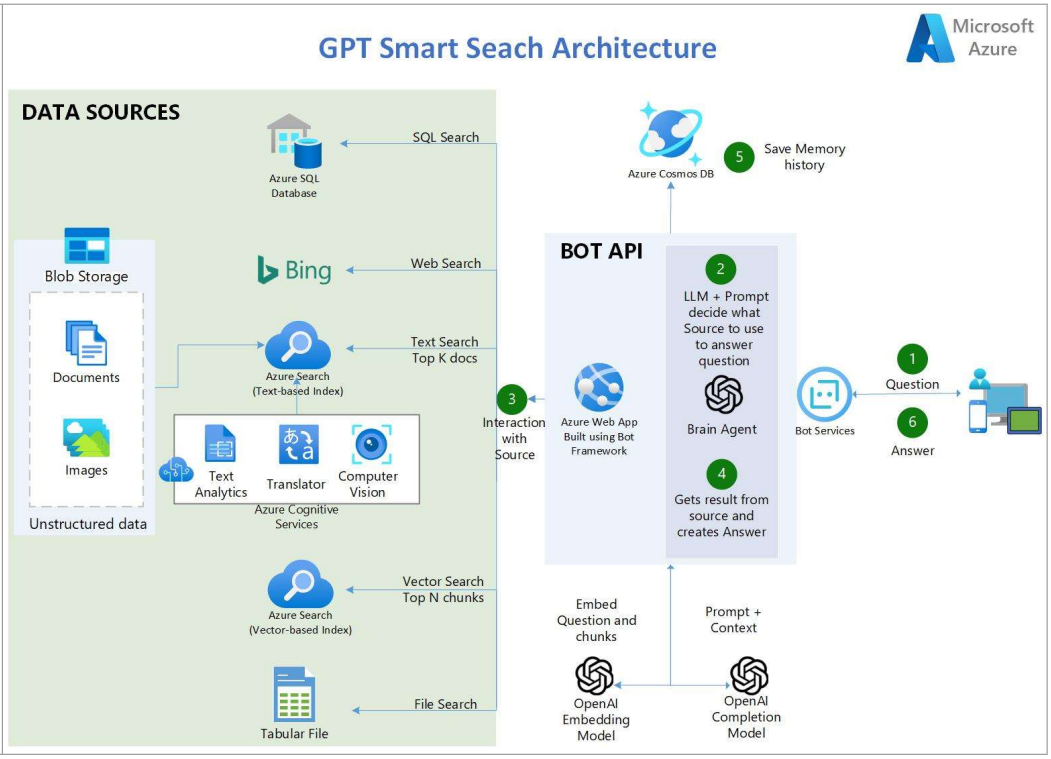
Azure OpenAI | VBD Delivery Training - Setup

sabato 12 agosto 2023 10:18

Repo to clone: https://github.com/maurominella/Azure-Cognitive-Search-Azure-OpenAI-Accelerator

Hi level architecture

1. The user asks a question.
2. In the app, an OpenAI GPT-4 LLM uses a clever prompt to determine which source to use based on the user input
3. Four types of sources are available:
   - 3a. Azure SQL Database - contains COVID-related statistics in the US.
   - 3b. Azure Bing Search API - provides access to the internet allowing scenerios like: QnA on public websites .
   - 3c. Azure Cognitive Search - contains AI-enriched documents from Blob Storage (10k PDFs and 90k articles).
     - 3c.1. Uses OpenAI to vectorize the top K document chunks
     - 3c.2. Fills up the vector-based indexes on-demand.
     - 3c.3. Gets the Top N Chunks by doing a vector search on vector-based indexes.
   - 3d. CSV Tabular File - contains COVID-related statistics in the US.
4. The app retrieves the result from the source and crafts the answer.
5. The tuple (Question and Answer) is saved to CosmosDB to keep a record of the interaction and further analysis.
6. The answer is delivered to the user.



