDemo
{API-VERSION}	The version of the API that you're calling.	2023-04-01

Headers

Use the following header to authenticate your request.

[Expand table](#)

Key	Value
Ocp-Apim-Subscription-Key	The key to your resource. Used for authenticating your API requests.

Body

The JSON body you send is similar to the following example. For more information about the JSON object, see the [reference documentation](#).

JSON

```
{
  "projectFileVersion": "{API-VERSION}",
  "stringIndexType": "Utf16CodeUnit",
  "metadata": {
    "projectKind": "Conversation",
    "settings": {
      "confidenceThreshold": 0.7
    },
    "projectName": "{PROJECT-NAME}",
    "multilingual": true,
    "description": "Trying out CLU",
    "language": "{LANGUAGE-CODE}"
  },
  "assets": {
    "projectKind": "Conversation",
    "intents": [
      {
        "category": "intent1"
      },
      {
        "category": "intent2"
      }
    ],
    "entities": [
      {
        "category": "entity1"
      }
    ],
    "utterances": [
      {
        "text": "text1",
        "dataset": "{DATASET}",
        "intent": "intent1",
        "entities": [
          {
            "category": "entity1",
            "offset": 5,
            "length": 5
          }
        ]
      },
      {
        "text": "text2",
        "language": "{LANGUAGE-CODE}",
        "dataset": "{DATASET}",

```

```
"intent": "intent2",  
"entities": []  
}  
]  
}  
}
```

[Expand table](#)

Key	Placeholder	Value	Example
{API-VERSION}	The version of the API that you're calling.	2023-04-01	
projectName	{PROJECT-NAME}	The name of your project. This value is case sensitive.	EmailAppDemo
language	{LANGUAGE-CODE}	A string that specifies the language code for the utterances used in your project. If your project is a multilingual project, choose the language code of most of the utterances.	en-us
multilingual	true	A Boolean value that enables you to have documents in multiple languages in your dataset. When your model is deployed, you can query the model in any supported language , including languages that aren't included in your training documents.	true
dataset	{DATASET}	For information on how to split your data between a testing and training set, see Label your utterances in Foundry . Possible values for this field are <code>Train</code> and <code>Test</code> .	Train

After a successful request, the API response contains an `operation-location` header with a URL that you can use to check the status of the import job. The header is formatted like this example:

HTTP

```
{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/import/jobs/{JOB-ID}?api-version={API-VERSION}
```

Get import job status

Replace the placeholders in the following request with your {SECONDARY-ENDPOINT}, and {SECONDARY-RESOURCE-KEY} that you obtained in the first step.

When you send a successful project import request, the full request URL for checking the import job's status (including your endpoint, project name, and job ID) is contained in the response's operation-location header.

Use the following **GET** request to query the status of your import job. You can use the URL you received from the previous step, or replace the placeholder values with your own values.

rest

```
{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/import/jobs/{JOB-ID}?api-version={API-VERSION}
```

[Expand table](#)

Placeholder	Value	Example
{ENDPOINT}	The endpoint for authenticating your API request.	https://<your-custom-subdomain>.cognitiveservices.azure.com
{PROJECT-NAME}	The name for your project. This value is case-sensitive.	myProject
{JOB-ID}	The ID for locating your import job status.	xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx
{API-VERSION}	The version of the API you're calling.	2023-04-01

Headers

Use the following header to authenticate your request.

[Expand table](#)

Key	Description	Value
Ocp-Apim-Subscription-Key	The key to your resource. Used for authenticating your API requests.	{YOUR-PRIMARY-RESOURCE-KEY}

Response body

Once you send the request, you get the following response. Keep polling this endpoint until the status parameter changes to "succeeded."

JSON

```
{
  "jobId": "xxxxxx-xxxxxx-xxxx-xxxxxx",
  "createdDateTime": "2022-04-18T15:17:20Z",
  "lastUpdatedDateTime": "2022-04-18T15:17:22Z",
  "expirationDateTime": "2022-04-25T15:17:20Z",
  "status": "succeeded"
}
```

Train your model

After importing your project, you only have copies of the project's assets and metadata. You still need to train your model, which incurs usage on your account.

