

PUBLIC

Detect and Highlight entities from unstructured text using machine learning

1162

Exercises / Solutions

Komal Narsinghani/ SAP Labs India

1 GETTING ON-BOARDED TO BER	3
1.1 SETTING UP TRIAL ACCOUNT	3
1.2 CREATING BER SERVICE INSTANCE	3
1.3 GET OAUTH ACCESS TOKEN FOR BER	3
1.4 ACCESS AND AUTHORIZE BER	4
2 BER TRAINING DATASET	6
2.1 CREATE DATASET	6
2.2 UNDERSTANDING THE TRAINING DOCUMENT.....	8
2.3 UPLOAD DOCUMENTS.....	9
3 BER TRAINING	11
3.1 TRIGGER TRAINING JOB	11
3.2 GET TRAINING JOB STATUS	13
3.3 DELETE TRAINING JOB	14
4 BER MODELS	16
4.1 GET ALL MODELS.....	16
4.2 GET MODEL DETAILS BY MODEL NAME.....	18
4.3 GET MODEL DETAILS BY MODEL NAME AND MODEL VERSION	21
5 BER INFERENCE	22
5.1 INFERENCE WITH SAP_EMAIL_BUSINESS_ENTITY.....	22
5.1.1 Submit the inference job for sap_email_business_entity model	22
5.1.2 Get the inference results for sap_email_business_entity model	24
5.2 INFERENCE WITH SAP_INVOICE_HEADER	26
5.2.1 Submit the inference job for sap_invoice_header model	26
5.2.2 Get the inference results for sap_invoice_header model	29

1 GETTING ON-BOARDED TO BER

Objective

In this exercise you would be able to create a trial account and subscribe to the BER service.

1.1 Setting up trial account

Follow the steps documented here: <https://developers.sap.com/tutorials/hcp-create-trial-account.html>

1.2 Creating BER service instance

Follow the steps documented here: <https://developers.sap.com/tutorials/cp-aibus-ber-service-instance.html>

1.3 Get OAuth access token for BER

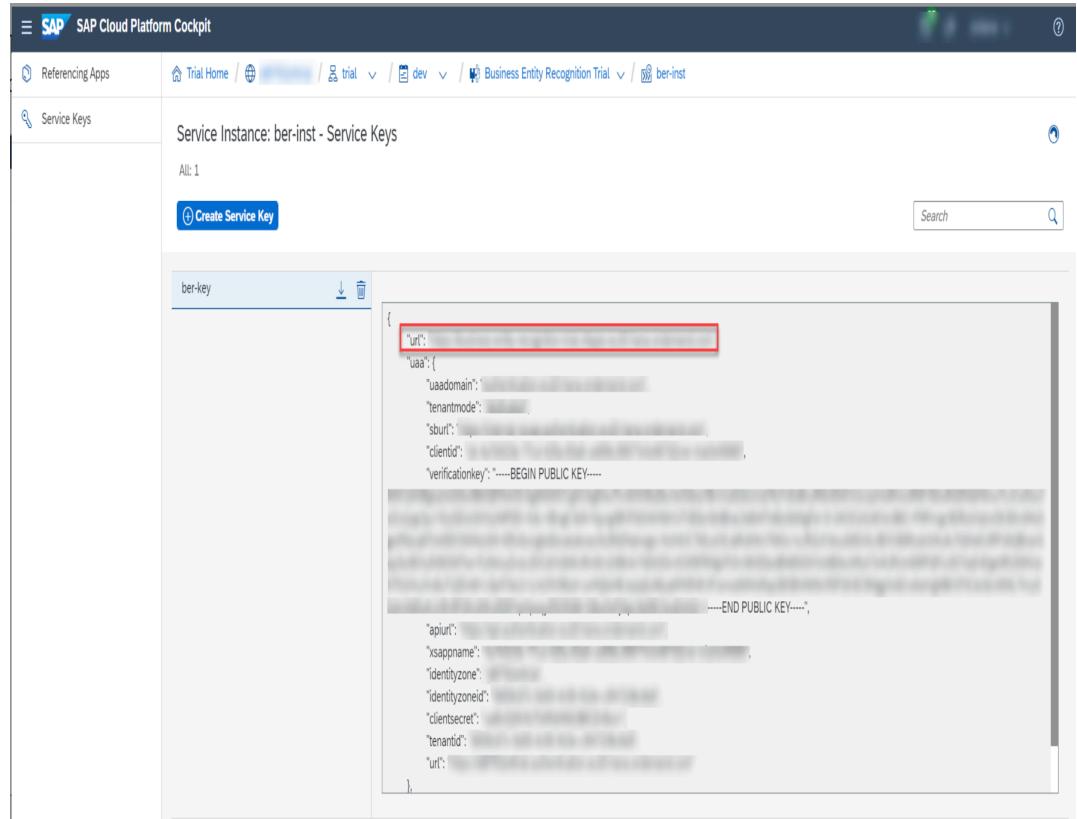
Follow the steps documented here: <https://developers.sap.com/tutorials/cp-aibus-ber-web-oauth-token.html>

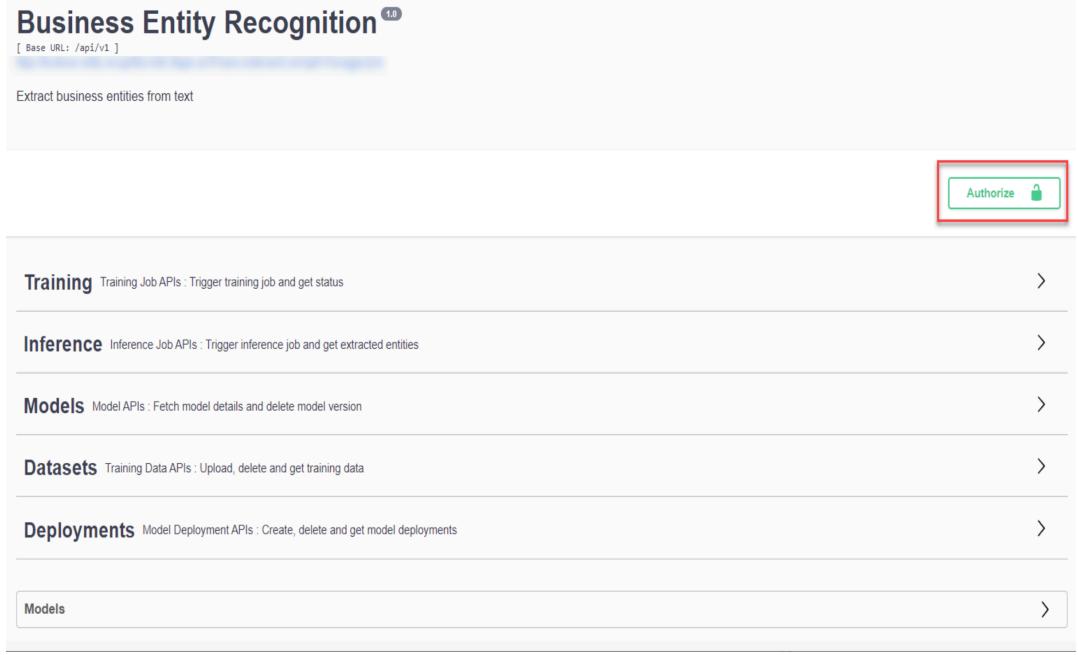
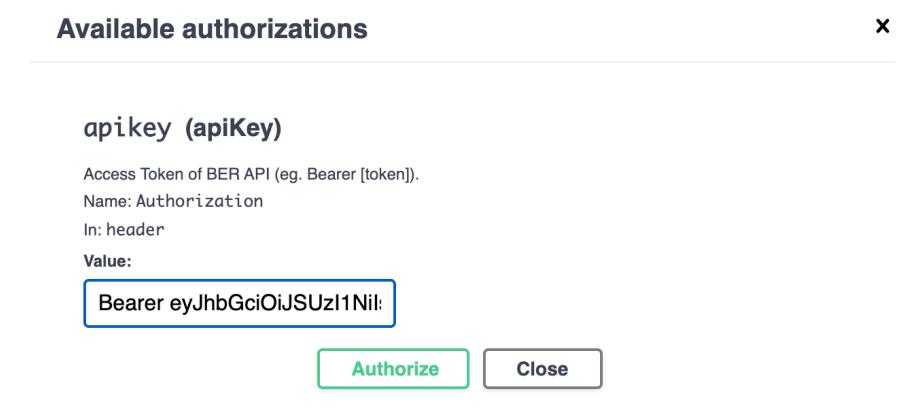
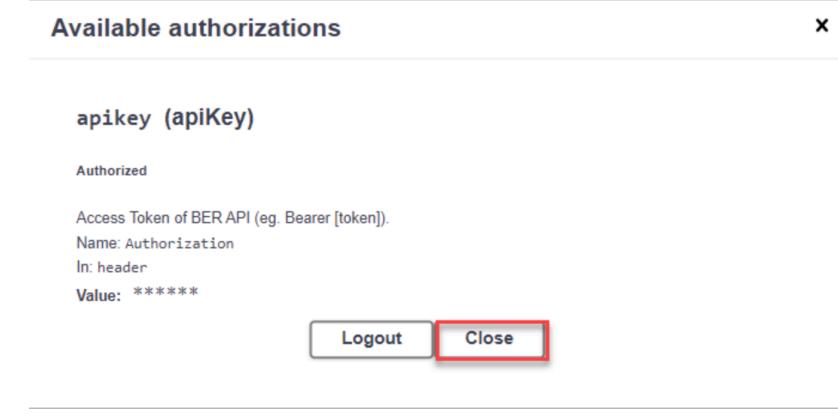
1.4 Access and Authorize BER

In this exercise we would access the Swagger UI of BER service and authorize it for subsequent exercises.

Pre-requisites:

- Make sure that you have completed sections 1.1, 1.2 and 1.3
- Keep the service key generated in section 1.2 handy for this exercise

Explanation	Screenshot
<ol style="list-style-type: none"> 1. Access the Business Entity Recognition Swagger UI. Add <code>/api/v1</code> to the url value (noted in section 1.2). 2. Paste this into any web browser and press Enter. 	 <p>The screenshot shows the SAP Cloud Platform Cockpit interface. In the top navigation bar, under 'Referencing Apps', the path is Trial Home / Trial / dev / Business Entity Recognition Trial / ber-inst. On the left sidebar, there's a 'Service Keys' section. Below it, a table lists 'Service Instance: ber-inst - Service Keys' with one entry: 'All: 1'. A blue button labeled 'Create Service Key' is visible. The main content area displays a JSON configuration for a service key named 'ber-key'. A red box highlights the 'url' field, which contains a placeholder URL. The JSON code is as follows:</p> <pre> { "url": "REDACTED", "uaa": { "uaadomain": "REDACTED", "tenantmode": "REDACTED", "sburl": "REDACTED", "clientid": "REDACTED", "verificationkey": "-----BEGIN PUBLIC KEY-----\nREDACTED\n-----END PUBLIC KEY-----", "apiurl": "REDACTED", "xsappname": "REDACTED", "identityzone": "REDACTED", "identityzoneid": "REDACTED", "clientsecret": "REDACTED", "tenantid": "REDACTED", "url": "REDACTED" } } </pre> <p>At the bottom of the page, there are links for 'Useful Links' and 'Legal Information'.</p>

Explanation	Screenshot
<p>3. Click Authorize in the top-right corner of the Swagger UI.</p>	 <p>The screenshot shows the 'Business Entity Recognition' API documentation. At the top right, there is a green 'Authorize' button with a lock icon. Below it, there are several sections: 'Training', 'Inference', 'Models', 'Datasets', and 'Deployments'. Each section has a small description and a right-pointing arrow. The 'Models' section is expanded, showing a 'Models' sub-section with its own right-pointing arrow.</p>
<p>4. Get the value of access_token created in step 1.3, then add Bearer in front of it, and enter it in the Value field. The format is: Bearer <access_token></p>	 <p>The screenshot shows a modal dialog titled 'Available authorizations'. It contains one entry: 'apikey (apiKey)'. The 'Value' field is filled with 'Bearer eyJhbGciOiJSUzI1NiI...'. Below the table are two buttons: 'Authorize' (highlighted with a red box) and 'Close'.</p>
<p>5. Click Authorize and then click Close</p>	 <p>The screenshot shows the same 'Available authorizations' dialog. The 'Value' field now contains '*****'. The 'Logout' and 'Close' buttons are visible at the bottom, with 'Close' highlighted with a red box.</p>

