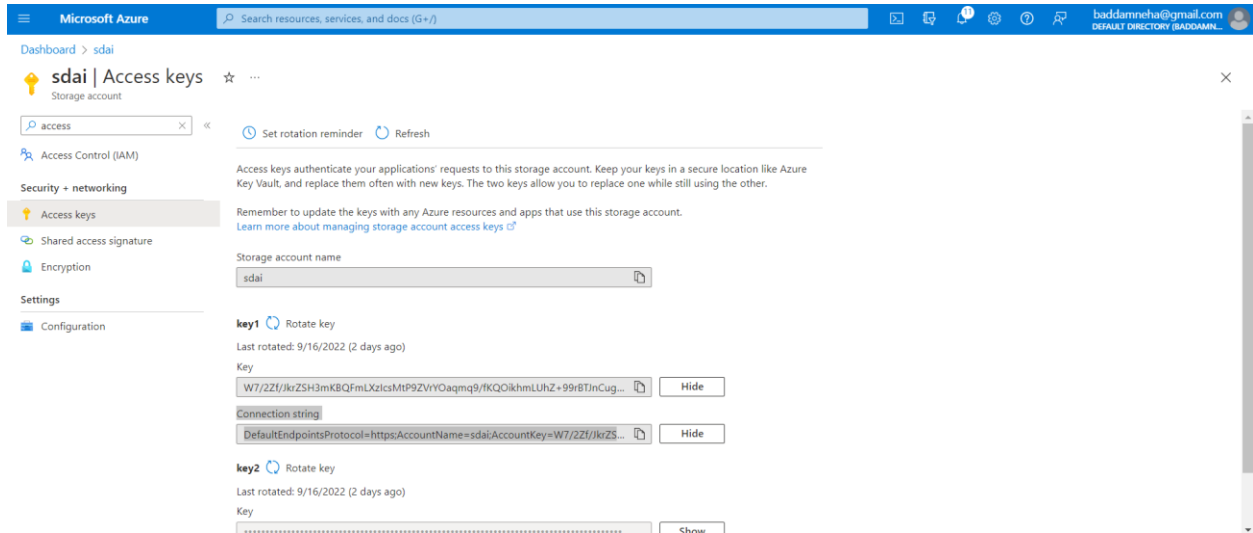


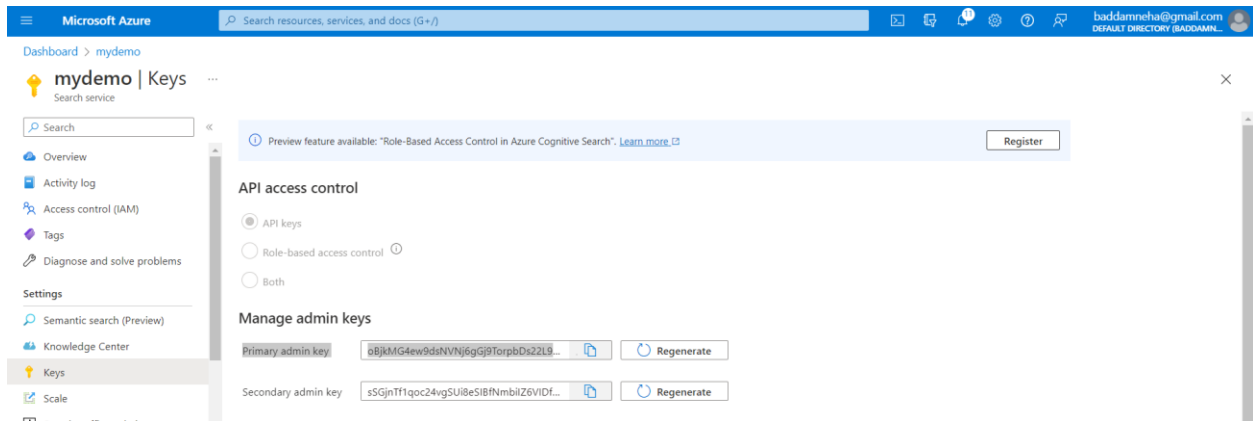
Pre-requisites to execute the code:

Connection string:



The screenshot shows the 'Access keys' page for a storage account named 'sdai'. The page includes a sidebar with navigation options like 'Access Control (IAM)', 'Security + networking', 'Shared access signature', 'Encryption', 'Settings', and 'Configuration'. The main content area displays instructions on how to use access keys, a 'Set rotation reminder' button, and a 'Refresh' button. It also shows the 'Storage account name' as 'sdai'. Under the 'key1' section, it displays the 'Last rotated' date as '9/16/2022 (2 days ago)', the 'Key' value, and the 'Connection string' value. The 'key2' section also shows the 'Last rotated' date and the 'Key' value.

Api-key:



The screenshot shows the 'Keys' page for a search service named 'mydemo'. The page includes a sidebar with navigation options like 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Settings', 'Semantic search (Preview)', 'Knowledge Center', 'Keys', and 'Scale'. The main content area displays a 'Preview feature available: "Role-Based Access Control in Azure Cognitive Search". Learn more' message with a 'Register' button. Below this, it shows the 'API access control' section with radio buttons for 'API keys', 'Role-based access control', and 'Both'. The 'Manage admin keys' section displays the 'Primary admin key' and 'Secondary admin key' values, each with a 'Regenerate' button.

Blob service containers:

Dashboard > sdai

sdai | Containers

Storage account

Search

+ Container Change access level Restore containers Refresh Delete

Search containers by prefix

Show deleted containers

Name	Last modified	Public access level	Lease state	
<input type="checkbox"/> \$blobchangefeed	9/16/2022, 12:17:47 PM	Private	Available	...
<input type="checkbox"/> \$logs	9/16/2022, 12:17:47 PM	Private	Available	...
<input checked="" type="checkbox"/> cog-search-demo	9/16/2022, 12:19:00 PM	Private	Available	...
<input checked="" type="checkbox"/> new-blob-service	9/18/2022, 8:59:42 PM	Private	Available	...