Submit training job

Replace the placeholders in the following request with your {SECONDARY-ENDPOINT}, and {SECONDARY-RESOURCE-KEY} that you obtained in the first step.

Create a **POST** request using the following URL, headers, and JSON body to submit a training job.

Request URL

Use the following URL when creating your API request. Replace the placeholder values with your own values.

rest


```
{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/:train?api-version={API-VERSION}
```

 Expand table

Placeholder	Value	Example
{ENDPOINT}	The endpoint for authenticating your API request.	https://<your-custom-subdomain>.cognitiveservices.azure.com
{PROJECT-NAME}	The name for your project. This value is case-sensitive.	EmailApp
{API-VERSION}	The version of the API you're calling.	2023-04-01

Headers

Use the following header to authenticate your request.

 Expand table

Key	Value
Ocp-Apim-Subscription-Key	The key to your resource. Used for authenticating your API requests.

Request body

Use the following object in your request. The model will be named after the value you use for the `modelLabel` parameter once training is complete.

JSON

```
{
  "modelLabel": "{MODEL-NAME}",
  "trainingMode": "{TRAINING-MODE}",
  "trainingConfigVersion": "{CONFIG-VERSION}",
  "evaluationOptions": {
    "kind": "percentage",
    "testingSplitPercentage": 20,
    "trainingSplitPercentage": 80
  }
}
```

 Expand table

Key	Placeholder	Value	Example
modelLabel	{MODEL-NAME}	Your Model name.	Model1
trainingConfigVersion	{CONFIG-VERSION}	The training configuration model version. By default, the latest model version is used.	2022-05-01
trainingMode	{TRAINING-MODE}	The training mode to be used for training. Supported modes are Standard training , faster training, but only available for English and Advanced training supported for other languages and multilingual projects, but involves longer training times. Learn more about training modes .	standard
kind	percentage	Split methods. Possible Values are <code>percentage</code> or <code>manual</code> . See how to train a model for more information.	percentage
trainingSplitPercentage	80	Percentage of your tagged data to be included in the training set. Recommended value is 80.	80
testingSplitPercentage	20	Percentage of your tagged data to be included in the testing set. Recommended value is 20.	20

ⓘ Note

The `trainingSplitPercentage` and `testingSplitPercentage` are only required if `kind` is set to `percentage` and the sum of both percentages should be equal to 100.

Once you send your API request, you receive a `202` response indicating success. In the response headers, extract the `operation-location` value formatted like this:

```
rest
```

```
{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/train/jobs/{JOB-ID}?api-version={API-VERSION}
```

You can use this URL to get the training job status.

Get Training Status

Replace the placeholders in the following request with your {SECONDARY-ENDPOINT}, and {SECONDARY-RESOURCE-KEY} that you obtained in the first step.


When you send a successful training request, the full request URL for checking the job's status (including your endpoint, project name, and job ID) is contained in the response's operation-location header.

Use the following **GET** request to get the status of your model's training progress. Replace the placeholder values with your own values.

Request URL

rest


{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/train/jobs/{JOB-ID}?api-version={API-VERSION}

 Expand table

Placeholder	Value	Example
{YOUR-ENDPOINT}	The endpoint for authenticating your API request.	https://<your-custom-subdomain>.cognitiveservices.azure.com
{PROJECT-NAME}	The name for your project. This value is case-sensitive.	EmailApp
{JOB-ID}	The ID for locating your model's training status.	xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx
{API-VERSION}	The version of the API you're calling.	2023-04-01

Headers

Use the following header to authenticate your request.

 Expand table

Key	Value
Ocp-Api-Subscription-Key	The key to your resource. Used for authenticating your API requests.

Response Body

Once you send the request, you get the following response. Keep polling this endpoint until the **status** parameter changes to "succeeded".

