

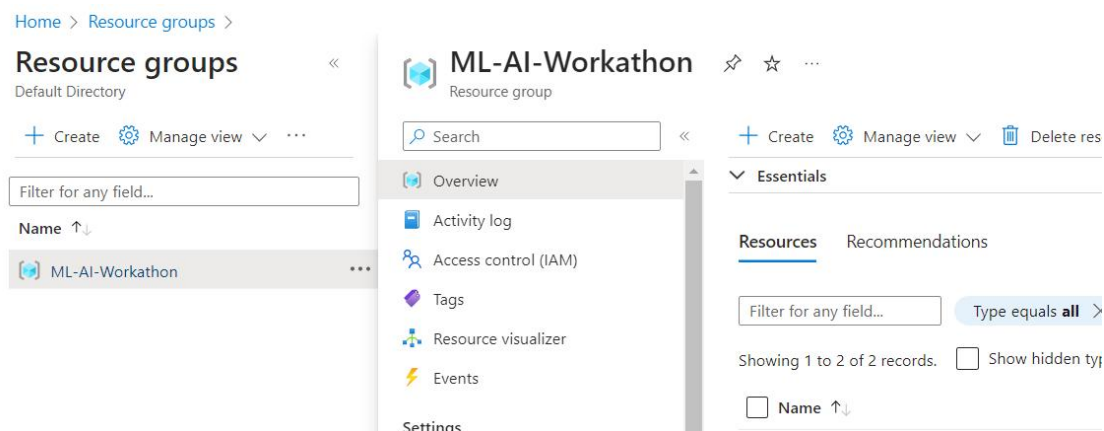
Task report

Task:

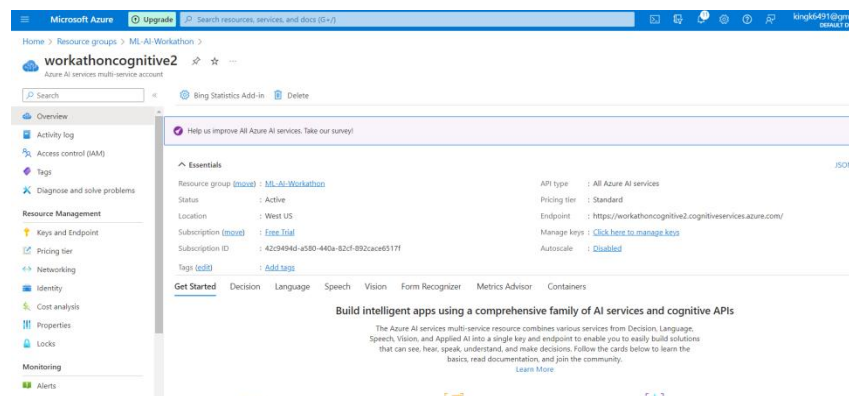
1. Download and install Postman, a free tool that allows you to make API calls.
 - Go to the Postman website and download the desktop application or use the web version.
 - Install Postman on your computer.
2. Create an Azure account or use your current Azure subscription.
 - Go to the Azure website and sign up for a free trial account or use your current Azure subscription.
 - Follow the instructions to create an Azure account.
3. Download the data for training and testing from the Data Folder in the Form Recognizer folder.
 - Go to the Form Recognizer folder and download the data for training and testing.
 - Save the data to a local folder on your computer.
4. Use the GUI Portal to train the model
 - Go to the Azure Portal and sign in to your account.
 - Create a new Form Recognizer resource.
 - Use the GUI Portal to train the model by uploading the training data and following the instructions.
5. Build a custom model for each form type and assimilate them together in a Composed model to make all these custom models work together and not in silos.
 - Use the GUI Portal to build a custom model for each form type.
 - Assimilate the custom models together in a Composed model to make all these custom models work together and not in silos.
6. Deploy the models on the cloud or on the edge.
 - Use the Azure Portal to deploy the models on the cloud or on the edge.
7. Pass a new unprocessed form belonging to any of the custom model categories comprising the composed model to generate the outcome.
8. Test the models and refine them as needed.
 - Test the models and refine them as needed to improve their accuracy and performance.
9. Demo, Develop, and Deploy your own custom use cases.
 - Use the models to develop and deploy your own custom