Microsoft Azure

Search resources, services, and docs (G+)

baddamneha@gmail.com
DEFAULT DIRECTORY (BADDAMN...

Dashboard > sdai | Containers

cog-search-demo

Container

s

Upload Change access level Refresh Delete Change tier Acquire lease Break lease View snapshots Create snapshot

Add filter

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state	
<input type="checkbox"/> 10-K-FY16.html	9/16/2022, 12:19:26 ...	Hot (Inferred)		Block blob	1.79 MiB	Available	...
<input type="checkbox"/> 5074.clip_image002_6FE27E85.png	9/16/2022, 12:19:20 ...	Hot (Inferred)		Block blob	471.02 KiB	Available	...
<input type="checkbox"/> Cognitive Services and Content Intelligence.pptx	9/16/2022, 12:19:38 ...	Hot (Inferred)		Block blob	10.97 MiB	Available	...
<input type="checkbox"/> Cognitive Services and Bots (spanish).pdf	9/16/2022, 12:19:29 ...	Hot (Inferred)		Block blob	3.79 MiB	Available	...
<input type="checkbox"/> create-search-collect-info.png	9/16/2022, 12:19:19 ...	Hot (Inferred)		Block blob	56.18 KiB	Available	...
<input type="checkbox"/> create-search-service.png	9/16/2022, 12:19:19 ...	Hot (Inferred)		Block blob	25.27 KiB	Available	...
<input type="checkbox"/> create-service-full-portal.png	9/16/2022, 12:19:19 ...	Hot (Inferred)		Block blob	66.82 KiB	Available	...
<input type="checkbox"/> guthrie.jpg	9/16/2022, 12:19:19 ...	Hot (Inferred)		Block blob	45.08 KiB	Available	...
<input type="checkbox"/> MSFT_cloud_architecture_contoso.pdf	9/16/2022, 12:19:26 ...	Hot (Inferred)		Block blob	1.76 MiB	Available	...
<input type="checkbox"/> MSFT_FY17_10K.docx	9/16/2022, 12:19:22 ...	Hot (Inferred)		Block blob	674.23 KiB	Available	...
<input type="checkbox"/> NYSE_LNKD_2015.PDF	9/16/2022, 12:19:20 ...	Hot (Inferred)		Block blob	393.43 KiB	Available	...
<input type="checkbox"/> redshirt.jpg	9/16/2022, 12:19:20 ...	Hot (Inferred)		Block blob	66.74 KiB	Available	...
<input type="checkbox"/> satyanadellinux.jpg	9/16/2022, 12:19:21 ...	Hot (Inferred)		Block blob	107.92 KiB	Available	...
<input type="checkbox"/> satyasletter.txt	9/16/2022, 12:19:20 ...	Hot (Inferred)		Block blob	5.8 KiB	Available	...

Microsoft Azure

Search resources, services, and docs (G+)

baddamneha@gmail.com
DEFAULT DIRECTORY (BADDAMN...

Home > sdai | Containers

new-blob-service

Container

Search

Upload Change access level Refresh Delete Change tier Acquire lease Break lease View snapshots Create snapshot

Authentication method: Access key (Switch to Azure AD User Account)

Location: new-blob-service

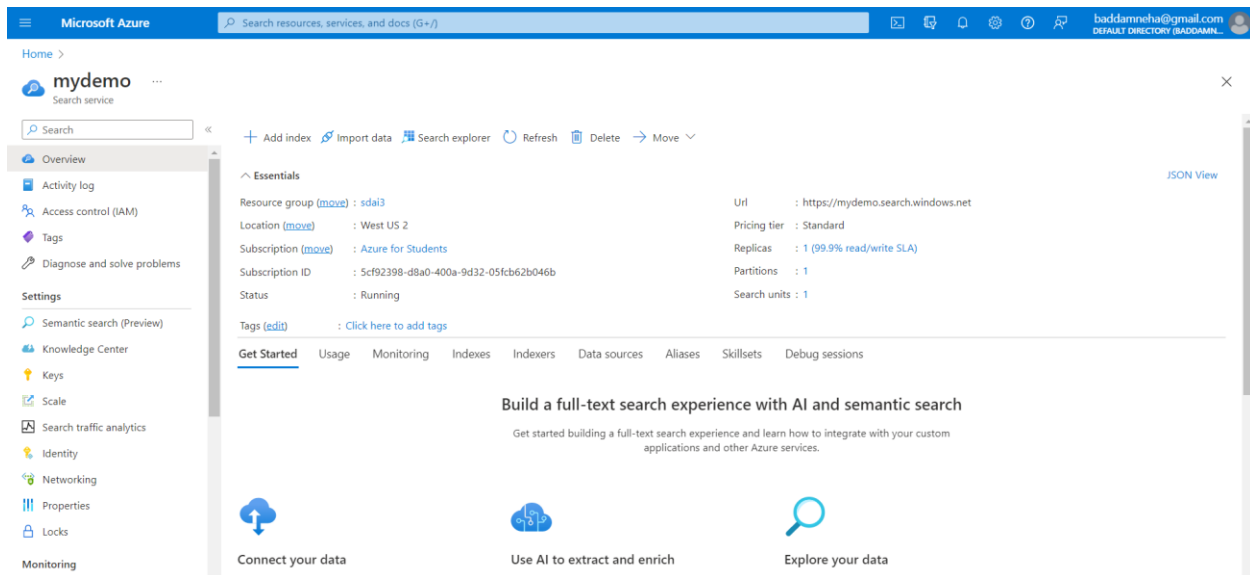
Search blobs by prefix (case-sensitive)