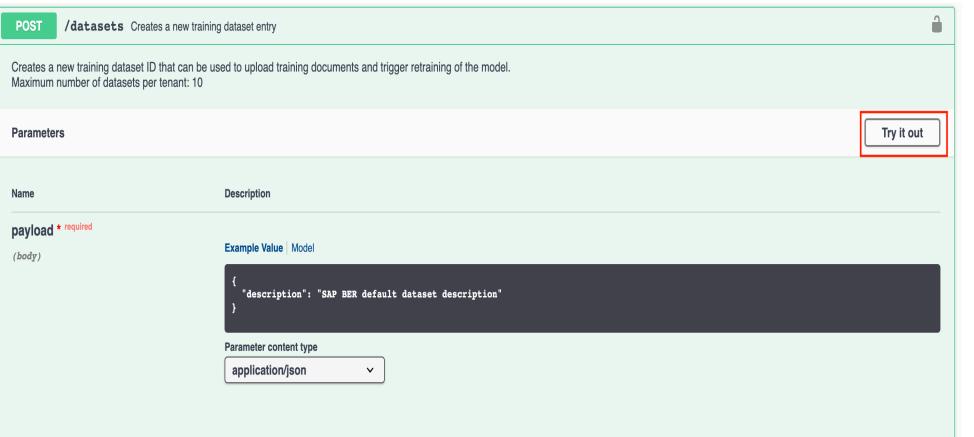
2 BER TRAINING DATASET

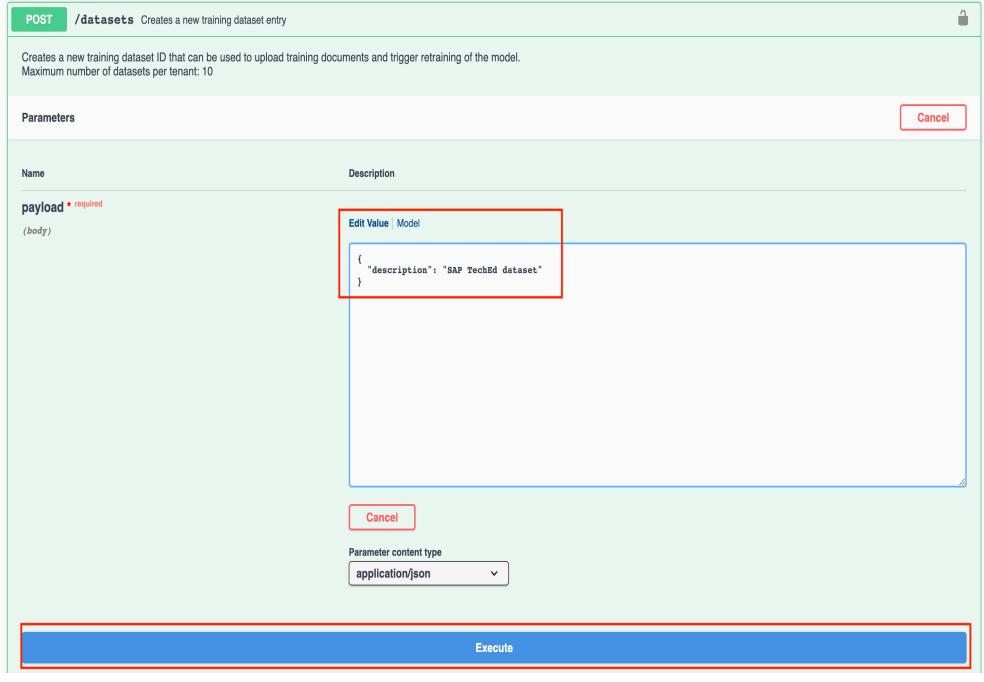
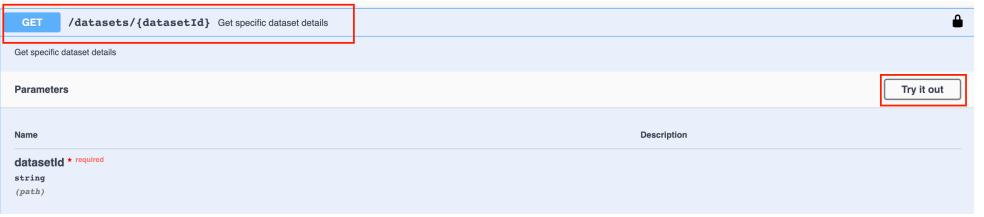
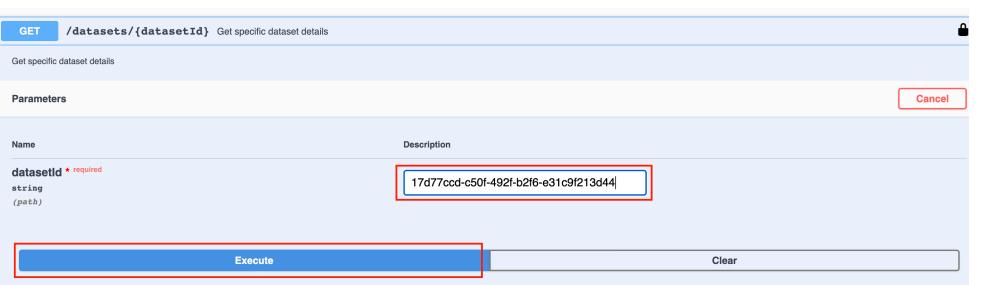
Objective

In this exercise, you will learn how to prepare the dataset for training a custom model.

2.1 Create Dataset

Follow these steps to create a dataset which would be used for the training of a custom model using BER.

Explanation	Screenshot				
<p>1. Click the endpoint POST /datasets, under Datasets to expand it.</p>	 <p>Datasets Training Data APIs : Upload, delete and get training data</p> <ul style="list-style-type: none"> GET /datasets Retrieve all datasets POST /datasets Creates a new training dataset entry DELETE /datasets/{datasetId} Deleting a dataset and its contents GET /datasets/{datasetId} Get specific dataset details GET /datasets/{datasetId}/documents Get documents present in a dataset POST /datasets/{datasetId}/documents Upload a training document DELETE /datasets/{datasetId}/documents/{documentId} Deletes specific training document of the given dataset ID GET /datasets/{datasetId}/documents/{documentId} Retrieve specific training document detail of the given dataset ID 				
<p>2. Click on Try it out</p>	 <p>POST /datasets Creates a new training dataset entry</p> <p>Creates a new training dataset ID that can be used to upload training documents and trigger retraining of the model. Maximum number of datasets per tenant: 10</p> <p>Parameters</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>payload * required (body)</td> <td> Example Value - Model <pre>{ "description": "SAP BER default dataset description" }</pre> Parameter content type <input type="text" value="application/json"/> </td> </tr> </tbody> </table>	Name	Description	payload * required (body)	Example Value - Model <pre>{ "description": "SAP BER default dataset description" }</pre> Parameter content type <input type="text" value="application/json"/>
Name	Description				
payload * required (body)	Example Value - Model <pre>{ "description": "SAP BER default dataset description" }</pre> Parameter content type <input type="text" value="application/json"/>				

Explanation	Screenshot				
<p>3. In payload, enter the description you want for your dataset. And click on Execute. In this example, you can use the following:</p> <pre data-bbox="62 587 543 720" style="background-color: #e0e0e0; padding: 5px;">JSON { "description": "SAP TechEd dataset" }</pre>	 <p>POST /datasets Creates a new training dataset entry</p> <p>Creates a new training dataset ID that can be used to upload training documents and trigger retraining of the model. Maximum number of datasets per tenant: 10</p> <p>Parameters</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>payload • required (body)</td> <td> <input style="width: 100%;" type="text" value='{"description": "SAP TechEd dataset"}'/> Edit Value Model </td> </tr> </tbody> </table> <p>Cancel</p> <p>Parameter content type application/json</p> <p style="text-align: center;">Execute</p>	Name	Description	payload • required (body)	<input style="width: 100%;" type="text" value='{"description": "SAP TechEd dataset"}'/> Edit Value Model
Name	Description				
payload • required (body)	<input style="width: 100%;" type="text" value='{"description": "SAP TechEd dataset"}'/> Edit Value Model				
<p>4. Copy the datasetId from Response body to upload documents in the next step.</p>	<p>Server response</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Code</th> <th>Details</th> </tr> </thead> <tbody> <tr> <td>201</td> <td> <p>Response body</p> <pre data-bbox="861 1184 1506 1326" style="background-color: black; color: green; padding: 10px;">{ "data": { "datasetId": "17d77cccd-c50f-492f-b2f6-e31c9f213d44", "message": "Dataset has been created successfully" } }</pre> </td> </tr> </tbody> </table>	Code	Details	201	<p>Response body</p> <pre data-bbox="861 1184 1506 1326" style="background-color: black; color: green; padding: 10px;">{ "data": { "datasetId": "17d77cccd-c50f-492f-b2f6-e31c9f213d44", "message": "Dataset has been created successfully" } }</pre>
Code	Details				
201	<p>Response body</p> <pre data-bbox="861 1184 1506 1326" style="background-color: black; color: green; padding: 10px;">{ "data": { "datasetId": "17d77cccd-c50f-492f-b2f6-e31c9f213d44", "message": "Dataset has been created successfully" } }</pre>				
<p>5. Verify that dataset is created. Click on the endpoint GET /datasets/{datasetId} and click on Try it Out</p>	 <p>GET /datasets/{datasetId} Get specific dataset details</p> <p>Get specific dataset details</p> <p>Parameters</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>datasetId • required string (path)</td> <td></td> </tr> </tbody> </table> <p>Try it out</p>	Name	Description	datasetId • required string (path)	
Name	Description				
datasetId • required string (path)					
<p>6. For datasetId field, enter the value copied in step 4 and click on Execute</p>	 <p>GET /datasets/{datasetId} Get specific dataset details</p> <p>Get specific dataset details</p> <p>Parameters</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>datasetId • required string (path)</td> <td><input type="text" value="17d77cccd-c50f-492f-b2f6-e31c9f213d44"/></td> </tr> </tbody> </table> <p>Cancel</p> <p style="text-align: center;">Execute</p> <p>Clear</p>	Name	Description	datasetId • required string (path)	<input type="text" value="17d77cccd-c50f-492f-b2f6-e31c9f213d44"/>
Name	Description				
datasetId • required string (path)	<input type="text" value="17d77cccd-c50f-492f-b2f6-e31c9f213d44"/>				

Explanation	Screenshot
<p>7. Verify that the dataset details are displayed</p>	 <p>The screenshot shows a REST API response. On the left, there is a table with two columns: 'Code' and 'Details'. The 'Code' column contains the value '200'. The 'Details' column contains a JSON object under the heading 'Response body'. A red box highlights this JSON object. The JSON content is:</p> <pre> { "data": { "datasetId": "17d77cccd-c50f-492f-b2f6-e31c9f213d44", "createdAt": "2020-10-06T09:39:08", "description": "SAP TechEd dataset", "documentCount": 0 } } </pre>

2.2 Understanding the training document

Business Entity Recognition requires a specific data format to train a model. The data format prerequisites are as follows:

- File format: JSONL
- File extension: JSON
- File size: maximum of 10 MB
- File should not contain malicious data
- File should not contain blank lines at the end
- Every line in the file should have one JSON object, which has the following mandatory fields:
 - ID (Number): used to represent the unique record ID and the ID should be unique throughout the dataset
 - Text (String): contains the data on which model must be trained
 - Labels (Array): contain the annotation details, with the text as an array of characters that starts with the index 0. Each label is in the following format:
 - startIndex (Number): starting char index of the phrase
 - endIndex (Number): ending char index of the phrase
 - Annotation (String): annotation (class label) name

Below is an example of how a training document looks:

name of file

Unique id's for every record

Starting index of word

Ending index of word

Class label

```

1 {"id": 1, "text": "Hi John, Please provide the details of invoice AB12345. Regards, Mike", "labels": [[48, 55, "InvoiceID"]]}
2 {"id": 2, "text": "Hi Kay, Vendor DE029876 has requested invoice AB13145 details, kindly process the request ASAP. Regards, John", "labels": [[16, 24, "VendorID"], [47, 54, "InvoiceID"]]}

```

Additional points to note:

- Every JSON object in the file is a record.
- If a record doesn't have any annotations, then that record is discarded.
- There should be a minimum of 50 references per annotation. If the number of references is lower than 50, that annotation is discarded.
- There should be at least one annotation in the dataset to trigger the training.