JSON

```
{
  "result": {
    "modelLabel": "{MODEL-LABEL}",
    "trainingConfigVersion": "{TRAINING-CONFIG-VERSION}",
    "trainingMode": "{TRAINING-MODE}",
    "estimatedEndDateTime": "2022-04-18T15:47:58.8190649Z",
    "trainingStatus": {
      "percentComplete": 3,
      "startDateTime": "2022-04-18T15:45:06.8190649Z",
      "status": "running"
    },
    "evaluationStatus": {
      "percentComplete": 0,
      "status": "notStarted"
    }
  },
  "jobId": "xxxxx-xxxxx-xxxx-xxxxx-xxxx",
  "createdDateTime": "2022-04-18T15:44:44Z",
  "lastUpdatedDateTime": "2022-04-18T15:45:48Z",
  "expirationDateTime": "2022-04-25T15:44:44Z",
  "status": "running"
}
```

[Expand table](#)

Key	Value	Example
modelLabel	The model name	Model1
trainingConfigVersion	The training configuration version. By default, the latest version is used.	2022-05-01
trainingMode	Your selected training mode .	standard

Key	Value	Example
startDateTime	The time training started	2022-04-14T10:23:04.2598544Z
status	The status of the training job	running
estimatedEndDateTime	Estimated time for the training job to finish	2022-04-14T10:29:38.2598544Z
jobId	Your training job ID	xxxxx-xxxx-xxxx-xxxx-xxxxxxxx
createdDateTime	Training job creation date and time	2022-04-14T10:22:42Z
lastUpdatedDateTime	Training job last updated date and time	2022-04-14T10:23:45Z
expirationDateTime	Training job expiration date and time	2022-04-14T10:22:42Z

Deploy your model

This step is where you make your trained model available form consumption via the [runtime prediction API](#) .

Tip

Use the same deployment name as your primary project for easier maintenance and minimal changes to your system to handle redirecting your traffic.

Submit deployment job

Replace the placeholders in the following request with your {SECONDARY-ENDPOINT} , and {SECONDARY-RESOURCE-KEY} that you obtained in the first step.

Create a **PUT** request using the following URL, headers, and JSON body to start deploying a conversational language understanding model.

Request URL

rest

```
{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/deployments/{DEPLOYMENT-NAME}?api-version={API-VERSION}
```

[Expand table](#)

Placeholder	Value	Example
{ENDPOINT}	The endpoint for authenticating your API request.	https://<your-custom-subdomain>.cognitiveservices.azure.com
{PROJECT-NAME}	The name for your project. This value is case-sensitive.	myProject
{DEPLOYMENT-NAME}	The name for your deployment. This value is case-sensitive.	staging
{API-VERSION}	The version of the API you're calling.	2023-04-01

Headers

Use the following header to authenticate your request.

[Expand table](#)

Key	Value
Ocp-Apim-Subscription-Key	The key to your resource. Used for authenticating your API requests.

Request Body

JSON

```
{  
  "trainedModelLabel": "{MODEL-NAME}",  
}
```

[Expand table](#)

Key	Placeholder	Value	Example
trainedModelLabel	{MODEL-NAME}	The model name that is assigned to your deployment. You can only assign successfully trained models. This value is case-sensitive.	myModel

Once you send your API request, you receive a 202 response indicating success. In the response headers, extract the `operation-location` value formatted like this:

```
rest
```

```
{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/deployments/{DEPLOYMENT-NAME}/jobs/{JOB-ID}?api-version={API-VERSION}
```

You can use this URL to get the deployment job status.

Get the deployment status

Replace the placeholders in the following request with your `{SECONDARY-ENDPOINT}`, and `{SECONDARY-RESOURCE-KEY}` that you obtained in the first step.


When you send a successful deployment request, the full request URL for checking the job's status (including your endpoint, project name, and job ID) is contained in the response's `operation-location` header.

Use the following **GET** request to get the status of your deployment job. Replace the placeholder values with your own values.

Request URL

```
rest
```

```
{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}/deployments/{DEPLOYMENT-NAME}/jobs/{JOB-ID}?api-version={API-VERSION}
```

 Expand table

Placeholder	Value	Example
{ENDPOINT}	The endpoint for authenticating your API request.	<code>https://<your-custom-subdomain>.cognitiveservices.azure.com</code>

Placeholder	Value	Example
{PROJECT-NAME}	The name for your project. This value is case-sensitive.	myProject
{DEPLOYMENT-NAME}	The name for your deployment. This value is case-sensitive.	staging
{JOB-ID}	The ID for locating your model's training status.	xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx
{API-VERSION}	The version of the API you're calling.	2023-04-01

Headers

Use the following header to authenticate your request.