Show deleted blobs

Add filter

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state	
<input type="checkbox"/> auto-parts-p-img2-1.jpg	9/18/2022, 9:25:22 PM	Hot (Inferred)		Block blob	78.38 KiB	Available	...
<input type="checkbox"/> auto-parts-p-img2.jpg	9/18/2022, 9:25:21 PM	Hot (Inferred)		Block blob	55.37 KiB	Available	...
<input type="checkbox"/> Azure Search Pricing Table.xlsx	9/18/2022, 9:23:23 PM	Hot (Inferred)		Block blob	11.72 KiB	Available	...
<input type="checkbox"/> azure-featured.png	9/18/2022, 9:26:49 PM	Hot (Inferred)		Block blob	22.01 KiB	Available	...
<input type="checkbox"/> azure.png	9/18/2022, 9:26:49 PM	Hot (Inferred)		Block blob	288.39 KiB	Available	...
<input type="checkbox"/> Deployment strategies and best practices.pdf	9/18/2022, 9:23:24 PM	Hot (Inferred)		Block blob	394.87 KiB	Available	...
<input type="checkbox"/> FAQ - Azure Search.pdf	9/18/2022, 9:23:24 PM	Hot (Inferred)		Block blob	160.58 KiB	Available	...
<input type="checkbox"/> index.html	9/18/2022, 9:25:03 PM	Hot (Inferred)		Block blob	548 B	Available	...
<input type="checkbox"/> Pricing details.docx	9/18/2022, 9:23:23 PM	Hot (Inferred)		Block blob	16.16 KiB	Available	...
<input type="checkbox"/> TDSP - GAIC Jan 2018.pptx	9/18/2022, 9:23:32 PM	Hot (Inferred)		Block blob	8.18 MiB	Available	...

Search service:



Running existing code in the tutorial

Initially we imported json, requests and pprint packages. Then we named data source, skillset, index and indexer. Then we setup the endpoint and header. Endpoint has the url of the service. Header has the api-key, that we get from the keys of the “mydemo” search service.

```
PythonTutorial-AzureSearch-AIEnrichment.ipynb
C:\Users\badba> Downloads > PythonTutorial-AzureSearch-AIEnrichment.ipynb > M Enrichment sample in Python for Azure Cognitive Search > import json
+ Code + Markdown + Run All + Clear Outputs of All Cells + Restart + Interrupt + Variables + Outline ... Python 3.10.5 64-bit

import json
import requests
from pprint import pprint

[1] ✓ 0.2s

# Define the names for the data source, skillset, index and indexer
datasource_name = "cogsrch-py-datasource"
skillset_name = "cogsrch-py-skillset"
index_name = "cogsrch-py-index"
indexer_name = "cogsrch-py-indexer"

[2] ✓ 0.4s

# Setup the endpoint
endpoint = 'https://mydemo.search.windows.net'
headers = {'Content-Type': 'application/json',
           'api-key': 'oBjKMG4ew9dsNwJeggj9TorpbDs22L9GQJ7DHTVHAzSeCRBmDe'}
params = {
    'api-version': '2020-06-30'
}

[7] ✓ 0.1s
```

Now we have created data source, that points to the azure storage account, that has the blob service that contains data. Then we print the request status code. Here it is 201 which indicates success.

```
# Create a data source
# This data source points to your Azure Storage account.
# You should already have a blob container that contains the sample data
datasourceConnectionString = "DefaultEndpointsProtocol=https;AccountName=sdal;AccountKey=W7/2Zf/3krZSH3mKBQfmlXz1csMTP9ZVrY0a9mq9/fKQ0ikhamLUhZ499rBTJnCugC6hTKPoJlNKg+ASTESUgzw==;End
datasource_payload = {
    "name": datasource_name,
    "description": "Demo files to demonstrate cognitive search capabilities.",
    "type": "azureblob",
    "credentials": {
        "connectionString": datasourceConnectionString
    },
    "container": {
        "name": "cog-search-demo"
    }
}
r = requests.put(endpoint + "/datasources/" + datasource_name,
                 data=json.dumps(datasource_payload), headers=headers, params=params)
print(r.status_code)
```

[0] ✓ 0.7s Python

.. 201

Now we create a skillset. Then we print the request status code. Here it is 201 which indicates success.

```
# Create a skillset
skillset_payload = {
    "name": skillset_name,
    "description":
        "Extract entities, detect language and extract key-phrases",
    "skills":
        [
            {
                "@odata.type": "#Microsoft.Skills.Text.EntityRecognitionSkill",
                "categories": ["Organization"],
                "defaultLanguageCode": "en",
                "inputs": [
                    {
                        "name": "text",
                        "source": "/document/content"
                    }
                ],
                "outputs": [
                    {
                        "name": "organizations",
                        "targetName": "organizations"
                    }
                ]
            },
            {
                "@odata.type": "#Microsoft.Skills.Text.LanguageDetectionSkill",
                "inputs": [
                    {
                        "name": "text",
                        "source": "/document/content"
                    }
                ],
                "outputs": [
                    {
                        "name": "languageCode",
                        "targetName": "languageCode"
                    }
                ]
            },
            {
                "@odata.type": "#Microsoft.Skills.Text.SplitsSkill",
                "textSplitMode": "pages",
                "maximumPageLength": 4000,
                "inputs": [
                    {
                        "name": "text",
                        "source": "/document/content"
                    },
                    {
                        "name": "languageCode",
                        "source": "/document/languageCode"
                    }
                ],
                "outputs": [
                    {
                        "name": "textItems",
                        "targetName": "pages"
                    }
                ]
            },
            {
                "@odata.type": "#Microsoft.Skills.Text.KeyPhraseExtractionSkill",
                "context": "/document/pages/*",
                "inputs": [
                    {
                        "name": "text",
                        "source": "/document/pages/*"
                    },
                    {
                        "name": "languageCode",
                        "source": "/document/languageCode"
                    }
                ],
                "outputs": [
                    {
                        "name": "keyPhrases",
                        "targetName": "keyPhrases"
                    }
                ]
            }
        ]
    }

r = requests.put(endpoint + "/skillsets/" + skillset_name,
                 data=json.dumps(skillset_payload), headers=headers, params=params)
print(r.status_code)
```

