Quick tip: Review the prerequisites before you run the lab

End Lab 01:29:59

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Open Google Cloud Console

Username

student-00-4b7d56d05044

Password

KtcZKS8dRgkj

GCP Project ID

qwiklabs-gcp-02-1e1ac1a

Create a Custom Search App using the Discovery API: Challenge Lab

Lab 1 hour 30 minutes No cost Intermediate

★★★★½

This lab may incorporate AI tools to support your learning.

Previous Next

Challenge Lab Overview

This lab will challenge you to perform actions and automation across products. Instead

system (shown on this page) provides feedback on whether you have completed your tasks correctly.

When you take a Challenge Lab, you will not be taught Google Cloud concepts. You will need to use your skills to assess how to build the solution to the challenge presented. This lab is only recommended for students who have those skills. Are you up for the challenge?

Objective

This lab challenges you to build a custom search app for programming info using AI Applications and Discovery API. You'll create a datastore with Python website URLs, build a search interface to query the data and display results.

- AI Applications
- Search Apps
- Agents
- Gemini

Setup and requirements

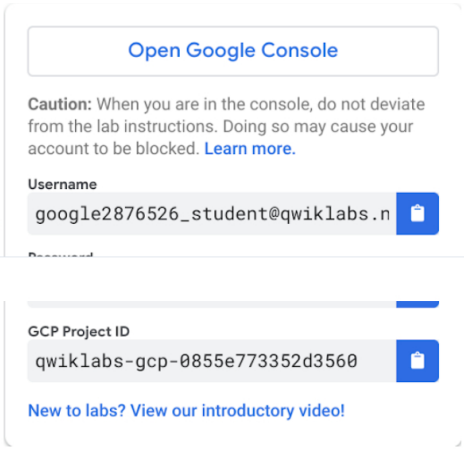
For each lab, you get a new Google Cloud project and set of resources for a fixed time at no cost.

1. Make sure you signed into Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, 02:00:00 and make sure you can finish in that time block.

There is no pause feature. You can restart if needed, but you have to start at the beginning.

3. When ready, click START LAB.
4. Note your lab credentials. You will use them to sign in to the Google Cloud

Console.



5. Click **Open Google Console**.
6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

If you use other credentials, you'll get errors or **incur charges**.

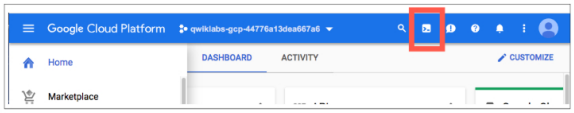
7. Accept the terms and skip the recovery resource page.

Do not click **End Lab** unless you are finished with the lab or want to restart it. This clears your work and removes the project.

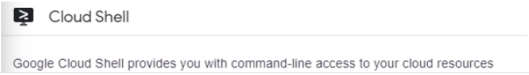
Activate Cloud Shell

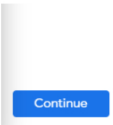
Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources.

In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.

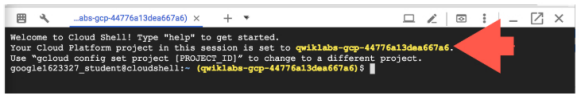


Click **Continue**.





It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your *PROJECT_ID*. For example:



`gcloud` is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

You can list the active account name with this command:

(Output)

```
Credentialed accounts:
 - <myaccount>@<mydomain>.com (active)
```

(Example output)

```
Credentialed accounts:
 - google1623327_student@qwiklabs.net
```

You can list the project ID with this command:

```
gcloud config list project
```

(Output)

(Example output)

```
[core]
project = qwiklabs-gcp-44776a13dea667a6
```

For full documentation of `gcloud` see the gcloud command-line tool overview.

Challenge Scenario

Here is a company overview as provided on the Cymbal Shops's website.

> Cymbal Shops is an American retail chain headquartered in Minneapolis that sells homeware, electronics, and clothing.
>
> Founded in 1974, Cymbal Shops started out as Cymbal Air, selling AC systems manufactured by Cymbal Group in Minnesota and neighboring states. The company quickly expanded into domestic merchandise in order to satisfy the need for quality products at an affordable price in the midst of a recession.
>
> Cymbal Shops' broad product assortment was once a benefit - capturing planned purchases and last minute splurges. In recent years, however, the company has struggled to adapt to the acceleration in e-commerce. Digitally native companies are on the rise and Cymbal Shops must implement significant changes in order to keep pace and maintain relevance. Today, Cymbal Shops operates 714 stores across North America and reported $15 billion in revenue in 2019. They currently

> Cymbal Shops is a digitally transforming legacy retailer.
>
> It is inspired by clients like: Bed Bath & Beyond, Best Buy, Home Depot, Nordstrom.

Your challenge

Cymbal Shops is looking for proficiency in creating a Custom Search App using the Discovery API. You work with Cymbal Shops, throughout the lab, you are tasked with deploying search app which involves two major tasks:

1. **Implementing Search Functionality:** Write a python code to connect with Discovery Engine, send user queries, and process results.
2. **Format search results:** Transform raw API data (titles, snippets, URLs) into a user-friendly display for the web application.