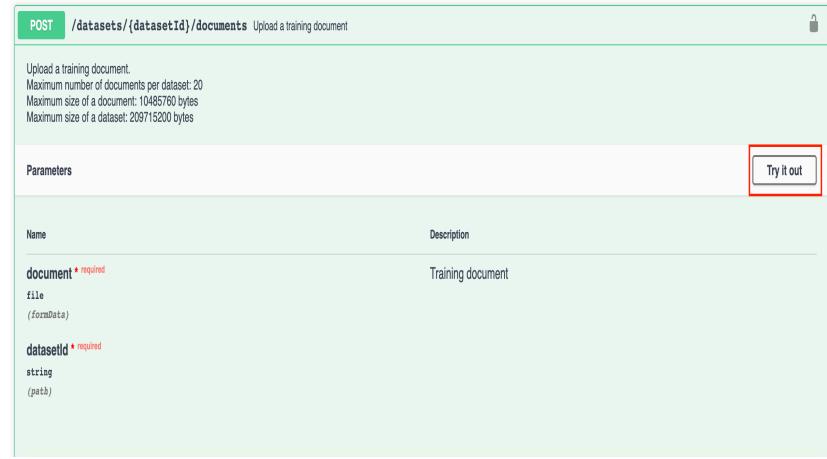
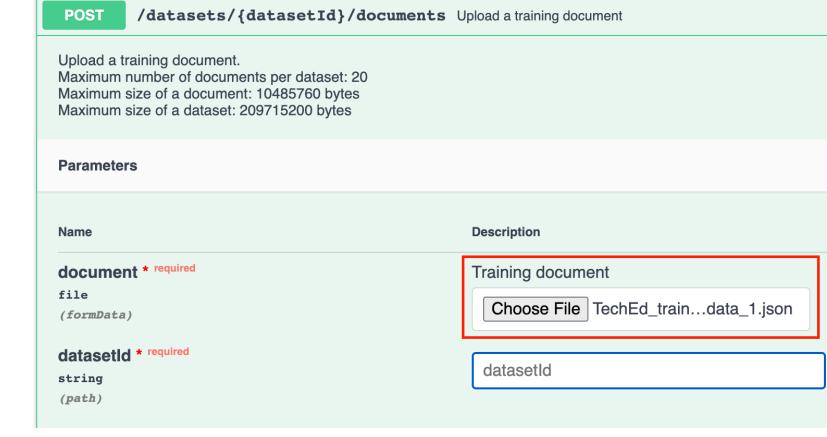
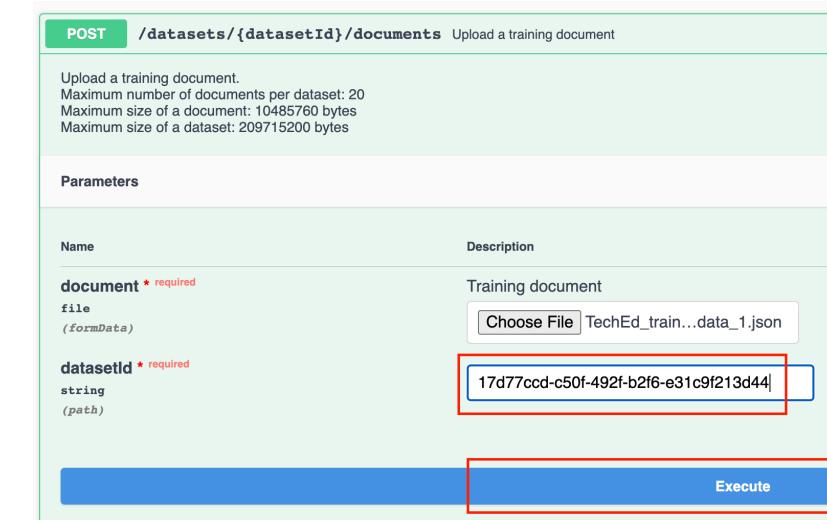
2.3 Upload documents

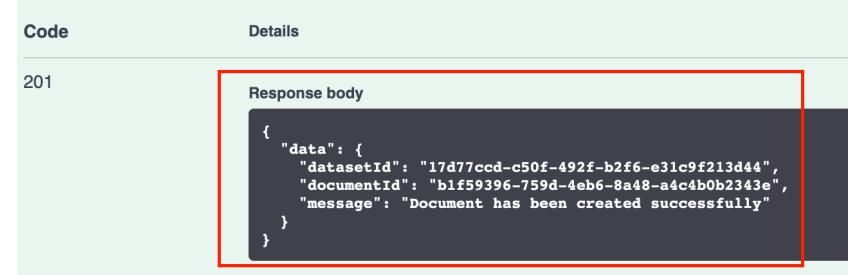
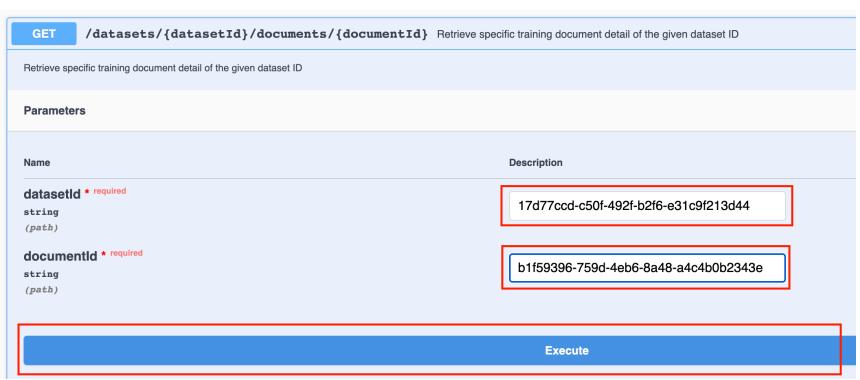
Pre-requisite:

Navigate to the folder – **exercises** and download the documents: **TechEd_training_data_1.json** and **TechEd_training_data_2.json**

Please do not upload any other documents here, especially the documents which might contain Personally Identifiable Information.

Explanation	Screenshot
<p>1. Click the endpoint POST /datasets/{datasetId}/documents, under Datasets to expand it.</p>	<p>Datasets Training Data APIs : Upload, delete and get training data</p> <pre> GET /datasets Retrieve all datasets POST /datasets Creates a new training dataset entry DELETE /datasets/{datasetId} Deleting a dataset and its contents GET /datasets/{datasetId} Get specific dataset details GET /datasets/{datasetId}/documents Get documents present in a dataset POST /datasets/{datasetId}/documents Upload a training document DELETE /datasets/{datasetId}/documents/{documentId} Deletes specific training document of the given dataset ID GET /datasets/{datasetId}/documents/{documentId} Retrieve specific training document detail of the given dataset ID </pre>

Explanation	Screenshot										
2. Click on Try it out	 <p>POST /datasets/{datasetId}/documents Upload a training document</p> <p>Upload a training document. Maximum number of documents per dataset: 20 Maximum size of a document: 10485760 bytes Maximum size of a dataset: 209715200 bytes</p> <p>Parameters</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>document * required</td> <td>Training document</td> </tr> <tr> <td>file (formData)</td> <td></td> </tr> <tr> <td>datasetId * required</td> <td></td> </tr> <tr> <td>string (path)</td> <td></td> </tr> </tbody> </table>	Name	Description	document * required	Training document	file (formData)		datasetId * required		string (path)	
Name	Description										
document * required	Training document										
file (formData)											
datasetId * required											
string (path)											
3. Click on Choose File . Navigate to the downloaded documents and choose the file TechEd_training_data_1.json .	 <p>POST /datasets/{datasetId}/documents Upload a training document</p> <p>Upload a training document. Maximum number of documents per dataset: 20 Maximum size of a document: 10485760 bytes Maximum size of a dataset: 209715200 bytes</p> <p>Parameters</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>document * required</td> <td>Training document</td> </tr> <tr> <td>file (formData)</td> <td>Choose File TechEd_training_data_1.json</td> </tr> <tr> <td>datasetId * required</td> <td></td> </tr> <tr> <td>string (path)</td> <td>datasetId</td> </tr> </tbody> </table>	Name	Description	document * required	Training document	file (formData)	Choose File TechEd_training_data_1.json	datasetId * required		string (path)	datasetId
Name	Description										
document * required	Training document										
file (formData)	Choose File TechEd_training_data_1.json										
datasetId * required											
string (path)	datasetId										
4. For datasetId field, enter the datasetId created in section 2.2. Click on Execute	 <p>POST /datasets/{datasetId}/documents Upload a training document</p> <p>Upload a training document. Maximum number of documents per dataset: 20 Maximum size of a document: 10485760 bytes Maximum size of a dataset: 209715200 bytes</p> <p>Parameters</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>document * required</td> <td>Training document</td> </tr> <tr> <td>file (formData)</td> <td>Choose File TechEd_training_data_1.json</td> </tr> <tr> <td>datasetId * required</td> <td>17d77cccd-c50f-492f-b2f6-e31c9f213d44</td> </tr> <tr> <td>string (path)</td> <td></td> </tr> </tbody> </table> <p>Execute</p>	Name	Description	document * required	Training document	file (formData)	Choose File TechEd_training_data_1.json	datasetId * required	17d77cccd-c50f-492f-b2f6-e31c9f213d44	string (path)	
Name	Description										
document * required	Training document										
file (formData)	Choose File TechEd_training_data_1.json										
datasetId * required	17d77cccd-c50f-492f-b2f6-e31c9f213d44										
string (path)											

Explanation	Screenshot
<p>5. Verify that you get a response code of 201. Copy the documentId from the response body</p>	 <p>The screenshot shows a REST API response. The 'Code' column displays '201'. The 'Details' column shows the 'Response body' which contains the following JSON:</p> <pre>{ "data": { "datasetId": "17d77cccd-c50f-492f-b2f6-e31c9f213d44", "documentId": "b1f59396-759d-4eb6-8a48-a4c4b0b2343e", "message": "Document has been created successfully" } }</pre>
<p>6. You can additionally verify the document details. Click on endpoint GET /datasets/{datasetId}/documents/{documentId}.</p> <p>7. Click on Try it Out and enter the values of datasetId and documentId which you have already copied.</p> <p>8. Click on Execute to get the document details</p>	 <p>The screenshot shows the 'Try it Out' interface for the specified API endpoint. It includes the following details:</p> <ul style="list-style-type: none"> Method: GET Endpoint: /datasets/{datasetId}/documents/{documentId} Description: Retrieve specific training document detail of the given dataset ID Parameters: <ul style="list-style-type: none"> Name: datasetId Description: 17d77cccd-c50f-492f-b2f6-e31c9f213d44 Name: documentId Description: b1f59396-759d-4eb6-8a48-a4c4b0b2343e Execute: A blue button at the bottom.

Repeat the steps 1 to 6 for all the downloaded document: **TechEd_training_data_2.json**.

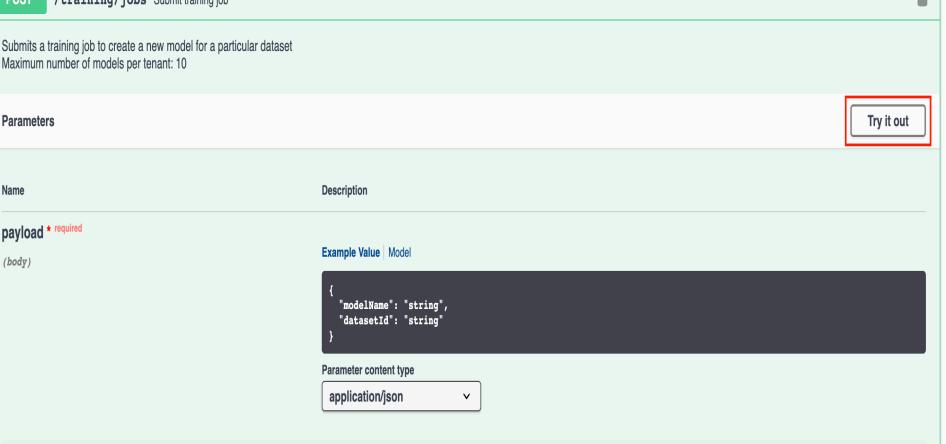
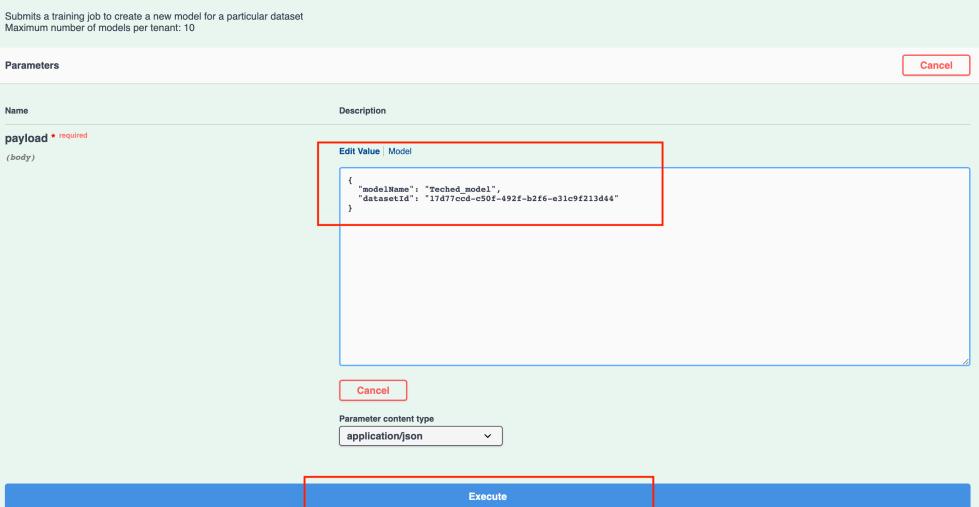
3 BER TRAINING