[9] ✓ 0.6s Python

.. 201

Now we create an index. Then we print the request status code. Here it is 201 which indicates success.

```
# Create an index
# Queries operate over the searchable fields and filterable fields in the index
index_payload = {
    "name": index_name,
    "fields": [
        {
            "name": "id",
            "type": "Edm.String",
            "key": "true",
            "searchable": "true",
            "filterable": "false",
            "facettable": "false",
            "sortable": "true"
        },
        {
            "name": "content",
            "type": "Edm.String",
            "sortable": "false",
            "searchable": "true",
            "filterable": "false",
            "facettable": "false"
        },
        {
            "name": "languageCode",
            "type": "Edm.String",
            "searchable": "true",
            "filterable": "false",
            "facettable": "false"
        },
        {
            "name": "keyPhrases",
            "type": "Collection(Edm.String)",
            "searchable": "true",
            "filterable": "false",
            "facettable": "false",
            "sortable": "false"
        },
        {
            "name": "organizations",
            "type": "Collection(Edm.String)",
            "searchable": "true",
            "sortable": "false",
            "filterable": "false",
            "facettable": "false"
        }
    ]
}

r = requests.put(endpoint + "/indexes/" + index_name,
                 data=json.dumps(index_payload), headers=headers, params=params)
print(r.status_code)
```

[10] ✓ 1.1s Python

... 201

Now we create an indexer. Then we print the request status code. Here it is 201 which indicates success.

```
# Create an indexer
indexer_payload = {
    "name": indexer_name,
    "dataSourceName": datasource_name,
    "targetIndexName": index_name,
    "skillsetName": skillset_name,
    "fieldMappings": [
        {
            "sourceFieldName": "metadata_storage_path",
            "targetFieldName": "id",
            "mappingFunction": {
                "name": "base64Encode"
            }
        },
        {
            "sourceFieldName": "content",
            "targetFieldName": "content"
        }
    ],
    "outputFieldMappings": [
        {
            "sourceFieldName": "/document/organizations",
            "targetFieldName": "organizations"
        },
        {
            "sourceFieldName": "/document/pages/*/keyphrases/*",
            "targetFieldName": "keyphrases"
        },
        {
            "sourceFieldName": "/document/languageCode",
            "targetFieldName": "languageCode"
        }
    ]
},
{
    "parameters": {
        "maxFailedItems": -1,
        "maxFailedItemsPerBatch": -1,
        "configuration": {
            "dataToExtract": "contentAndMetadata",
            "imageAction": "generateNormalizedImages"
        }
    }
}

r = requests.put(endpoint + "/indexers/" + indexer_name,
                 data=json.dumps(indexer_payload), headers=headers, params=params)
print(r.status_code)
```

[11] ✓ 1m 0.6s Python

... 201

Now we print the indexer status.

```
# Optionally, get indexer status to confirm that it's running
r = requests.get(endpoint + "/indexers/" + indexer_name +
                 "/status", headers=headers, params=params)
pprint(json.dumps(r.json(), indent=1))
```

[12] ✓ 0.5s Python

```
{
  '@odata.context': 'https://mydemo.search.windows.net/$metadata#Microsoft.Azure.Search.V2020_06_30.IndexerExecutionInfo',
  'name': 'cogsrch-py-indexer',
  'status': 'running',
  'lastResult': {
    'status': 'inProgress'
  },
  'errorMessage': null,
  'startTime': '2022-09-16T18:46:10.243Z',
  'endTime': null,
  'itemsProcessed': 0,
  'itemsFailed': 0,
  'initialTrackingState': null,
  'finalTrackingState': null,
  'errors': [],
  'warnings': [],
  'metrics': null,
},
{
  'executionHistory': [],
  'limits': {
    'maxRunTime': 'PT2H',
    'maxDocumentExtractionSize': 134217728,
    'maxDocumentContentCharactersToExtract': 4194304
  }
}
```

Now we get the service for the index definition.

```
# Query the service for the index definition
# Query responses can be verbose. If you get "Output exceeds the size limit. Open the full output data in a text editor", open the output in an editor.
r = requests.get(endpoint + "/indexes/" + index_name,
                  headers=headers, params=params)
pprint(json.dumps(r.json(), indent=1))

[13] ✓ 0.4s Python

... Output exceeds the size limit. Open the full output data in a text editor.
({
  '@odata.context': 'https://mydemo.search.windows.net/$metadata#indexes/$entity',
  '@odata.etag': '"0x8DA981392B30412\\\""',
  'name': 'cogsrch-py-index',
  'defaultScoringProfile': null,
  'fields': [
    {
      'name': 'id',
      'type': 'Edm.String',
      'searchable': true,
      'filterable': false,
      'retrievable': true,
      'sortable': true,
      'facetable': false,
      'key': true,
      'indexAnalyzer': null,
      'searchAnalyzer': null,
      'analyzer': null,
      'synonymMaps': []
    },
    {
      'name': 'content',
      'type': 'Edm.String',
      'searchable': true,
      'filterable': false,
      'retrievable': true,
      'sortable': false,
      'facetable': false,
      'key': false,
      'indexAnalyzer': null,
      'searchAnalyzer': null,
      'analyzer': null,
      'synonymMaps': []
    },
    {
      'name': 'languageCode',
      'type': 'Edm.String',
      'searchable': false,
      'filterable': false,
      'retrievable': true,
      'sortable': false,
      'facetable': false,
      'key': false,
      'indexAnalyzer': null,
      'searchAnalyzer': null,
      'analyzer': null,
      'synonymMaps': []
    }
  ]
})
```