Procedure

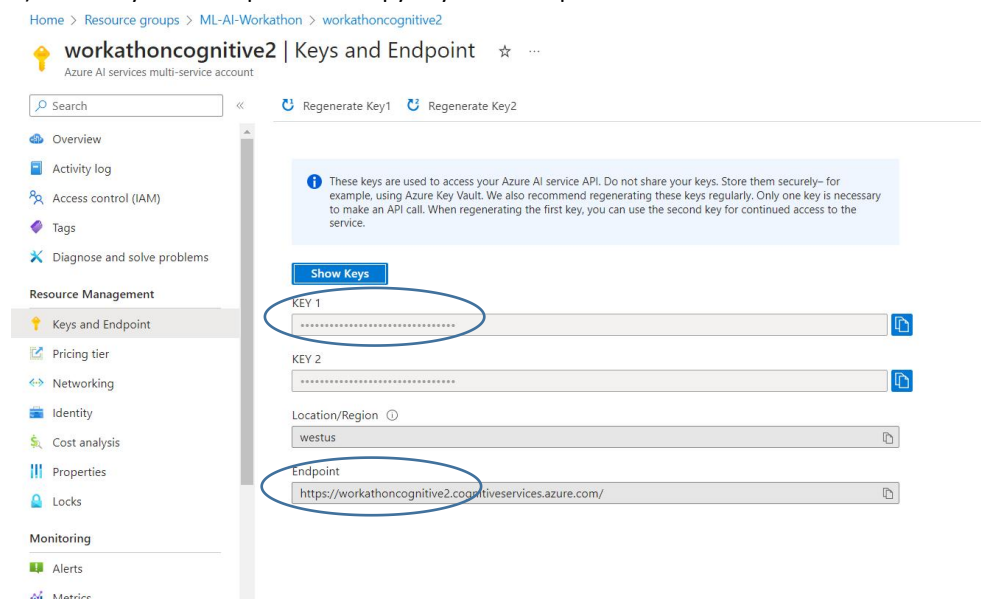
1) Go to azure and create a resource group: ML-AI-Workathon



2) In this resource group create cognitive services: workathoncognitive2



3) Go to keys and endpoint and copy key1 and endpoint



4) Now, create storage account

The screenshot shows the Azure portal interface for a storage account named 'workathonstorage'. The left sidebar contains navigation options like Overview, Activity log, Tags, Diagnose and solve problems, Access Control (IAM), Data migration, Events, Storage browser, Storage Mover, Data storage, Containers, and File shares. The main area displays the 'Essentials' section with details about the resource group (ML-AI-Workathon), location (Central India), subscription (Free Trial), subscription ID, and disk state (Available). It also shows performance (Standard), replication (Locally-redundant storage (LRS)), account kind (StorageV2 (general purpose v2)), provisioning state (Succeeded), and creation time (8/16/2023, 1:04:57 PM). Below this, the 'Properties' section includes 'Blob service' settings (Hierarchical namespace: Disabled, Default access tier: Hot) and 'Security' settings (Require secure transfer for REST API operations: Enabled, Storage account key access: Enabled).

In data storage of the storage account, go to containers and click on the + container button

[Home](#) > [Resource groups](#) > [ML-AI-Workathon](#) > [workathonstorage](#)

The screenshot shows the 'Containers' page for the 'workathonstorage' storage account. The left sidebar is similar to the previous screenshot, but the 'Containers' option is highlighted under 'Data storage'. The main area has a search bar and a '+ Container' button. Below the search bar, there is a 'Search containers by prefix' input field. The 'Access Control (IAM)' option is also visible in the sidebar.

In the name, write formaadhaar and then click on create. Now select this blob storage, click on upload and then select atleast 5 samples of adhaar card from the internet

In this storage account, go to resource sharing (CORS) and enter the following details:

The screenshot shows the 'CORS settings' page for the 'Blob service' of the 'workathonstorage' storage account. The page has a table with columns: Allowed origins, Allowed methods, Allowed headers, Exposed headers, and Max age. The 'Allowed origins' column has a text input field with a plus icon. The 'Allowed methods' column has a dropdown menu with '8 selected'. The 'Allowed headers' column has a text input field with a plus icon. The 'Exposed headers' column has a text input field with a plus icon. The 'Max age' column has a text input field with '200' and a trash icon.