Task 1. Create an AI Applications Data store and Search app

Data store and search app

In this section, you will utilize AI Applications to create a data store. Data stores act as knowledge bases, storing information from websites and documents that your agent can use to answer user questions. Therefore in this task,

1. Enable the AI Applications API.
2. Create a data store that will be used by a generative agent to answer user questions. Configure the data store using the following details:

Property	Value
Data Store Name	`cymbal-data-store`
Data Source	Website Content
Website URLs	docs.python.org , www.w3schools.com/python

Click **Check my progress** to verify the objectives.

Create an AI Applications Data store

Check my progress

3. Finally, create an application of type Search with the following configurations. This application will use the data store named `cymbal-data-store`.

Property	Value
App Type	Search
Content	Generic
Enterprise edition features	on
Advanced LLM features	on
Name the app	`cymbal-programming-search-app`
Company Name	`cymbal-python-programming-company`
Data Store	`cymbal-data-store`

4. Preview your app and ask it a question similar to: `What is the syntax of a Loop in Python?`

Note: It may take a short time for the Data store to be available. If you get an error, wait a couple of minutes and try again.

Click **Check my progress** to verify the objectives.

Create a Search App

Check my progress

Task 2. Deploy a Python website

In this section, you will deploy a Python website. You can execute your Flask application directly within Cloud Shell, situated in the app directory, and preview it by clicking the **Web Preview** button located at the top of the Cloud Shell editor.

1. In **Google Cloud Shell**, run the following command to copy files from the `vertex-search-assessment` folder.

```
gsutil cp -r gs://partner-genai-bucket/genai059/vertex-search-assessment/ .
```

2. You can further explore these files.
3. Setup the virtual environment, using the following commands.

```
pip install virtualenv
virtualenv ~/genai-assessment
source ~/genai-assessment/bin/activate
```

4. Populate the `requirements.txt` file with the Vertex AI module, and then install the packages listed in the `requirements.txt` file using `pip`.

the packages listed in the `requirements.txt` file using `pip`:

```
echo "vertexai" >> requirements.txt
```

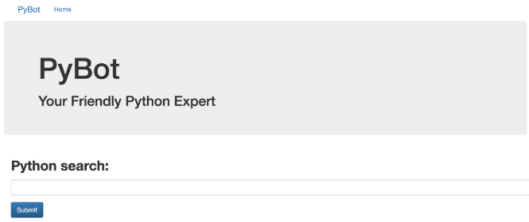
```
pip install -r requirements.txt
```

5. In the `main.py` file, the code snippet outlines a Flask application that interacts with the Discovery Engine. Run the file using:

```
python main.py
```

Preview it on port 8080 by clicking on the **Web Preview** option located at the top-right of the Cloud Shell window.

If the deployment is successful, your application should look like this:



Build a search interface for Google Discovery Engine Datastore

In the previous sub-section, you deployed a Python Flask application. Now, in this sub-section, you'll further develop the application by implementing the existing `search_data_store` and `format_response` functions within the `main.py` file. These functions are already defined but not yet implemented. Once implemented, they will enable users to search your Datastore and display formatted results.

`format_response` functions in the `main.py` file. There are provided comments within the code to guide you through the implementation.

2. The `search_data_store` function interacts with the Discovery Engine API to search your Datastore based on a given user query. It should return a list of `discoveryengine.SearchResponse` objects containing the search results.

3. The `format_response` function takes the raw search results obtained from `search_data_store` and transforms them into a suitable format for displaying on a web page. It accepts a list of `discoveryengine.SearchResponse` objects as input.

   The function should return a list of dictionaries. Each dictionary represents a formatted search result with the following keys:

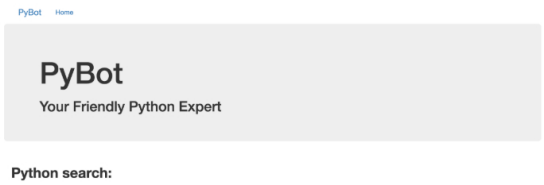
   - `title`: The title of the search result document.
   - `snippet`: A relevant snippet from the document.
   - `url`: The formatted URL of the search result document.

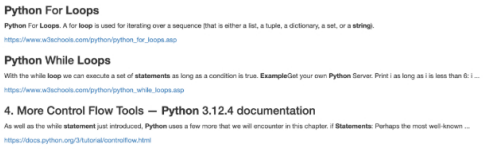
```
python main.py
```

Then, preview it on port 8080 by clicking on the **Web Preview** option located at the top-right of the Cloud Shell window.

Ask the **PyBot** application a sample question, such as: `What is the syntax of a Loop in Python?`

Output:





Congratulations!

At the end of this lab, you have gained practical experience in using AI Applications and the Discovery API to create a custom search solution.

Manual Last Updated June 13, 2024

Lab Last Tested June 13, 2024

Course Badge