Below is the full output:

```
PythonTutorial-AzureSearch-AIEnrichment.Ipynb PythonTutorial-AzureSearch-AIEnrichment.Ipynb (output)

1 ({
2   '@odata.context': 'https://mydemo.search.windows.net/$metadata#indexes/$entity',
3   '@odata.etag': '"0x8DA981392B30412\\\""',
4   'name': 'cogsrch-py-index',
5   'defaultScoringProfile': null,
6   'fields': [
7     {
8       'name': 'id',
9       'type': 'Edm.String',
10      'searchable': true,
11      'filterable': false,
12      'retrievable': true,
13      'sortable': true,
14      'facetable': false,
15      'key': true,
16      'indexAnalyzer': null,
17      'searchAnalyzer': null,
18      'analyzer': null,
19      'synonymMaps': []
20    },
21    {
22      'name': 'content',
23      'type': 'Edm.String',
24      'searchable': true,
25      'filterable': false,
26      'retrievable': true,
27      'sortable': false,
28      'facetable': false,
29      'key': false,
30      'indexAnalyzer': null,
31      'searchAnalyzer': null,
32      'analyzer': null,
33      'synonymMaps': []
34    },
35    {
36      'name': 'languageCode',
37      'type': 'Edm.String',
38      'searchable': false,
39      'filterable': false,
40      'retrievable': true,
41      'sortable': false,
42      'facetable': false,
43      'key': false,
44      'indexAnalyzer': null,
45      'searchAnalyzer': null,
46      'analyzer': null,
47      'synonymMaps': []
48    }
49  ]
50 })
```



```
PythonTutorial-AzureSearch-AIEnrichment.ipynb PythonTutorial-AzureSearch-AIEnrichment.ipynb (output)
39     "searchable": true,\n
40     "filterable": false,\n
41     "retrievable": true,\n
42     "sortable": true,\n
43     "facettable": false,\n
44     "key": false,\n
45     "indexAnalyzer": null,\n
46     "searchAnalyzer": null,\n
47     "analyzer": null,\n
48     "synonymMaps": []\n
49 },\n
50 {\n
51     "name": "keyPhrases",\n
52     "type": "collection(Edm.String)",\n
53     "searchable": true,\n
54     "filterable": false,\n
55     "retrievable": true,\n
56     "sortable": false,\n
57     "facettable": false,\n
58     "key": false,\n
59     "indexAnalyzer": null,\n
60     "searchAnalyzer": null,\n
61     "analyzer": null,\n
62     "synonymMaps": []\n
63 },\n
64 {\n
65     "name": "organizations",\n
66     "type": "collection(Edm.String)",\n
67     "searchable": true,\n
68     "filterable": false,\n
69     "retrievable": true,\n
70     "sortable": false,\n
71     "facettable": false,\n
72     "key": false,\n
73     "indexAnalyzer": null,\n
74     "searchAnalyzer": null,\n
75     "analyzer": null,\n
76     "synonymMaps": []\n
77 },\n
78 },\n
79 "scoringProfiles": [],\n
80 "corsOptions": null,\n
81 "suggesters": [],\n
82 "analyzers": [],\n
83 "tokenizers": [],\n
84 "tokenFilters": [],\n
85 "charFilters": [],\n
86 "encryptionKey": null,\n
87 "similarity": {\n
88     "@odata.type": "#Microsoft.Azure.Search.BM25Similarity",\n
89     "k1": null,\n
90     "b": null\n
91 },\n
92 '})\n
93
```

Now we query the index to return contents of organizations

```
# Query the index to return the contents of "organizations", created through Entity Recognition during enrichment
# For keyword search, replace the asterisk with comma-separated query terms: search=microsoft,azure
r = requests.get(endpoint + "/indexes/" + index_name +
                 "/docs?search=&$select=organizations", headers=headers, params=params)
pprint(json.dumps(r.json(), indent=1))

[14] ✓ 0.6s Python

... ('{\n
    '@odata.context': '\n
    https://mydemo.search.windows.net/indexes(\'cogsrch-py-index\')/$metadata#docs(*)',\n
    'value': []\n
}')

```

Running code in the tutorial with new data

Below is the data set in the new blob container:

Microsoft Azure | Search resources, services, and docs (G+)

Home > sdai | Containers >

new-blob-service Container

Search | Upload | Change access level | Refresh | Delete | Change tier | Acquire lease | Break lease | View snapshots | Create snapshot

Overview | Diagnose and solve problems | Access Control (IAM) | Settings

Shared access tokens | Access policy | Properties | Metadata

Authentication method: Access key (Switch to Azure AD User Account)
Location: new-blob-service

Search blobs by prefix (case-sensitive) | Show deleted blobs

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
<input type="checkbox"/> auto-parts-p-img2-1.jpg	9/18/2022, 9:25:22 PM	Hot (Inferred)		Block blob	78.38 KIB	Available ***
<input type="checkbox"/> auto-parts-p-img2.jpg	9/18/2022, 9:25:21 PM	Hot (Inferred)		Block blob	55.37 KIB	Available ***
<input type="checkbox"/> Azure Search Pricing Table.xlsx	9/18/2022, 9:23:23 PM	Hot (Inferred)		Block blob	11.72 KIB	Available ***
<input type="checkbox"/> azure-featured.png	9/18/2022, 9:26:49 PM	Hot (Inferred)		Block blob	22.01 KIB	Available ***
<input type="checkbox"/> azure.png	9/18/2022, 9:26:49 PM	Hot (Inferred)		Block blob	288.39 KIB	Available ***
<input type="checkbox"/> Deployment strategies and best practices.pdf	9/18/2022, 9:23:24 PM	Hot (Inferred)		Block blob	394.87 KIB	Available ***
<input type="checkbox"/> FAQ - Azure Search.pdf	9/18/2022, 9:23:24 PM	Hot (Inferred)		Block blob	160.58 KIB	Available ***
<input type="checkbox"/> index.html	9/18/2022, 9:25:03 PM	Hot (Inferred)		Block blob	548 B	Available ***
<input type="checkbox"/> Pricing details.docx	9/18/2022, 9:23:23 PM	Hot (Inferred)		Block blob	16.16 KIB	Available ***
<input type="checkbox"/> TDSF - GAIC Jan 2018.pptx	9/18/2022, 9:23:32 PM	Hot (Inferred)		Block blob	8.18 MiB	Available ***

Initially we imported json, requests and pprint packages. Then we named data source, skillset, index and indexer. Then we setup the endpoint and header. Endpoint has the url of the service. Header has the api-key, that we get from the keys of the “mydemo” search service.

```
[1] ✓ 0.1s Python
import json
import requests
from pprint import pprint

[2] ✓ 0.2s Python
# Define the names for the data source, skillset, index and indexer
datasource_name = "cogsrch-py-datasource-new"
skillset_name = "cogsrch-py-skillset-new"
index_name = "cogsrch-py-index-new"
indexer_name = "cogsrch-py-indexer-new"

[3] ✓ 0.4s Python
# Setup the endpoint
endpoint = 'https://mydemo.search.windows.net'
headers = {'Content-Type': 'application/json',
           'api-key': 'oBjKMG4ew0dsNwJ6G5j9TorpBDS22L9GWQJ7DHTVHAzSeCR8mDe'}
params = {
    'api-version': '2020-06-30'
}
```