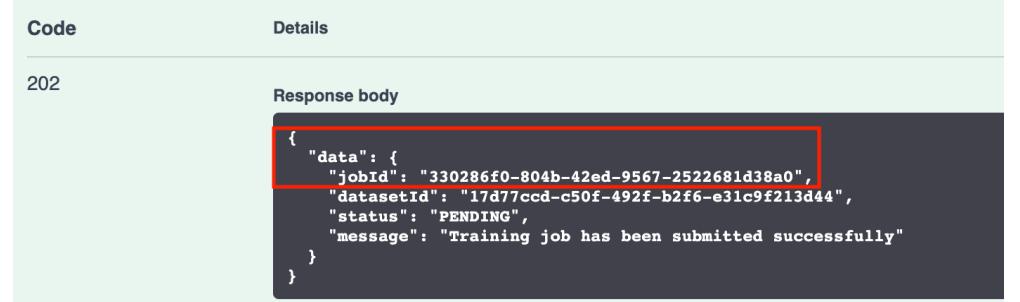
Objective

In this exercise, you would be able to explore training related APIs for BER. Since, BER training takes around 5 hours or more to complete, we would not be able to verify the resulting model in this exercise.

3.1 Trigger training job

Follow these steps to submit a training job with the created dataset.

Explanation	Screenshot
<p>1. Click the endpoint POST /training/jobs, under Training to expand it.</p>	<p>Training Training Job APIs : Trigger training job and get status</p>  <p>POST /training/jobs Submit training job</p> <p>DELETE /training/jobs/{jobId} delete training job</p> <p>GET /training/jobs/{jobId} Get training status</p>
<p>2. Click on Try it out</p>	 <p>POST /training/jobs Submit training job</p> <p>Submits a training job to create a new model for a particular dataset Maximum number of models per tenant: 10</p> <p>Parameters</p> <p>Name Description</p> <p>payload * required</p> <p>(body)</p> <p>Example Value Model</p> <pre>{ "modelName": "string", "datasetId": "string" }</pre> <p>Parameter content type</p> <p>application/json</p>
<p>3. In payload, enter the name of the model you want to train for the modelName field. Also enter the datasetId copied in section 2 for the datasetId field. Click on Execute.</p> <p>JSON</p> <pre>{ "modelName": "Teched_model", "datasetId": <datasetId> }</pre>	 <p>POST /training/jobs Submit training job</p> <p>Submits a training job to create a new model for a particular dataset Maximum number of models per tenant: 10</p> <p>Parameters</p> <p>Name Description</p> <p>payload * required</p> <p>(body)</p> <p>Edit Value Model</p> <pre>{ "modelName": "Teched_model", "datasetId": "1d77cc0-c50f-492f-b2f6-a31c9f211d44" }</pre> <p>Cancel</p> <p>Parameter content type</p> <p>application/json</p> <p>Execute</p>

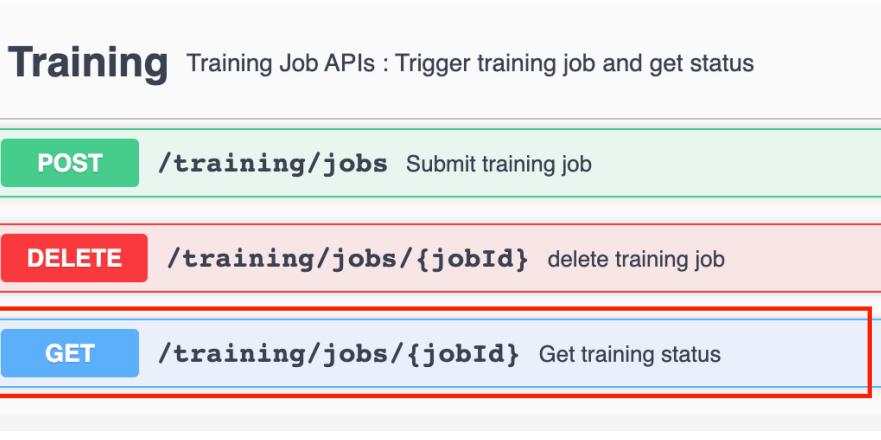
Explanation	Screenshot
<p>4. Verify that you get a response code of 202. This indicates that your training job has been successfully submitted. Copy the jobId from the response body</p>	 <p>The screenshot shows a REST API response. The 'Code' column displays '202'. The 'Details' column shows the 'Response body' which contains the following JSON:</p> <pre> { "data": { "jobId": "330286f0-804b-42ed-9567-2522681d38a0", "datasetId": "17d77cccd-c50f-492f-b2f6-e31c9f213d44", "status": "PENDING", "message": "Training job has been submitted successfully" } } </pre>

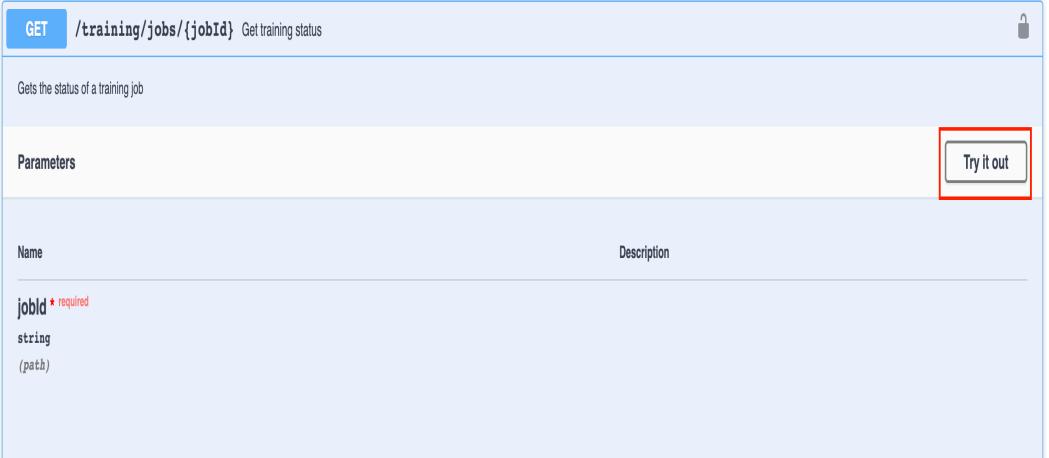
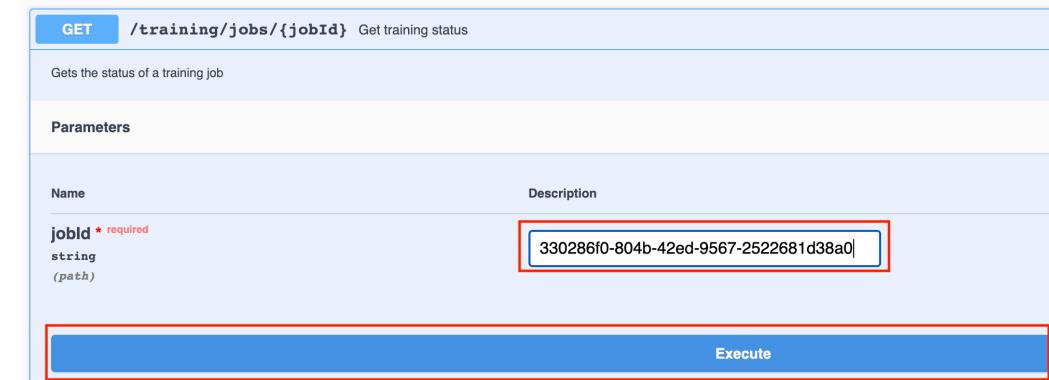
Note: modelName should adhere to following conventions:

- Can't have the same name as a Business Entity Recognition pre-trained model
- Can't have "sap_" as a prefix
- Starts with an alphanumeric character
- May include the special characters "-" and "
- Can have a maximum of 64 characters

3.2 Get training job status

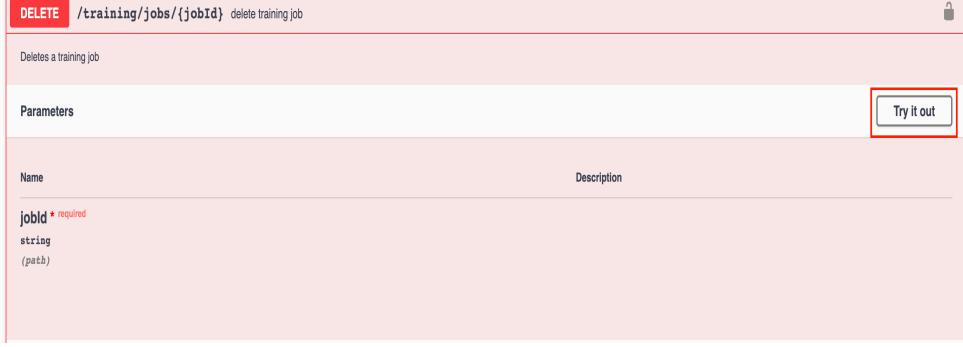
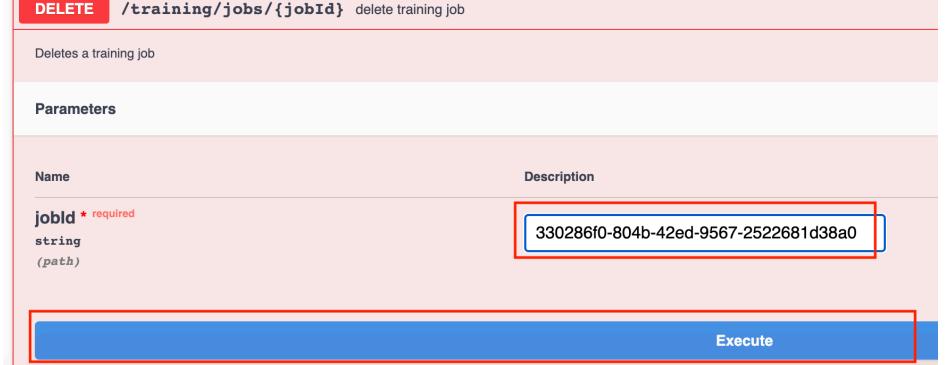
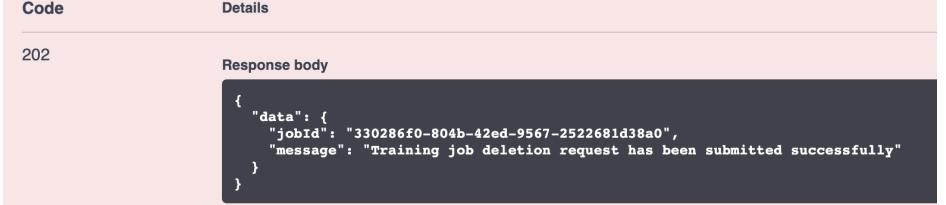
Follow these steps to get the status of your submitted training job

Explanation	Screenshot
<p>1. Click the endpoint GET /training/jobs/{jobId}, under Training to expand it.</p>	 <p>The screenshot shows the 'Training' section of the API documentation. It includes the following information:</p> <ul style="list-style-type: none"> Training Job APIs : Trigger training job and get status POST /training/jobs Submit training job DELETE /training/jobs/{jobId} delete training job GET /training/jobs/{jobId} Get training status (This endpoint is highlighted with a red box)

Explanation	Screenshot				
2. Click on Try it out	 <p>GET /training/jobs/{jobId} Get training status</p> <p>Gets the status of a training job</p> <p>Parameters</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>jobId * required</td> <td>string (path)</td> </tr> </tbody> </table> <p>Try it out</p>	Name	Description	jobId * required	string (path)
Name	Description				
jobId * required	string (path)				
3. For jobId field, enter the jobId copied in the previous exercise. Click on Execute	 <p>GET /training/jobs/{jobId} Get training status</p> <p>Gets the status of a training job</p> <p>Parameters</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>jobId * required</td> <td>string (path)</td> </tr> </tbody> </table> <p>330286f0-804b-42ed-9567-2522681d38a0</p> <p>Execute</p>	Name	Description	jobId * required	string (path)
Name	Description				
jobId * required	string (path)				
4. Verify that you get a response of 200 which indicates that job status has been retrieved successfully.	<table border="1"> <thead> <tr> <th data-bbox="507 1248 812 1290">Code</th> <th data-bbox="812 1248 1556 1290">Details</th> </tr> </thead> <tbody> <tr> <td data-bbox="507 1290 812 1571">200</td> <td data-bbox="812 1290 1556 1571"> <p>Response body</p> <pre>{ "data": { "jobId": "330286f0-804b-42ed-9567-2522681d38a0", "status": "RUNNING" } }</pre> </td></tr> </tbody> </table>	Code	Details	200	<p>Response body</p> <pre>{ "data": { "jobId": "330286f0-804b-42ed-9567-2522681d38a0", "status": "RUNNING" } }</pre>
Code	Details				
200	<p>Response body</p> <pre>{ "data": { "jobId": "330286f0-804b-42ed-9567-2522681d38a0", "status": "RUNNING" } }</pre>				

3.3 Delete training job

BER permits one concurrent training job per tenant and considering long running jobs for BER, follow these steps to delete an ongoing training job.