Now, go to the container in data storage, right click on formaadhaar, click generate sas

The screenshot shows the 'Containers' page for the 'workathonstorage' storage account. The left sidebar is similar to the previous screenshot, but the 'Containers' option is highlighted under 'Data storage'. The main area has a search bar and a '+ Container' button. Below the search bar, there is a 'Search containers by prefix' input field. A table lists the containers: '\$logs', 'formaadhaar', and 'formlicense'. The 'formaadhaar' container is selected. A context menu is open over the 'formaadhaar' container, showing options: Container properties, Generate SAS, Access policy, Acquire lease, Break lease, Change access level, and Edit metadata.

Enter the following details. Note: keep the expiry date for a later date. Generate sas and then copy it.

Generate SAS

A shared access signature (SAS) is a URI that grants restricted access to an Azure Storage container. Use it when you want to grant access to storage account resources for a specific time range without sharing your storage account key. [Learn more](#)

Signing method
☒ Account key ☐ User delegation key

Signing key
Key 1

Permissions *
7 selected

Start and expiry date/time

Start
08/08/2021 3:47:47 PM
(UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi

Expiry
06/30/2022 11:47:47 PM
(UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi

Allowed IP addresses
for example, 168.1.5.65 or 168.1.5.65-168.1....

Allowed protocols
☒ HTTPS only ☐ HTTPS and HTTP

Generate SAS token and URL

Blob SAS token
sp=racwdli&st=2021-08-08T10:17:47Z&se=2022-06-30T18:17:47Z&spr=https&sv=202...

Blob SAS URL
https://workathonstorage.blob.core.windows.net/formaadhaar?sp=racwdli&st=2021-0...

5) After completing the above steps, go to <https://fott-2-1.azurewebsites.net/>
Go to connections, and click on the + icon. Enter the following details

Connection Settings

▲ Warning: CORS (Cross Domain Resource Sharing) must be enabled on the Azure Blob Storage account, in order to use it as a source or target connection. More information on enabling CORS can be found in the [Azure Documentation](#).

Display name*
aadhaardata ✓

Description
Aadhaar data for Avengers ✓

Provider
Azure blob container ✓

Azure blob container

SAS URI*
..... ✓

Shared access signature URI to the blob container

Save Connection Cancel

Paste the SAS url of the blob storage you copied earlier.

6) Go to home tab, click on + custom model and enter the following details. In the form recognizer service url: paste the endpoint you copied earlier. In the API key, paste the key copied earlier

Project Settings

Display name*
aadhaamodel ✓

Security token
Generate New Security Token ✓
Used to encrypt sensitive data within project files

Source connection*
aadhaardata ✓ Add Connection
Where to load assets from

Folder path
✓
Subfolder path to your files

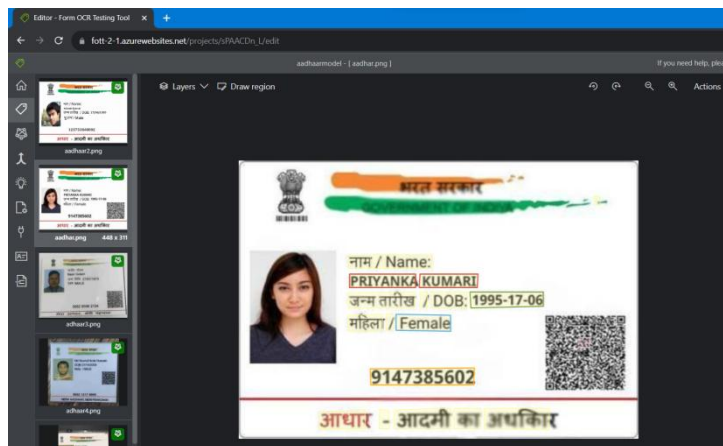
Form recognizer service URI*
https://workathoncognitive2.cognitiveservices.azure.com/ ✓
Form recognizer service URI