Now we have created data source, that points to the azure storage account, that has the blob service that contains data. Then we print the request status code. Here it is 201 which indicates success.

```
[4] ✓ 1.3s Python
# Create a data source
# This data source points to your Azure Storage account.
# You should already have a blob container that contains the sample data
datasourceConnectionString = "DefaultEndpointsProtocol=https;AccountName=sdai;AccountKey=w7/2Zf/JkrZSH3mKBQFmLXzIcsMTP9ZVrY0aqmqP/fKQ0ikhmLUhZ+99rBTJnCugC6htKPo3JNKG+ASTEsUgZw==;End
datasource_payload = {
    "name": datasource_name,
    "description": "Demo files to demonstrate cognitive search capabilities.",
    "type": "azureblob",
    "credentials": {
        "connectionString": datasourceConnectionString
    },
    "container": {
        "name": "new-blob-service"
    }
}
r = requests.put(endpoint + "/datasources/" + datasource_name,
                data=json.dumps(datasource_payload), headers=headers, params=params)
print(r.status_code)

... 201
```

Now we create a skillset. Then we print the request status code. Here it is 201 which indicates success.

```
# Create a skillset
skillset_payload = {
    "name": skillset_name,
    "description":
        "Extract entities, detect language and extract key-phrases",
    "skills":
        [
            {
                "@odata.type": "##Microsoft.Skills.Text.V3.EntityRecognitionSkill",
                "categories": ["Organization"],
                "defaultLanguageCode": "en",
                "inputs": [
                    {
                        "name": "text",
                        "source": "/document/content"
                    }
                ],
                "outputs": [
                    {
                        "name": "organizations",
                        "targetName": "organizations"
                    }
                ]
            },
            {
                "@odata.type": "##Microsoft.Skills.Text.LanguageDetectionSkill",
                "inputs": [
                    {
                        "name": "text",
                        "source": "/document/content"
                    }
                ],
                "outputs": [
                    {
                        "name": "languageCode",
                        "targetName": "languageCode"
                    }
                ]
            },
            {
                "@odata.type": "##Microsoft.Skills.Text.SplitSkill",
                "textSplitMode": "pages",
                "maximumPageLength": 4000,
                "inputs": [
                    {
                        "name": "text",
                        "source": "/document/content"
                    },
                    {
                        "name": "languageCode",
                        "source": "/document/languageCode"
                    }
                ],
                "outputs": [
                    {
                        "name": "textItems",
                        "targetName": "pages"
                    }
                ]
            },
            {
                "@odata.type": "##Microsoft.Skills.Text.KeyPhraseExtractionSkill",
                "context": "/document/pages/**",
                "inputs": [
                    {
                        "name": "text",
                        "source": "/document/pages/**"
                    },
                    {
                        "name": "languageCode",
                        "source": "/document/languageCode"
                    }
                ],
                "outputs": [
                    {
                        "name": "keyPhrases",
                        "targetName": "keyPhrases"
                    }
                ]
            }
        ]
    }

r = requests.put(endpoint + "/skillsets/" + skillset_name,
                 data=json.dumps(skillset_payload), headers=headers, params=params)
print(r.status_code)
```

(5) ✓ 2.9s

Python

... 201

Now we create an index. Then we print the request status code. Here it is 201 which indicates success.

```
# Create an index
# Queries operate over the searchable fields and filterable fields in the index
index_payload = {
    "name": index_name,
    "fields": [
        {
            "name": "id",
            "type": "Edm.String",
            "key": "true",
            "searchable": "true",
            "filterable": "false",
            "facettable": "false",
            "sortable": "true"
        },
        {
            "name": "content",
            "type": "Edm.String",
            "sortable": "false",
            "searchable": "true",
            "filterable": "false",
            "facettable": "false"
        },
        {
            "name": "languageCode",
            "type": "Edm.String",
            "searchable": "true",
            "filterable": "false",
            "facettable": "false"
        },
        {
            "name": "keyPhrases",
            "type": "collection(Edm.String)",
            "searchable": "true",
            "filterable": "false",

```

```
            "filterable": "false",
            "facettable": "false"
        },
        {
            "name": "organizations",
            "type": "collection(Edm.String)",
            "searchable": "true",
            "sortable": "false",
            "filterable": "false",
            "facettable": "false"
        }
    ]
}

r = requests.put(endpoint + "/indexes/" + index_name,
                 data=json.dumps(index_payload), headers=headers, params=params)
print(r.status_code)
```

[6] ✓ 1.1s Python

... 201

Now we create an indexer. Then we print the request status code. Here it is 201 which indicates success.

```
# Create an indexer
indexer_payload = {
    "name": indexer_name,
    "dataSourceName": datasource_name,
    "targetIndexName": index_name,
    "skillsetName": skillset_name,
    "fieldMappings": [
        {
            "sourceFieldName": "metadata_storage_path",
            "targetFieldName": "id",
            "mappingFunction": {
                "name": "base64Encode"
            }
        },
        {
            "sourceFieldName": "content",
            "targetFieldName": "content"
        }
    ],
    "outputFieldMappings": [
        {
            "sourceFieldName": "/document/organizations",
            "targetFieldName": "organizations"
        },
        {
            "sourceFieldName": "/document/pages/*/keyPhrases/*",
            "targetFieldName": "keyPhrases"
        },
        {
            "sourceFieldName": "/document/languageCode",
            "targetFieldName": "languageCode"
        }
    ]
},
```

```
    "parameters": {
        "maxFailedItems": -1,
        "maxFailedItemsPerBatch": -1,
        "configuration": {
            "dataToExtract": "contentAndMetadata",
            "imageAction": "generateNormalizedImages"
        }
    }

r = requests.put(endpoint + "/indexers/" + indexer_name,
                 data=json.dumps(indexer_payload), headers=headers, params=params)
print(r.status_code)
```