Explanation	Screenshot				
<p>1. Click the endpoint DELETE /training/jobs/{jobId}, under Training to expand it.</p>	<p>Training Training Job APIs : Trigger training job and get status</p>  <pre> POST /training/jobs Submit training job DELETE /training/jobs/{jobId} delete training job GET /training/jobs/{jobId} Get training status </pre>				
<p>2. Click on Try it out</p>	 <p>DELETE /training/jobs/{jobId} delete training job</p> <p>Deletes a training job</p> <p>Parameters</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>jobId * required</td> <td>string (path)</td> </tr> </tbody> </table> <p>Try it out</p>	Name	Description	jobId * required	string (path)
Name	Description				
jobId * required	string (path)				
<p>3. For jobId field, enter the jobId copied in the previous exercise. Click on Execute</p>	 <p>DELETE /training/jobs/{jobId} delete training job</p> <p>Deletes a training job</p> <p>Parameters</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>jobId * required</td> <td>string (path)</td> </tr> </tbody> </table> <p>330286f0-804b-42ed-9567-2522681d38a0</p> <p>Execute</p>	Name	Description	jobId * required	string (path)
Name	Description				
jobId * required	string (path)				
<p>4. Verify that you get a response of 202 which indicates that delete request has been submitted successfully.</p>	 <table border="1"> <thead> <tr> <th>Code</th> <th>Details</th> </tr> </thead> <tbody> <tr> <td>202</td> <td> <p>Response body</p> <pre>{ "data": { "jobId": "330286f0-804b-42ed-9567-2522681d38a0", "message": "Training job deletion request has been submitted successfully" } }</pre> </td> </tr> </tbody> </table>	Code	Details	202	<p>Response body</p> <pre>{ "data": { "jobId": "330286f0-804b-42ed-9567-2522681d38a0", "message": "Training job deletion request has been submitted successfully" } }</pre>
Code	Details				
202	<p>Response body</p> <pre>{ "data": { "jobId": "330286f0-804b-42ed-9567-2522681d38a0", "message": "Training job deletion request has been submitted successfully" } }</pre>				

Explanation	Screenshot	
	Code	Details
	404	<p>Error: Not Found</p> <p>Response body</p> <pre>{ "errors": { "message": "Training job id not found" } }</pre>

4 BER MODELS

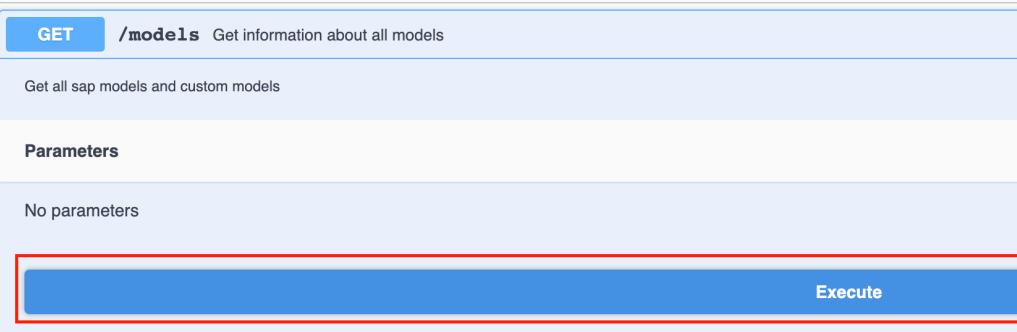
Objective

In this exercise you would be able to explore the APIs related to model management. You would be able to get the details of available models and versions.

4.1 Get all models

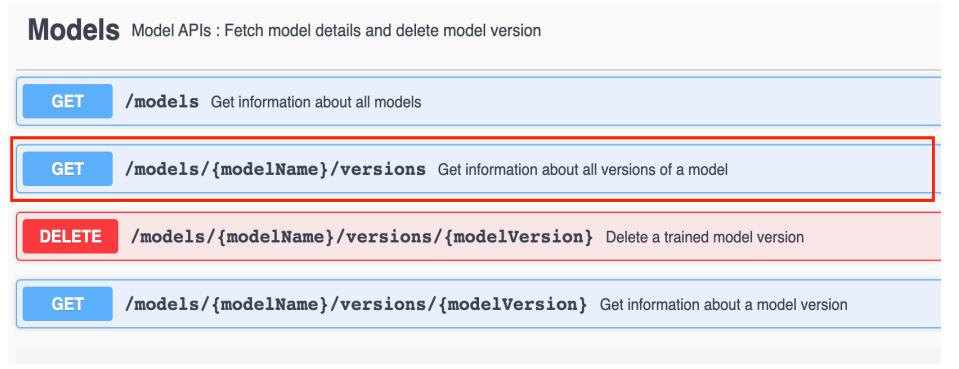
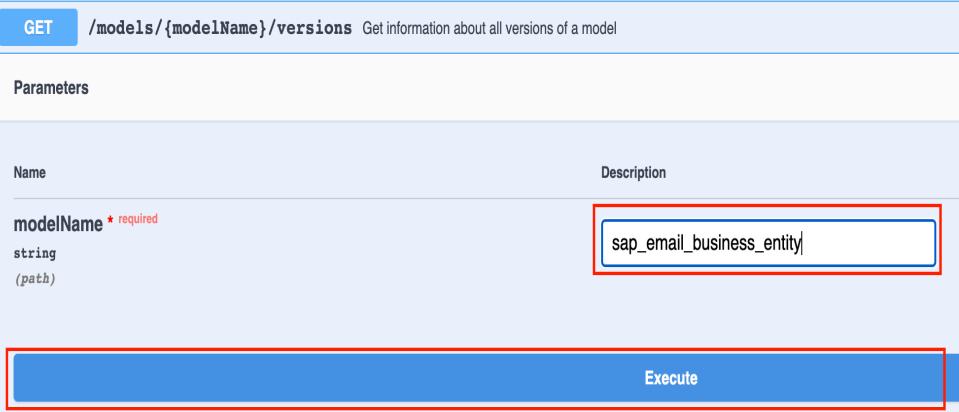
Follow these steps to get all the models available for your tenant.

Explanation	Screenshot
<p>1. Click the endpoint GET /models, under Models to expand it.</p>	<p>Models Model APIs : Fetch model details and delete model version</p>  <pre> Models Model APIs : Fetch model details and delete model version GET /models Get information about all models GET /models/{modelName}/versions Get information about all versions of a model DELETE /models/{modelName}/versions/{modelVersion} Delete a trained model version GET /models/{modelName}/versions/{modelVersion} Get information about a model version </pre>

Explanation	Screenshot				
2. Click on Try it out	 <p>GET /models Get information about all models</p> <p>Get all sap models and custom models</p> <p>Parameters</p> <p>No parameters</p> <p>Try it out</p>				
3. Click on Execute	 <p>GET /models Get information about all models</p> <p>Get all sap models and custom models</p> <p>Parameters</p> <p>No parameters</p> <p>Execute</p>				
<p>4. Verify that you get a response of 200. Examine the response to see the models that are available for your tenant.</p> <p>You would see 2 models: sap_email_business_entity and sap_invoice_header. These are the pre-trained models that are immediately productive with the BER service.</p>	<table border="1"> <thead> <tr> <th data-bbox="551 1030 812 1064">Code</th> <th data-bbox="812 1030 1568 1064">Details</th> </tr> </thead> <tbody> <tr> <td data-bbox="551 1094 812 1127">200</td> <td data-bbox="812 1094 1568 1704"> <p>Response body</p> <pre>{ "data": { "sapModels": { "count": 2, "models": [{ "modelName": "sap_invoice_header", "versions": [{ "modelVersion": 1 }] }, { "modelName": "sap_email_business_entity", "versions": [{ "modelVersion": 1 }, { "modelVersion": 2 }] }] } } }</pre> </td></tr> </tbody> </table>	Code	Details	200	<p>Response body</p> <pre>{ "data": { "sapModels": { "count": 2, "models": [{ "modelName": "sap_invoice_header", "versions": [{ "modelVersion": 1 }] }, { "modelName": "sap_email_business_entity", "versions": [{ "modelVersion": 1 }, { "modelVersion": 2 }] }] } } }</pre>
Code	Details				
200	<p>Response body</p> <pre>{ "data": { "sapModels": { "count": 2, "models": [{ "modelName": "sap_invoice_header", "versions": [{ "modelVersion": 1 }] }, { "modelName": "sap_email_business_entity", "versions": [{ "modelVersion": 1 }, { "modelVersion": 2 }] }] } } }</pre>				

4.2 Get model details by model name

Follow these steps to get information about different versions of a model.

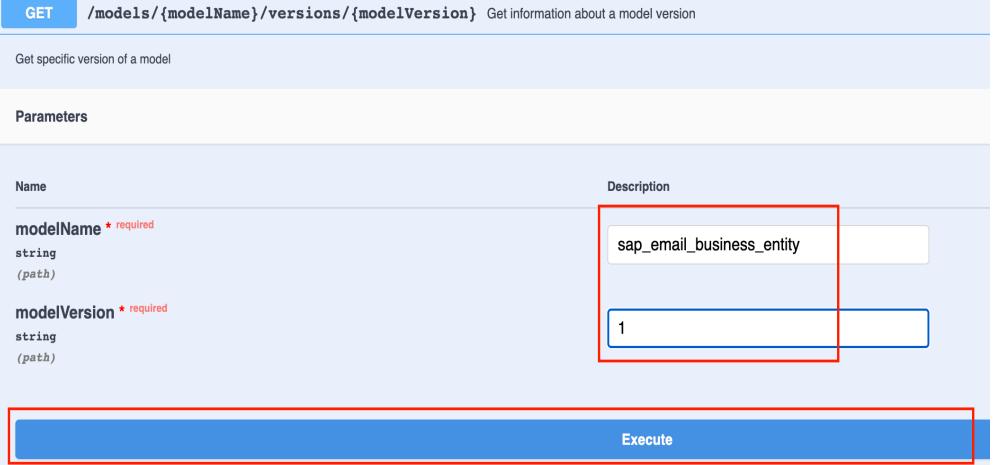
Explanation	Screenshot				
<p>1. Click the endpoint GET /models/{modelName}/versions, under Models to expand it.</p>	 <p>Models Model APIs : Fetch model details and delete model version</p> <p>GET /models Get information about all models</p> <p>GET /models/{modelName}/versions Get information about all versions of a model</p> <p>DELETE /models/{modelName}/versions/{modelVersion} Delete a trained model version</p> <p>GET /models/{modelName}/versions/{modelVersion} Get information about a model version</p>				
<p>2. Click on Try it out. For the modelName field, enter sap_email_business_entity. Click on Execute.</p>	 <p>GET /models/{modelName}/versions Get information about all versions of a model</p> <p>Parameters</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>modelName <small>* required</small> string (path)</td> <td>sap_email_business_entity</td> </tr> </tbody> </table> <p>Execute</p>	Name	Description	modelName <small>* required</small> string (path)	sap_email_business_entity
Name	Description				
modelName <small>* required</small> string (path)	sap_email_business_entity				