API key
***** ✓ 🔑 📄
API key

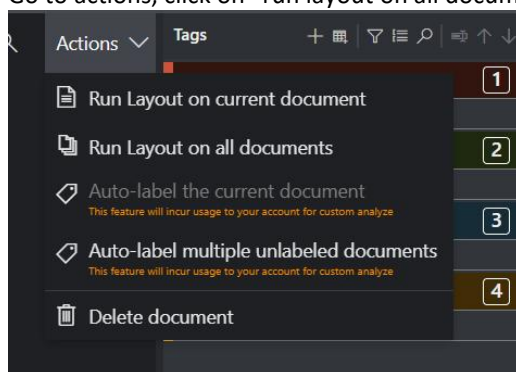
API version

Click on save project

7) Go to tags editor tab, it will look like this:



Go to actions, click on “run layout on all documents”



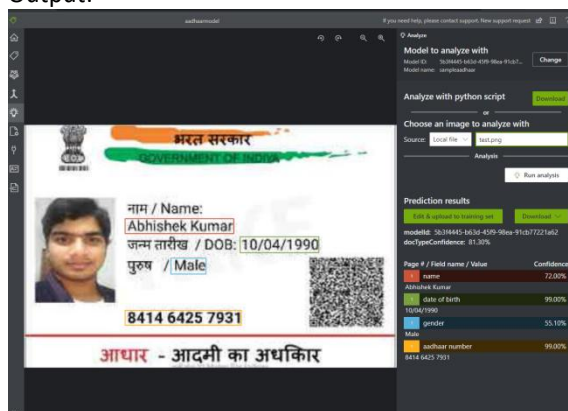
Now click on the plus icon and create the following tags:
Name, date of birth, gender, aadhaar number.

Now click on the name of the person, ex PRIYANKA KUMARI, and then click on name tag you just created. You will notice the name value is displayed under name tag. Continue the same process for other tags

Tags

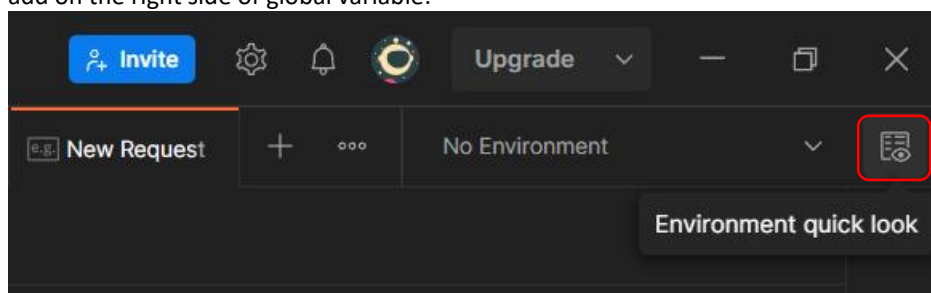
name	1
PRIYANKA KUMARI	
date of birth	2
1995-17-06	
gender	3
Female	
aadhaar number	4
9147385602	

8) Go to train tab. Click on train a new model, enter the name sample aadhaar. Your model is trained
 9) For testing your model, go to analyse tab, keep the source local file and browse the file location. I have given it a file 'test.png'.
 Output:



Model was able to recognise the parameters

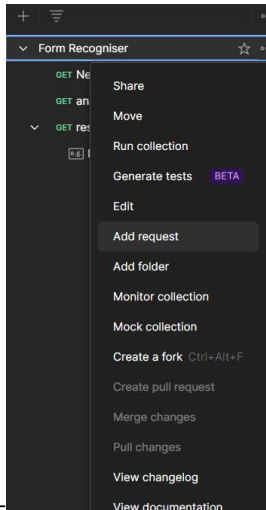
10) Now go to postman (installed earlier). click on the environment quick look button, then click on add on the right side of global variable:



Give the values of endpoint and key you copied earlier and save it:

Filter variables				
Variable	Type	Initial value	Current value	
<input checked="" type="checkbox"/> Endpoint	default		https://workathoncognitive2.cognitiveservices.azure.com/	
<input checked="" type="checkbox"/> Key	default		d8	b274
Add new variable				