[7] ✓ 23.3s

Python

... 201

Now we print the indexer status.

```
# Optionally, get indexer status to confirm that it's running
r = requests.get(endpoint + "/indexers/" + indexer_name +
                 "/status", headers=headers, params=params)
pprint(json.dumps(r.json(), indent=1))
```

[8] ✓ 0.6s

Python

```
... ('{\n'
    '@odata.context': 'https://mydemo.search.windows.net/$metadata#Microsoft.Azure.Search.V2020_06_30.IndexerExecutionInfo',\n'
    'name': 'cogsrch-py-indexer-new',\n'
    'status': 'running',\n'
    'lastResult': {\n'
    'status': 'InProgress',\n'
    'errorMessage': null,\n'
    'startTime': '2022-09-19T02:31:21.752',\n'
    'endTime': null,\n'
    'itemsProcessed': 0,\n'
    'itemsFailed': 0,\n'
    'initialTrackingState': null,\n'
    'finalTrackingState': null,\n'
    'errors': [],\n'
    'warnings': [],\n'
    'metrics': null,\n'
    },\n'
    'executionHistory': [],\n'
    'limits': {\n'
    'maxRunTime': 'PT2H',\n'
    'maxDocumentExtractionSize': 134217728,\n'
    'maxDocumentContentCharactersToExtract': 4194304,\n'
    },\n'
    })
```

Now we get the service for the index definition.

```
# Query the service for the index definition
# Query responses can be verbose. If you get "Output exceeds the size limit. Open the full output data in a text editor", open the output in an editor.
r = requests.get(endpoint + "/indexes/" + index_name,
                  headers=headers, params=params)
pprint(json.dumps(r.json(), indent=1))

[9] ✓ 0.3s Python

... Output exceeds the size limit. Open the full output data in a text editor
{'\n'
  '@odata.context': '
    "https://mydemo.search.windows.net/$metadata#indexes/$entity",\n'
  '@odata.etag': "\"0x8DA99E6F8CB1757\"",\n'
  "name": "cogsrch-py-index-new",\n'
  "defaultScoringProfile": null,\n'
  "fields": [\n'
    {\n'
      "name": "id",\n'
      "type": "Edm.String",\n'
      "searchable": true,\n'
      "filterable": false,\n'
      "retrievable": true,\n'
      "sortable": true,\n'
      "facettable": false,\n'
      "key": true,\n'
      "indexAnalyzer": null,\n'
      "searchAnalyzer": null,\n'
      "analyzer": null,\n'
      "synonymMaps": []\n'
    },\n'
    {\n'
      "name": "content",\n'
      "type": "Edm.String",\n'
      "searchable": true,\n'
      "filterable": false,\n'
      "retrievable": true,\n'
      "sortable": false,\n'
      "facettable": false,\n'
      "key": false,\n'
      "indexAnalyzer": null,\n'
      "searchAnalyzer": null,\n'
      "analyzer": null,\n'
      "synonymMaps": []\n'
    },\n'
    {\n'
      "name": "languageCode",\n'
      "type": "Edm.String",\n'
      "searchable": true,\n'
      "filterable": false,\n'
      "retrievable": true,\n'
      "sortable": false,\n'
      "facettable": false,\n'
      "key": false,\n'
      "indexAnalyzer": null,\n'
      "searchAnalyzer": null,\n'
      "analyzer": null,\n'
      "synonymMaps": []\n'
    },\n'
  ],\n'
}
```

Below is the full output:

```
1 {'\n'
2   '@odata.context': '
3     "https://mydemo.search.windows.net/$metadata#indexes/$entity",\n'
4   '@odata.etag': "\"0x8DA99E6F8CB1757\"",\n'
5   "name": "cogsrch-py-index-new",\n'
6   "defaultScoringProfile": null,\n'
7   "fields": [\n'
8     {\n'
9       "name": "id",\n'
10      "type": "Edm.String",\n'
11      "searchable": true,\n'
12      "filterable": false,\n'
13      "retrievable": true,\n'
14      "sortable": true,\n'
15      "facettable": false,\n'
16      "key": true,\n'
17      "indexAnalyzer": null,\n'
18      "searchAnalyzer": null,\n'
19      "analyzer": null,\n'
20      "synonymMaps": []\n'
21    },\n'
22    {\n'
23      "name": "content",\n'
24      "type": "Edm.String",\n'
25      "searchable": true,\n'
26      "filterable": false,\n'
27      "retrievable": true,\n'
28      "sortable": false,\n'
29      "facettable": false,\n'
30      "key": false,\n'
31      "indexAnalyzer": null,\n'
32      "searchAnalyzer": null,\n'
33      "analyzer": null,\n'
34      "synonymMaps": []\n'
35    },\n'
36    {\n'
37      "name": "languageCode",\n'
38      "type": "Edm.String",\n'
39      "searchable": true,\n'
40      "filterable": false,\n'
41      "retrievable": true,\n'
42      "sortable": false,\n'
43      "facettable": false,\n'
44      "key": false,\n'
45      "indexAnalyzer": null,\n'
46      "searchAnalyzer": null,\n'
47      "analyzer": null,\n'
48      "synonymMaps": []\n'
49    },\n'
50  ],\n'
51 }
```

```

39 * "searchable": true,\n'
40 * "filterable": false,\n'
41 * "retrievable": true,\n'
42 * "sortable": true,\n'
43 * "facettable": false,\n'
44 * "key": false,\n'
45 * "indexAnalyzer": null,\n'
46 * "searchAnalyzer": null,\n'
47 * "analyzer": null,\n'
48 * "synonymMaps": []\n'
49 * },\n'
50 * {\n'
51 *   "name": "keyPhrases",\n'
52 *   "type": "Collection(Edm.String)",\n'
53 *   "searchable": true,\n'
54 *   "filterable": false,\n'
55 *   "retrievable": true,\n'
56 *   "sortable": false,\n'
57 *   "facettable": false,\n'
58 *   "key": false,\n'
59 *   "indexAnalyzer": null,\n'
60 *   "searchAnalyzer": null,\n'
61 *   "analyzer": null,\n'
62 *   "synonymMaps": []\n'
63 * },\n'
64 * {\n'
65 *   "name": "organizations",\n'
66 *   "type": "Collection(Edm.String)",\n'
67 *   "searchable": true,\n'
68 *   "filterable": false,\n'
69 *   "retrievable": true,\n'
70 *   "sortable": false,\n'
71 *   "facettable": false,\n'
72 *   "key": false,\n'
73 *   "indexAnalyzer": null,\n'
74 *   "searchAnalyzer": null,\n'
75 *   "analyzer": null,\n'
76 *   "synonymMaps": []\n'
77 * },\n'
78 * ],\n'
79 * "scoringProfiles": [],\n'
80 * "corsOptions": null,\n'
81 * "suggesters": [],\n'
82 * "analyzers": [],\n'
83 * "tokenizers": [],\n'
84 * "tokenFilters": [],\n'
85 * "charFilters": [],\n'
86 * "encryptionKey": null,\n'
87 * "similarity": {\n'
88 *   "@odata.type": "#Microsoft.Azure.Search.BM25Similarity",\n'
89 *   "k1": null,\n'
90 *   "b": null,\n'
91 * },\n'
92 * '})
93