Explanation	Screenshot				
<p>3. Verify that you get a response of 200. Examine the response to see the versions of the sap_email_business_entity that are available. You will notice 2 versions of the model.</p>	<div style="display: flex; justify-content: space-between;"> Code Details </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">200</td> <td style="padding: 5px;">Response body</td> </tr> <tr> <td colspan="2" style="padding: 10px; background-color: #f0f0f0;"> <pre>{ "data": { "modelName": "sap_email_business_entity", "modelType": "sapModel", "count": 2, "supportedLanguages": ["EN", "DE"], "versions": [{ "modelVersion": 1, "capabilities": [{ "entity": "invoiceReferenceNumber" }, { "entity": "vendorID" }] }, { "modelVersion": 2, "capabilities": [{ "entity": "invoiceReferenceNumber" }, { "entity": "customerID" }] }] } }</pre> </td> </tr> </table>	200	Response body	<pre>{ "data": { "modelName": "sap_email_business_entity", "modelType": "sapModel", "count": 2, "supportedLanguages": ["EN", "DE"], "versions": [{ "modelVersion": 1, "capabilities": [{ "entity": "invoiceReferenceNumber" }, { "entity": "vendorID" }] }, { "modelVersion": 2, "capabilities": [{ "entity": "invoiceReferenceNumber" }, { "entity": "customerID" }] }] } }</pre>	
200	Response body				
<pre>{ "data": { "modelName": "sap_email_business_entity", "modelType": "sapModel", "count": 2, "supportedLanguages": ["EN", "DE"], "versions": [{ "modelVersion": 1, "capabilities": [{ "entity": "invoiceReferenceNumber" }, { "entity": "vendorID" }] }, { "modelVersion": 2, "capabilities": [{ "entity": "invoiceReferenceNumber" }, { "entity": "customerID" }] }] } }</pre>					
Explanation	Screenshot				
<p>1. Click the endpoint GET /models/{modelName}/versions, under Models to expand it.</p>	<div style="display: flex; justify-content: space-between;"> Models Model APIs : Fetch model details and delete model version </div> <div style="margin-top: 10px;"> <p>GET /models Get information about all models</p> <p>GET /models/{modelName}/versions Get information about all versions of a model</p> <p>DELETE /models/{modelName}/versions/{modelVersion} Delete a trained model version</p> <p>GET /models/{modelName}/versions/{modelVersion} Get information about a model version</p> </div>				

Explanation	Screenshot						
<p>2. Click on Try it out. For the modelName field, enter sap_invoice_header. Click on Execute.</p>	 <p>GET /models/{modelName}/versions Get information about all versions of a model</p> <p>Parameters</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>modelName * required</td> <td>sap_invoice_header</td> </tr> <tr> <td>string</td> <td>(path)</td> </tr> </tbody> </table> <p style="text-align: right;">Execute</p>	Name	Description	modelName * required	sap_invoice_header	string	(path)
Name	Description						
modelName * required	sap_invoice_header						
string	(path)						
<p>3. Verify that you get a response of 200. Examine the response to see the versions of the sap_invoice_header that are available.</p>	<table border="1"> <thead> <tr> <th data-bbox="621 783 817 817">Code</th> <th data-bbox="817 783 1596 817">Details</th> </tr> </thead> <tbody> <tr> <td data-bbox="621 817 817 1537">200</td> <td data-bbox="817 817 1596 1537"> <p>Response body</p> <pre>{ "data": { "modelName": "sap_invoice_header", "modelType": "sapModel", "count": 1, "supportedLanguages": ["EN", "DE", "FR"], "versions": [{ "modelVersion": 1, "capabilities": [{ "entity": "vendorName" }, { "entity": "vendorAddress" }, { "entity": "invoiceNo" }, { "entity": "vendorBankAccountNo" }, { "entity": "buyerName" }] }] } }</pre> </td></tr> </tbody> </table>	Code	Details	200	<p>Response body</p> <pre>{ "data": { "modelName": "sap_invoice_header", "modelType": "sapModel", "count": 1, "supportedLanguages": ["EN", "DE", "FR"], "versions": [{ "modelVersion": 1, "capabilities": [{ "entity": "vendorName" }, { "entity": "vendorAddress" }, { "entity": "invoiceNo" }, { "entity": "vendorBankAccountNo" }, { "entity": "buyerName" }] }] } }</pre>		
Code	Details						
200	<p>Response body</p> <pre>{ "data": { "modelName": "sap_invoice_header", "modelType": "sapModel", "count": 1, "supportedLanguages": ["EN", "DE", "FR"], "versions": [{ "modelVersion": 1, "capabilities": [{ "entity": "vendorName" }, { "entity": "vendorAddress" }, { "entity": "invoiceNo" }, { "entity": "vendorBankAccountNo" }, { "entity": "buyerName" }] }] } }</pre>						

4.3 Get model details by model name and model version

Follow these steps to know the details of a specific version of any model.

Explanation	Screenshot										
<p>1. Click the endpoint GET /models/{modelName}/versions/{modelVersion}, under Models to expand it.</p>	<p>Models Model APIs : Fetch model details and delete model version</p>  <pre> GET /models Get information about all models GET /models/{modelName}/versions Get information about all versions of a model DELETE /models/{modelName}/versions/{modelVersion} Delete a trained model version GET /models/{modelName}/versions/{modelVersion} Get information about a model version </pre>										
<p>2. Click on Try it out. For the modelName field, enter sap_email_business_entity. For the modelVersion field, enter 1. Click on Execute.</p>	 <p>GET /models/{modelName}/versions/{modelVersion} Get information about a model version</p> <p>Get specific version of a model</p> <p>Parameters</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>modelName * required</td> <td>sap_email_business_entity</td> </tr> <tr> <td>string (path)</td> <td></td> </tr> <tr> <td>modelVersion * required</td> <td>1</td> </tr> <tr> <td>string (path)</td> <td></td> </tr> </tbody> </table> <p>Execute</p>	Name	Description	modelName * required	sap_email_business_entity	string (path)		modelVersion * required	1	string (path)	
Name	Description										
modelName * required	sap_email_business_entity										
string (path)											
modelVersion * required	1										
string (path)											

Explanation	Screenshot
<p>3. Verify that you get a response of 200. Examine the response to see the details of version 1 of sap_email_business_entity.</p>	<p>Code</p> <p>200</p> <p>Details</p> <p>Response body</p> <pre>{ "data": { "modelName": "sap_email_business_entity", "modelVersion": 1, "modelType": "sapModel", "supportedLanguages": ["EN", "DE"], "capabilities": [{ "entity": "invoiceReferenceNumber" }, { "entity": "vendorID" }] } }</pre>

5 BER INFERENCE

Objective

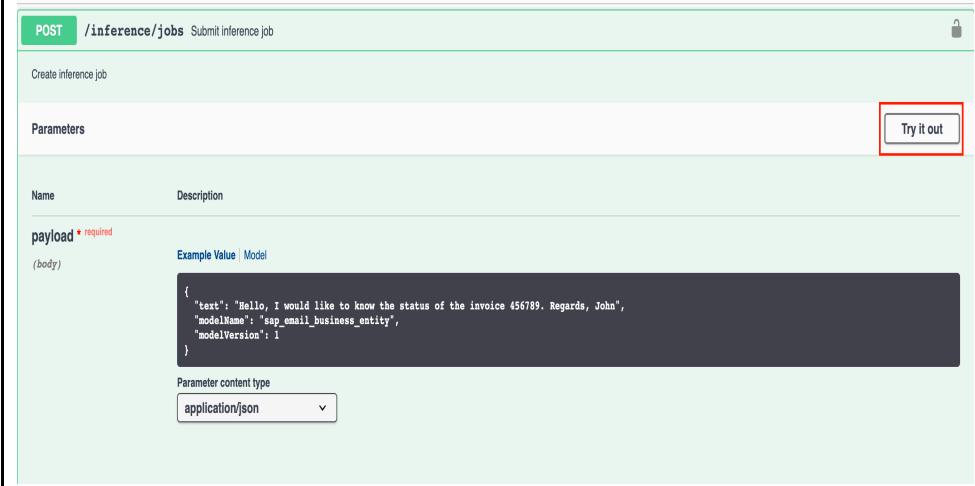
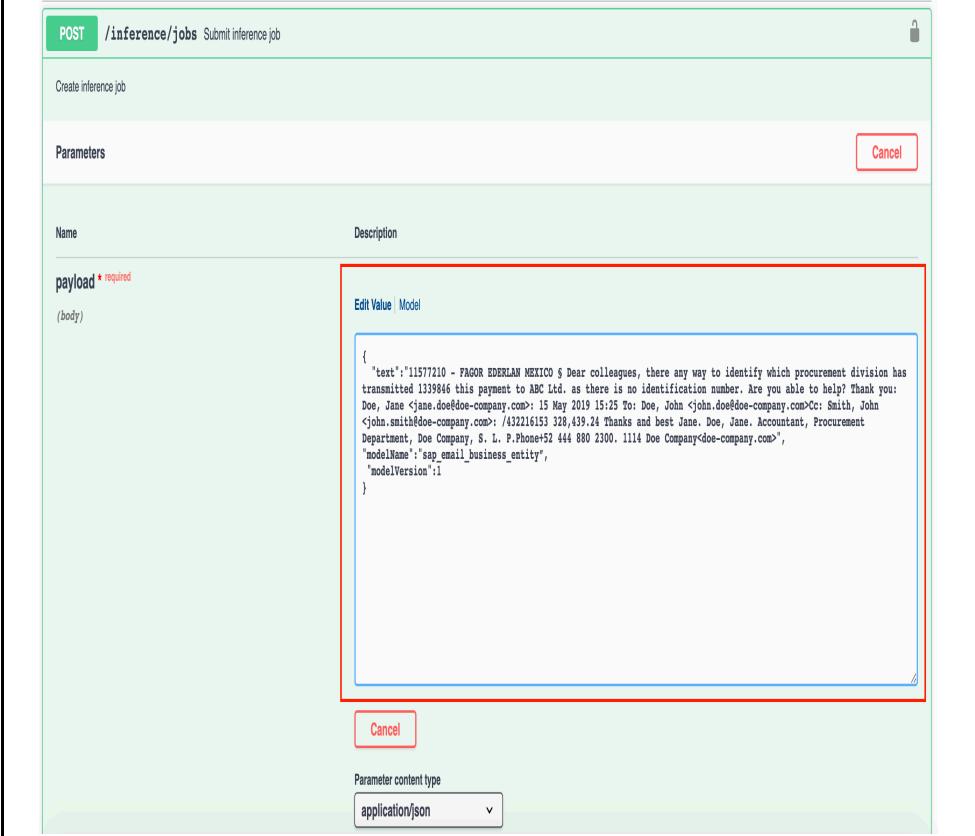
In this exercise, you would be able to see the BER service in action. You can provide a text and the service would extract and highlight the relevant entities with the help of pre-trained models

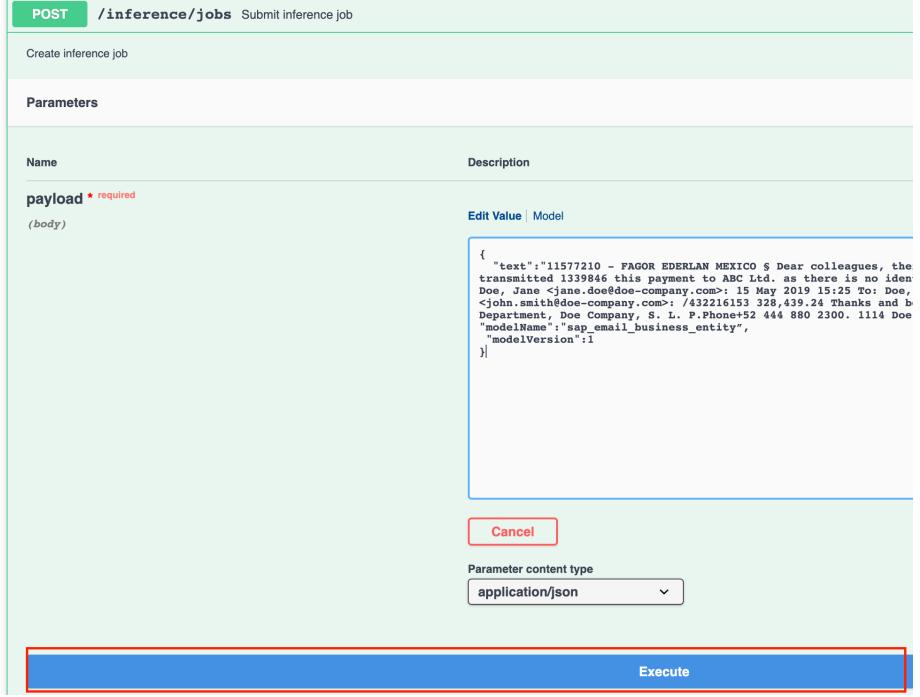
5.1 Inference with sap_email_business_entity