11) click on collections then click on New. Give the name: 'form recognizer'
 Right click on form recognizer, then click on add request

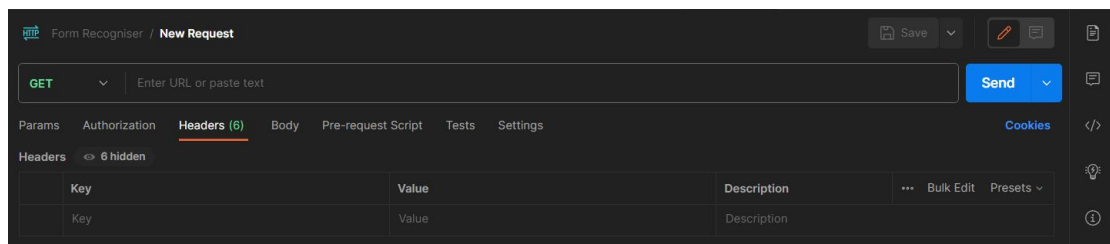


12) You can call this API request to get the details of all the custom & composed models you have created.

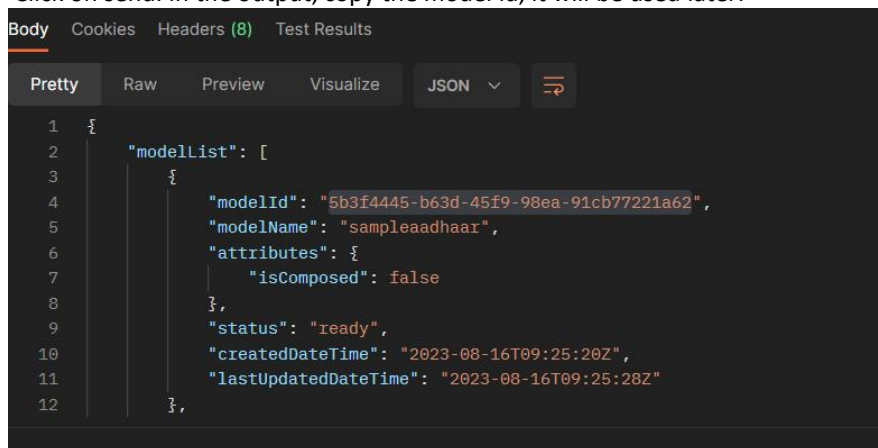
URL : `{{endpoint}}/formrecognizer/v2.1/custom/models`

Headers :

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



Click on send. In the output, copy the model id, it will be used later:



13) Add another request, but this time instead of get, select post

API CALL 1 (POST)

URL 1: `{{endpoint}}/formrecognizer/v2.1/custom/models/9b619e32-a40f-4442-8a83-344557b592a9/analyze? includeTextDetails=true`

Replace the underlined part with the Composed model key you copied in above step.

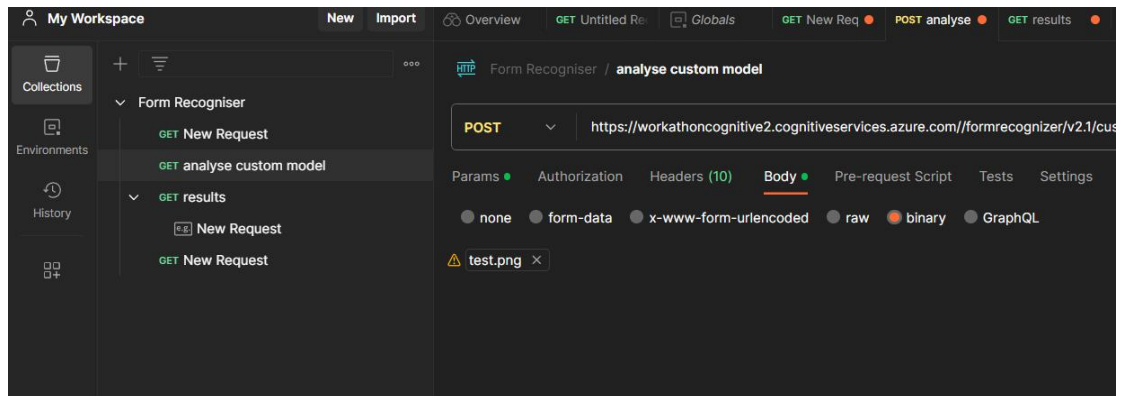
Headers :