```

Now we query the index to return contents of organizations

```

▶ # Query the index to return the contents of "organizations", created through Entity Recognition during enrichment
# For keyword search, replace the asterisk with comma-separated query terms: search-microsoft,azure
r = requests.get(endpoint + "/indexes/" + index_name +
                "/docs?search=*$select=organizations", headers=headers, params=params)
pprint(json.dumps(r.json(), indent=1))

[10] ✓ 0.5s Python

... {'@odata.context': '
  "https://mydemo.search.windows.net/indexes(\'cogsrch-py-index-new\')/$metadata#docs(*)",
  "value": []}

```

Output

Data sources created:

Microsoft Azure Search resources, services, and docs (G+J)

Home > mydemo Search service

Search

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Semantic search (Preview)

Knowledge Center

Keys

Scale

Search traffic analytics

Identity

Networking

Properties

Locks

+ Add index Import data Search explorer Refresh Delete Move

Essentials

Resource group (move) : sdai3

Location (move) : West US 2

Subscription (move) : Azure for Students

Subscription ID : 5cf92398-d8a0-400a-9d32-05fcb62b046b

Status : Running

Url : https://mydemo.search.windows.net

Pricing tier : Standard

Replicas : 1 (99.9% read/write SLA)

Partitions : 1

Search units : 1

Tags (edit) : Click here to add tags

Get Started Usage Monitoring Indexes Indexers Data sources Aliases Skillsets Debug sessions

+ Add data source Refresh Delete

Name	Type	Table/Collection
cogsrch-py-datasource-new	Azure Blob Storage	new-blob-service
cogsrch-py-datasource	Azure Blob Storage	cog-search-demo

Skillsets created:

Microsoft Azure Search resources, services, and docs (G+J)

Home > mydemo Search service

Search

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Semantic search (Preview)

Knowledge Center

Keys

Scale

Search traffic analytics

Identity

Networking

Properties

Locks

+ Add index Import data Search explorer Refresh Delete Move

Essentials

Resource group (move) : sdai3

Location (move) : West US 2

Subscription (move) : Azure for Students

Subscription ID : 5cf92398-d8a0-400a-9d32-05fcb62b046b

Status : Running

Url : https://mydemo.search.windows.net

Pricing tier : Standard

Replicas : 1 (99.9% read/write SLA)

Partitions : 1

Search units : 1

Tags (edit) : Click here to add tags

Get Started Usage Monitoring Indexes Indexers Data sources Aliases Skillsets Debug sessions

+ Add skillset Refresh Delete

Name	Number of Skills
cogsrch-py-skillset-new	4
cogsrch-py-skillset	4

Indexes created:

Microsoft Azure Search resources, services, and docs (G+)

Home > mydemo Search service

Search

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Semantic search (Preview)

Knowledge Center

Keys

Scale

Search traffic analytics

Identity

Networking

Properties

Locks

+ Add index Import data Search explorer Refresh Delete Move

Essentials

Resource group (move) : sdai3

Location (move) : West US 2

Subscription (move) : Azure for Students

Subscription ID : 5cf92398-d8a0-400a-9d32-05fcb62b046b

Status : Running

Url : https://mydemo.search.windows.net

Pricing tier : Standard

Replicas : 1 (99.9% read/write SLA)

Partitions : 1

Search units : 1

Tags (edit) : Click here to add tags

Get Started Usage Monitoring Indexes Indexers Data sources Aliases Skillsets Debug sessions

+ Add index Refresh Delete

Name	Document Count	Storage Size
cogsrch-py-index	14	2.22 MB
cogsrch-py-index-new	10	142.58 KB

Indexers created:

Microsoft Azure Search resources, services, and docs (G+)

Home > mydemo Search service

Search

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Semantic search (Preview)

Knowledge Center

Keys

Scale

Search traffic analytics

Identity

Networking

Properties

Locks

Monitoring

+ Add index Import data Search explorer Refresh Delete Move

Essentials

Resource group (move) : sdai3

Location (move) : West US 2

Subscription (move) : Azure for Students

Subscription ID : 5cf92398-d8a0-400a-9d32-05fcb62b046b

Status : Running

Url : https://mydemo.search.windows.net

Pricing tier : Standard

Replicas : 1 (99.9% read/write SLA)

Partitions : 1

Search units : 1

Tags (edit) : Click here to add tags

Get Started Usage Monitoring Indexes Indexers Data sources Aliases Skillsets Debug sessions

+ Add indexer Refresh Delete

Status	Name	Last run	Docs succeeded	Errors/Warnings
Success	cogsrch-py-indexer-new	15 minutes ago	10/10	0/12
Success	cogsrch-py-indexer	2 days ago	14/14	0/27