Follow these steps to extract entities with the sap_email_business_entity model

5.1.1 Submit the inference job for sap_email_business_entity model

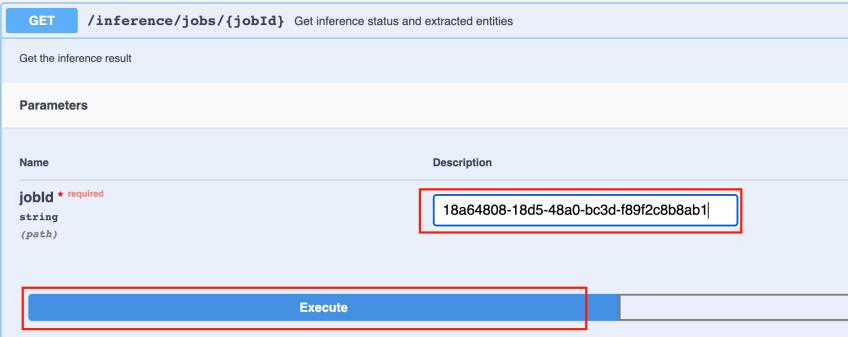
Explanation	Screenshot
<p>1. Click the endpoint POST /inference/jobs, under Inference to expand it.</p>	<p>Inference Inference Job APIs : Trigger inference job and get extracted entities</p> <p>POST /inference/jobs Submit inference job</p> <p>GET /inference/jobs/{jobId} Get inference status and extracted entities</p>

Explanation	Screenshot
<p>2. Click on Try it out</p>	 <p>The screenshot shows the 'Try it out' interface for a POST request to the endpoint '/inference/jobs'. The request is titled 'Submit inference job'. In the 'Parameters' section, there is a single parameter named 'payload' which is marked as required. The example value for 'payload' is a JSON object:</p> <pre>{ "text": "Hello, I would like to know the status of the invoice 456789. Regards, John", "modelName": "sap_email_business_entity", "modelVersion": 1 }</pre> <p>The 'Parameter content type' dropdown is set to 'application/json'.</p>
<p>3. In payload, enter the text you want to extract named entities from, modelName and modelVersion. In this exercise you can use the following:</p> <p>JSON</p> <pre>{ "text": "11577210 - FAGOR EDERLAN MEXICO § Dear colleagues, there any way to identify which procurement division has transmitted 1339846 this payment to ABC Ltd. as there is no identification number. Are you able to help? Thank you: Doe, Jane <jane.doe@doe-company.com>: 15 May 2019 15:25 To: Doe, John <john.doe@doe-company.com>Cc: Smith, John <john.smith@doe-company.com>: /432216153 328,439.24 Thanks and best Jane. Doe, Jane. Accountant, Procurement Department, Doe Company, S. L. P.Phone+52 444 880 2300. 1114 Doe Company<doe-company.com>", "modelName": "sap_email_business_entity", "modelVersion": 1 }</pre>	 <p>The screenshot shows the 'Try it out' interface for the same POST request. The 'payload' field now contains a modified JSON object with a red box highlighting the 'text' field:</p> <pre>{ "text": "11577210 - FAGOR EDERLAN MEXICO § Dear colleagues, there any way to identify which procurement division has transmitted 1339846 this payment to ABC Ltd. as there is no identification number. Are you able to help? Thank you: Doe, Jane <jane.doe@doe-company.com>: 15 May 2019 15:25 To: Doe, John <john.doe@doe-company.com>Cc: Smith, John <john.smith@doe-company.com>: /432216153 328,439.24 Thanks and best Jane. Doe, Jane. Accountant, Procurement Department, Doe Company, S. L. P.Phone+52 444 880 2300. 1114 Doe Company<doe-company.com>", "modelName": "sap_email_business_entity", "modelVersion": 1 }</pre> <p>The 'Parameter content type' dropdown is set to 'application/json'.</p>

Explanation	Screenshot				
<p>4. Click Execute</p>	 <pre>{ "text": "11577210 - FAGOS EDERLAN MEXICO \$ Dear colleagues, the transmitted 1339846 this payment to ABC Ltd. as there is no iden Doe, Jane<jane.doe@doe-company.com>; 15 May 2019 15:25 to: Doe, John<john.doe@doe-company.com>; 7432216153 328,439,24 Thanks and b Department, Doe Company, S. L. P. Phone+52 444 880 2300. 1114 Doe \"modelName\":\"sap_email_business_entity\", \"modelVersion\":1 }</pre>				
<p>5. Verify that you get a response code of 202. This indicates that your inference job has been successfully submitted. Copy the id from the response body to see result of the extraction in the next step.</p>	<table border="1"> <thead> <tr> <th data-bbox="659 1072 829 1106">Code</th><th data-bbox="829 1072 1600 1106">Details</th></tr> </thead> <tbody> <tr> <td data-bbox="659 1106 829 1419">202</td><td data-bbox="829 1106 1600 1419"> <p>Response body</p> <pre>{ "data": { "id": "18a64808-18d5-48a0-bc3d-f89f2c8b8ab1", "status": "PENDING", "message": "Inference job has been submitted", "modelName": "sap_email_business_entity", "modelVersion": 1 } }</pre> </td></tr> </tbody> </table>	Code	Details	202	<p>Response body</p> <pre>{ "data": { "id": "18a64808-18d5-48a0-bc3d-f89f2c8b8ab1", "status": "PENDING", "message": "Inference job has been submitted", "modelName": "sap_email_business_entity", "modelVersion": 1 } }</pre>
Code	Details				
202	<p>Response body</p> <pre>{ "data": { "id": "18a64808-18d5-48a0-bc3d-f89f2c8b8ab1", "status": "PENDING", "message": "Inference job has been submitted", "modelName": "sap_email_business_entity", "modelVersion": 1 } }</pre>				

5.1.2 Get the inference results for sap_email_business_entity model

Follow these steps to see the text extraction results and the confidence level of the sap_email_business_entity model

Explanation	Screenshot						
<p>1. Click the endpoint GET /inference/jobs/{jobId}, under Inference to expand it.</p>	 <p>Inference Inference Job APIs : Trigger inference job and get extracted entities</p> <p>POST /inference/jobs Submit inference job</p> <p>GET /inference/jobs/{jobId} Get inference status and extracted entities</p>						
<p>2. Click on Try it out. For the jobId field, enter the id copied in previous step. Click on Execute.</p>	 <p>Get the inference result</p> <p>Parameters</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>jobId <small>* required</small> string (path)</td> <td>18a64808-18d5-48a0-bc3d-f89f2c8b8ab1</td> </tr> </tbody> </table> <p>Execute</p>	Name	Description	jobId <small>* required</small> string (path)	18a64808-18d5-48a0-bc3d-f89f2c8b8ab1		
Name	Description						
jobId <small>* required</small> string (path)	18a64808-18d5-48a0-bc3d-f89f2c8b8ab1						
<p>3. Verify that you get a response of 200. In the response, you will find the prediction for the extracted entities. The prediction is made with a probability indicated by the confidence field which represents how certain the model is about the value provided. A confidence of 1 means that the model is 100% sure about its prediction.</p>	<table border="1"> <thead> <tr> <th data-bbox="752 963 817 984">Code</th> <th data-bbox="1008 963 1073 984">Details</th> </tr> </thead> <tbody> <tr> <td data-bbox="752 1015 796 1036">200</td> <td data-bbox="1008 1015 1139 1036">Response body</td> </tr> <tr> <td></td> <td data-bbox="1008 1058 1584 1522"> <pre>{ "data": { "id": "18a64808-18d5-48a0-bc3d-f89f2c8b8ab1", "status": "SUCCESS", "result": [{ "vendorId": [{ "value": "1339846", "confidence": 1 }], "invoiceReferenceNumber": [{ "value": "11577210", "confidence": 0.71 }] }] } }</pre> </td> </tr> </tbody> </table>	Code	Details	200	Response body		<pre>{ "data": { "id": "18a64808-18d5-48a0-bc3d-f89f2c8b8ab1", "status": "SUCCESS", "result": [{ "vendorId": [{ "value": "1339846", "confidence": 1 }], "invoiceReferenceNumber": [{ "value": "11577210", "confidence": 0.71 }] }] } }</pre>
Code	Details						
200	Response body						
	<pre>{ "data": { "id": "18a64808-18d5-48a0-bc3d-f89f2c8b8ab1", "status": "SUCCESS", "result": [{ "vendorId": [{ "value": "1339846", "confidence": 1 }], "invoiceReferenceNumber": [{ "value": "11577210", "confidence": 0.71 }] }] } }</pre>						

Optional exercise

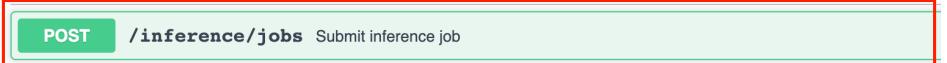
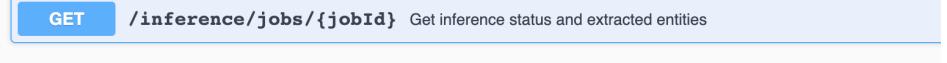
You can also use the following **payload** example with the **sap_email_business_entity** model:

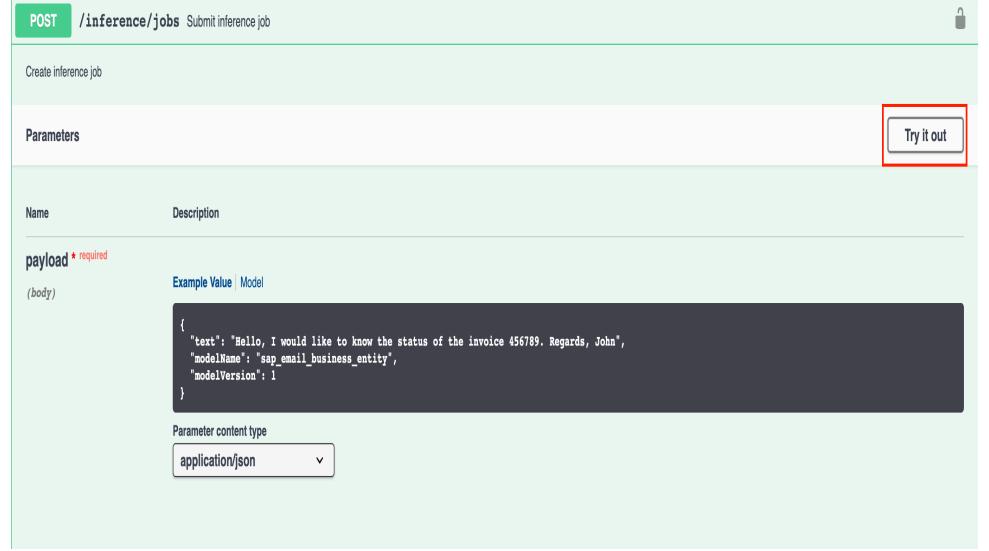
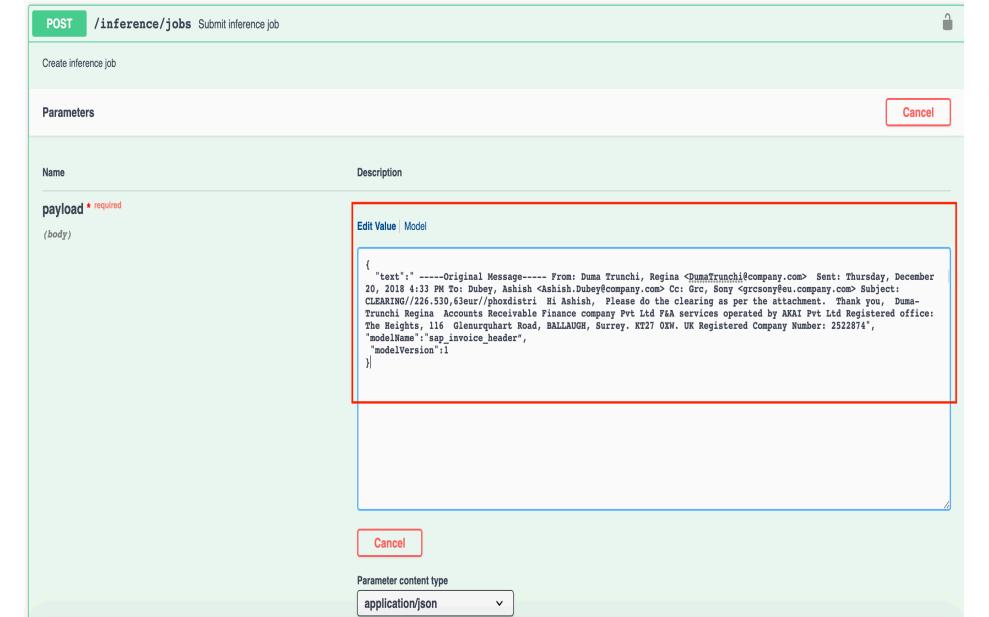
JSON
<pre>{ "text": "Von: lena nova <office@scomputercenter.bg> Gesendet: Montag, 21. Januar 2019, 19:13 Uhr An: canu, ana <ana.canu@company.com>; office@computercenter-bg.com Betreff: RE: M & B EOOD 2639710315 Sehr geehrter Kunde, ich sende Ihnen Informationen für unsere Zahlung. Proforma 198305906 ist ein Teil der Rechnung 4830476570 / 11.01.2019. Rechnungsdatum Betrag in EUR 4830473772 14.12.2018 9,28 € 4830474632 21.12.2018 29,16 € 4830475244 28.12.2018 46,19 € 4830475919 4.1.2019 9,28 € 198305906 10.1.2019 9,29 € 103,20 € 4830476570 11.1.2019 37,13 198305906 10.1.2019 -9,29 € 27,84 Einen schönen Tag noch! Mit freundlichen Grüßen, lena lena nova Verkaufsleiterin M & B EOOD-Computer Center 11, Angel Kanchev Str; 1000, Sofia, Bulgarien Tel.: +359 2 981 58 57; Mobil: + 359 889 61 13 21 E-Mail: office@scomputercenter.bg Besuchen Sie uns unter: www.computercenter.bg", "modelName": "sap_email_business_entity", "modelVersion": 2 }</pre>