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

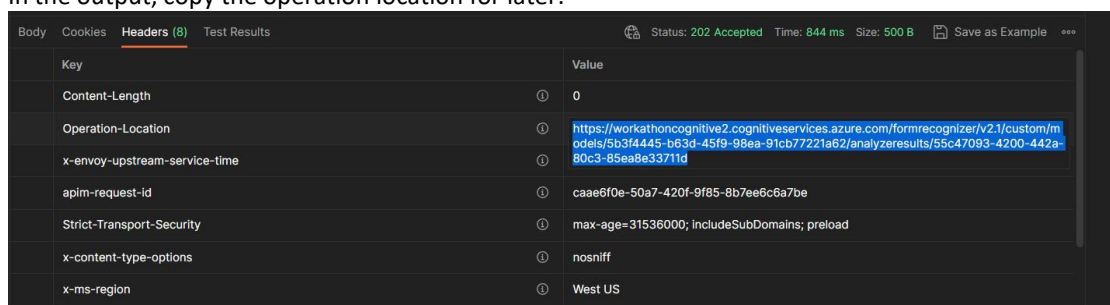
Content-Type : image/png

Body :

Select Binary and upload the test image for either Driver's Licence or Aadhaar card. Click send



In the output, copy the operation location for later:

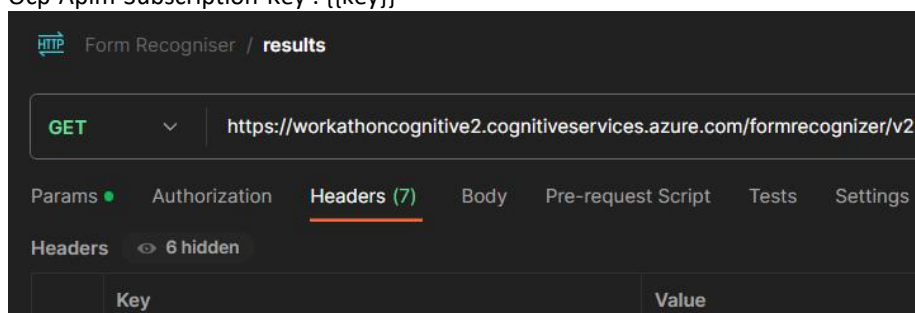


14) Add another GET request,
API CALL 2 (GET)

URL2 : Operation-Location fetched earlier

Headers :

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



Final output:

The image shows two screenshots of a web browser's developer console, specifically the 'Body' tab. The first screenshot displays a JSON object for a 'gender' field. The second screenshot displays a JSON object for an 'aadhaar number' field. Both JSON objects include a 'type', 'valueString', 'text', 'page', and 'boundingBox' (an array of coordinates). The 'confidence' value is 0.551 for the gender field and 0.99 for the aadhaar number field.

```
88 "fields": {
89   "gender": {
90     "type": "string",
91     "valueString": "Male",
92     "text": "Male",
93     "page": 1,
94     "boundingBox": [
95       138.0,
96       125.0,
97       171.0,
98       125.0,
99       171.0,
100      140.0,
101      138.0,
102      140.0
103    ],
104     "confidence": 0.551
105   },
106   "date of birth": {
107     "type": "string",
108     "valueString": "10/04/1990",
109     "text": "10/04/1990",
110     "page": 1,
```

```
122 },
123   "aadhaar number": {
124     "type": "string",
125     "valueString": "8414 6425 7931",
126     "text": "8414 6425 7931",
127     "page": 1,
128     "boundingBox": [
129       95.0,
130       174.0,
131       204.0,
132       174.0,
133       204.0,
134       188.0,
135       95.0,
136       188.0
137     ],
138     "confidence": 0.99
139   },
140   "name": {
141     "type": "string",
142     "valueString": "Abhishek Kumar",
143     "text": "Abhishek Kumar",
144     "page": 1,
```

you will get all the details.