[Expand table](#)

Key	Value
Ocp-Apim-Subscription-Key	The key to your resource. Used for authenticating your API requests.

Response Body

Once you send the request, you'll get the following response. Keep polling this endpoint until the **status** parameter changes to "succeeded".

JSON

```
{
  "jobId": "{JOB-ID}",
  "createdDateTime": "{CREATED-TIME}",
  "lastUpdatedDateTime": "{UPDATED-TIME}",
  "expirationDateTime": "{EXPIRATION-TIME}",
  "status": "running"
}
```

Changes in calling the runtime

Within your system, at the step where you call [runtime API](#) check for the response code returned from the submitted task API. If you observe a **consistent** failure in submitting the request, it could indicate an outage in your primary region. Failure once doesn't mean an outage. It may also be a transient issue. Retry submitting the job through the secondary resource you created. For the second request use your {YOUR-SECONDARY-ENDPOINT} and secondary key, if you followed the previous steps, {PROJECT-NAME} and {DEPLOYMENT-NAME} are the same, so no changes are required to the request body.

In case you revert to using your secondary resource, you may observe a slight increase in latency because of the difference in regions where your model is deployed.

Check if your projects are out of sync

Maintaining the freshness of both projects is an important part of process. You need to frequently check if any updates were made to your primary project so that you move them over to your secondary project. This way if your primary region fail and you move into the secondary region you should expect similar model performance since it already contains the latest updates. Setting the frequency of checking if your projects are in sync is an important choice, we recommend that you do this check daily in order to guarantee the freshness of data in your secondary model.

Get project details

Use the following url to get your project details, one of the keys returned in the body indicates the last modified date of the project. Repeat the following step twice, one for your primary project and another for your secondary project. Then compare the timestamp returned for both to check if they're in or out of sync.

Use the following **GET** request to get your project details. You can use the URL you received from the previous step, or replace the placeholder values with your own values.

```
rest
```

```
{ENDPOINT}/language/authoring/analyze-conversations/projects/{PROJECT-NAME}?api-version={API-VERSION}
```

 Expand table

Placeholder	Value	Example
{ENDPOINT}	The endpoint for authenticating your API request.	https://<your-custom-subdomain>.cognitiveservices.azure.com
{PROJECT-NAME}	The name for your project. This value is case-sensitive.	myProject
{API-VERSION}	The version of the API you're calling.	2023-04-01

Headers

Use the following header to authenticate your request.

 Expand table

Key	Description	Value
Ocp-Apim-Subscription-Key	The key to your resource. Used for authenticating your API requests.	{YOUR-PRIMARY-RESOURCE-KEY}

Response body

JSON

```
{
  "createdDateTime": "2022-04-18T13:53:03Z",
  "lastModifiedDateTime": "2022-04-18T13:53:03Z",
  "lastTrainedDateTime": "2022-04-18T14:14:28Z",
  "lastDeployedDateTime": "2022-04-18T14:49:01Z",
  "projectKind": "Conversation",
  "projectName": "{PROJECT-NAME}",
  "multilingual": true,
  "description": "This is a sample conversation project.",
  "language": "{LANGUAGE-CODE}"
}
```

Repeat the same steps for your replicated project using {SECONDARY-ENDPOINT}, and {SECONDARY-RESOURCE-KEY}. Compare the returned lastModifiedDateTime from both projects. If your primary project was modified sooner than your secondary one, you need to repeat the steps of [exporting](#), [importing](#), [training](#), and [deploying](#) your model.

Next steps

In this article, you learned how to use the export and import APIs to replicate your project to a secondary Language resource in other region. Next, explore the API reference docs to see what else you can do with authoring APIs.

- [Authoring REST API reference](#)
- [Runtime prediction REST API reference](#)

Last updated on 11/18/2025