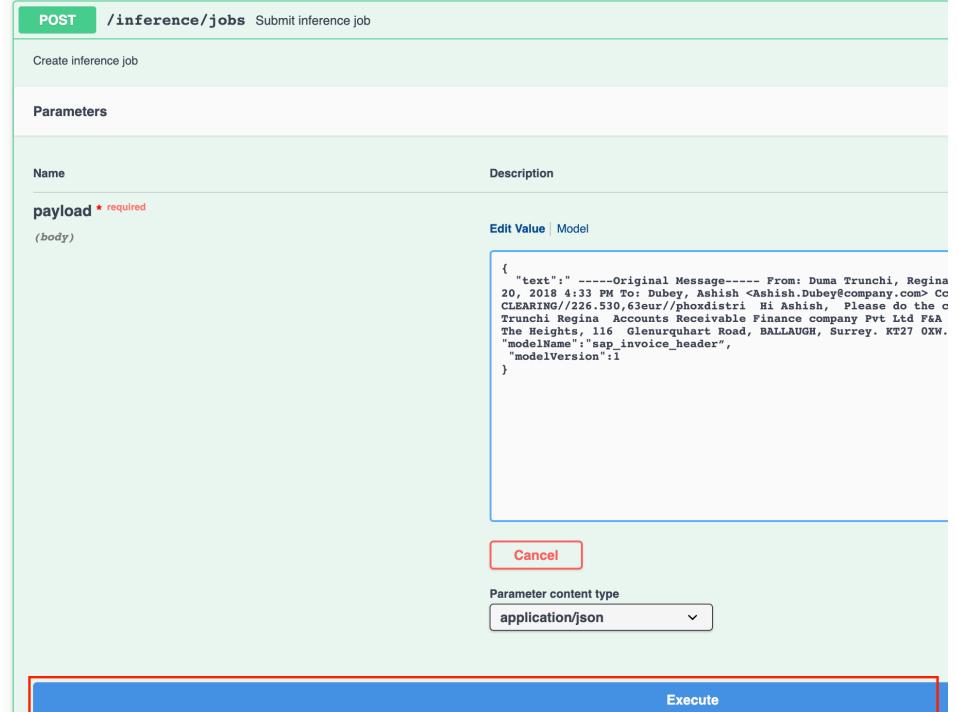
5.2 Inference with sap_invoice_header

Follow these steps to extract entities with the sap_invoice_header model

5.2.1 Submit the inference job for sap_invoice_header model

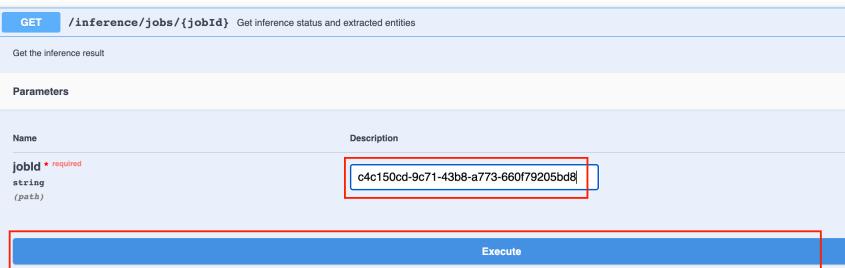
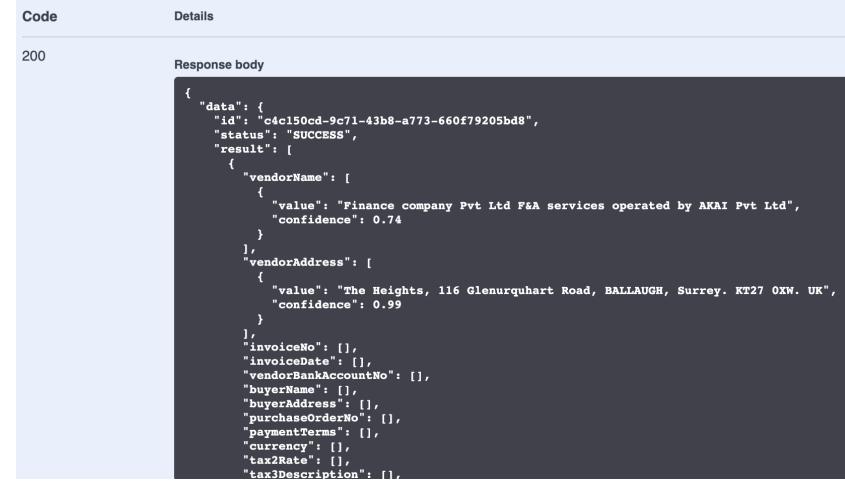
Explanation	Screenshot
<p>1. Click the endpoint POST /inference/jobs, under Inference to expand it.</p>	<p>Inference Inference Job APIs : Trigger inference job and get extracted entities</p>  <p>POST /inference/jobs Submit inference job</p>  <p>GET /inference/jobs/{jobId} Get inference status and extracted entities</p>

Explanation	Screenshot
<p>2. Click on Try it out</p>	 <p>The screenshot shows the 'Try it out' button in the top right corner of the API interface.</p>
<p>3. In payload, enter the text you want to extract named entities from, modelName and modelVersion. In this exercise you can use the following</p> <pre data-bbox="78 1157 567 1790" style="background-color: #f0f0f0; padding: 10px;"> JSON { "text": "-----Original Message----- From: Duma Trunchi, Regina <DumaTrunchi@company.com> Sent: Thursday, December 20, 2018 4:33 PM To: Dubey, Ashish <Ashish.Dubey@company.com> Cc: Grc, Sony <grcsny@eu.company.com> Subject: CLEARING//226.530,63eur//phoxdistri Hi Ashish, Please do the clearing as per the attachment. Thank you, Duma-Trunchi Regina Accounts Receivable Finance company Pvt Ltd F&A services operated by AKAI Pvt Ltd Registered office: The Heights, 116 Glenurquhart Road, BALLAUGH, Surrey. KT27 0XW. UK Registered Company Number: 2522874", "modelName": "sap_invoice_header", "modelVersion": 1 } </pre>	 <p>The screenshot shows the JSON payload entered into the 'payload' field, which is highlighted with a red box.</p>

Explanation	Screenshot				
4. Click Execute	 <pre> POST /inference/jobs Submit inference job Create inference job Parameters Name Description payload * required Edit Value Model (body) { "text": "-----Original Message----- From: Duma Trunchi, Regina 20, 2018 4:33 PM To: Dubey, Ashish <Ashish.Dubey@company.com> Cc CLEARING//226.530.63eur//phoxdistri Hi Ashish, Please do the c Trunchi, Regina Accounts Receivable Finance company Pvt Ltd F&A The Heights, 116 Glenurquhart Road, BALLLAUGH, Surrey. KT27 0XW. "modelName": "sap_invoice_header", "modelVersion": 1 } </pre> <p>Cancel</p> <p>Parameter content type application/json</p> <p>Execute</p>				
5. Verify that you get a response code of 202 . This indicates that your inference job has been successfully submitted. Copy the id from the response body to see result of the extraction in the next step.	<table border="1"> <thead> <tr> <th data-bbox="638 1100 714 1136">Code</th> <th data-bbox="714 1100 1600 1136">Details</th> </tr> </thead> <tbody> <tr> <td data-bbox="638 1136 714 1425">202</td> <td data-bbox="714 1136 1600 1425"> <p>Response body</p> <pre> { "data": { "id": "c4c150cd-9c71-43b8-a773-660f79205bd8", "status": "PENDING", "message": "Inference job has been submitted", "modelName": "sap_invoice_header", "modelVersion": 1 } } </pre> </td></tr> </tbody> </table>	Code	Details	202	<p>Response body</p> <pre> { "data": { "id": "c4c150cd-9c71-43b8-a773-660f79205bd8", "status": "PENDING", "message": "Inference job has been submitted", "modelName": "sap_invoice_header", "modelVersion": 1 } } </pre>
Code	Details				
202	<p>Response body</p> <pre> { "data": { "id": "c4c150cd-9c71-43b8-a773-660f79205bd8", "status": "PENDING", "message": "Inference job has been submitted", "modelName": "sap_invoice_header", "modelVersion": 1 } } </pre>				

5.2.2 Get the inference results for sap_invoice_header model

Follow these steps to see the text extraction results and the confidence level of the sap_invoice_header model

Explanation	Screenshot				
<p>1. Click the endpoint GET /inference/jobs/{jobId}, under Inference to expand it.</p>	 <p>Inference Inference Job APIs : Trigger inference job and get extracted entities</p> <p>POST /inference/jobs Submit inference job</p> <p>GET /inference/jobs/{jobId} Get inference status and extracted entities</p>				
<p>2. Click on Try it out. For the jobId field, enter the id copied in previous step. Click on Execute.</p>	 <p>Get the inference result</p> <p>Parameters</p> <table border="1"> <tr> <td>Name</td> <td>Description</td> </tr> <tr> <td>jobId * required string (patch)</td> <td>c4c150cd-9c71-43b8-a773-660f79205bd8</td> </tr> </table> <p>Execute</p>	Name	Description	jobId * required string (patch)	c4c150cd-9c71-43b8-a773-660f79205bd8
Name	Description				
jobId * required string (patch)	c4c150cd-9c71-43b8-a773-660f79205bd8				
<p>3. Verify that you get a response of 200. In the response, you will find the prediction for the extracted entities. The prediction is made with a probability indicated by the confidence field which represents how certain the model is about the value provided. A confidence of 1 means that the model is 100% sure about its prediction.</p>	 <p>Code: 200</p> <p>Details</p> <p>Response body</p> <pre>{ "data": { "id": "c4c150cd-9c71-43b8-a773-660f79205bd8", "status": "SUCCESS", "result": [{ "vendorName": [{ "value": "Finance company Pvt Ltd F&A services operated by AKAI Pvt Ltd", "confidence": 0.74 }], "vendorAddress": [{ "value": "The Heights, 116 Glenurquhart Road, BALLAUGH, Surrey. KT27 0XW. UK", "confidence": 0.99 }], "invoiceNo": [], "invoiceDate": [], "vendorBankAccountNo": [], "buyerName": [], "buyerAddress": [], "purchaseOrderNo": [], "paymentTerms": [], "currency": [], "taxRate": [], "taxDescription": [] }] } }</pre>				

Optional exercise

You can also use the following **payload** example with the **sap_invoice_header** model:

JSON
{ "text":"Order number 12345678 Order type Sales Force Order abcdef Date 01/01/2001 Customer number 9876543 Customer service 123/ 456 6789 E-mail eshop@gmail.com Your Reference: Order receipt confirmation Customer address 508 W. St Margarets St. Brooklyn, NY 11228 Billing address 508 W. St Margarets St. Brooklyn, NY 11228 Ordered by Fenton Moon Shipping type: UPS 2nd Day Air PM Terms of payment: 30 Days Net due Your message: Item Material Description Qty. Price per Unit Net steel gray / Length 12 mm / 1 35.00 35.00 Gross Amount USD 35.00 Freight/ packaging USD 15.00 Tax 2 % USD 1 Tax 4 % USD 2 Total amount USD 53", "modelName":"sap_invoice_header", "modelVersion":1 }

www.sap.com/contactsap

© 2020 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See www.sap.com/copyright for additional trademark information and notices.