Features

- Uses Bot Framework and Bot Service to Host the Bot API Backend and to expose it to multiple channels including MS Teams.
- 100% Python.
- Uses Azure Cognitive Services to index and enrich unstructured documents: Detect Language, OCR images, Key-phrases extraction, entity recognition (persons, emails, addresses, organizations, urls).
- Uses Vector Search Capabilities of Azure Cognitive Search to provide the best semantic answer.
- Creates vectors on-demand as users interact with the system. (versus vectorizing the whole datalake at the beginning)
- Uses LangChain as a wrapper for interacting with Azure OpenAI , vector stores, constructing prompts and creating agents.
- Multi-Lingual (ingests, indexes and understand any language)
- Multi-Index -> multiple search indexes
- Tabular Data Q&A with CSV files and SQL flavor Databases
- Uses Azure AI Document Intelligence SDK (former Form Recognizer) to parse complex/large PDF documents
- Uses Bing Search API to power internet searches and Q&A over public websites.
- Uses CosmosDB as persistent memory to save user's conversations.
- Uses Streamlit to build the Frontend web application in python.

Repo Setup

Find right directory

```
conda deactivate
cd ~/cloudfiles/code/Users/mauro.minella/git_repos/
PS1='\u:\W\$ '
clear
```

Open the parent folder on you machine and clone your repo

```
git clone https://github.com/maurominella/Azure-Cognitive-Search-Azure-OpenAI-Accelerator.git
```

```
20.103.123.96 - PuTTY
azureuser:git_repos$ git clone https://github.com/maurominella/Azure-Cognitive-Search-Azure-OpenAI-Accelerator.git
Cloning into 'Azure-Cognitive-Search-Azure-OpenAI-Accelerator'...
remote: Enumerating objects: 3700, done.
remote: Counting objects: 100% (703/703), done.
remote: Compressing objects: 100% (193/193), done.
remote: Total 3700 (delta 585), reused 570 (delta 508), pack-reused 2997
Receiving objects: 100% (3700/3700), 169.24 MiB | 17.01 MiB/s, done.
Resolving deltas: 100% (2377/2377), done.
Updating files: 100% (62/62), done.
azureuser:git_repos$
```

Rename your local folder to an easier one, then CD into that folder (we see both Windows and Linux commands here on the right):

```
Linux:
azureuser:git_repos$ mv Azure-Cognitive-Search-Azure-OpenAI-Accelerator openaivbd20231215

Windows:
azureuser:git_repos$ ren Azure-Cognitive-Search-Azure-OpenAI-Accelerator openaivbd20231215

Both Linux and Windows:
azureuser:git_repos$ cd openaivbd20231215
```

If you need to clone a specific branch: qui aggiungiamo il branch 20231215 e poi cancelliamo il branch main originale

```
azureuser:openaivbd20231124$ git branch -a
* main
  remotes/origin/20230903
```

```
  remotes/origin/20231124
  remotes/origin/HEAD -> origin/main
  remotes/origin/main

azureuser:openaivbd20231215$ git checkout -b 20231215
Switched to a new branch '20231215'

azureuser:openaivbd20231215$ git branch -a
* 20231124
  main
  remotes/origin/20230903
  remotes/origin/20231124
  remotes/origin/HEAD -> origin/main
  remotes/origin/main

azureuser:openaivbd20231215$ git branch -d main
Deleted branch main (was 077bf8e).

azureuser:openaivbd20231215$ git branch -a
* 20231124
  remotes/origin/20230903
  remotes/origin/20231124
  remotes/origin/HEAD -> origin/main
  remotes/origin/main

azureuser:openaivbd20231215$ git branch --set-upstream-to=origin/20231215 20231215
azureuser:openaivbd20231215$ git pull
```

If you need to commit and push changes to a branch (part 1)

```
Make sure you're on the right branch:
azureuser:openaivbd20231215$ git branch -a
  main
* openaivbd_azuresearchapi2023-11-01
  remotes/origin/20230903
  remotes/origin/20231124
  remotes/origin/20231215
  remotes/origin/HEAD -> origin/main
  remotes/origin/langchain_experimental
  remotes/origin/main

Set up user (of course you need the private key stored in this machine):
azureuser:openaivbd20231215$ git config --local user.email "mauro.minella@microsoft.com"
azureuser:openaivbd20231215$ git config --local user.name "Mauro Minella"


azureuser:openaivbd20231215$
```

If you need to commit and push changes to a branch (part 2)

When you add a remote to your Git configuration, you're specifying the URL of the remote repository, not a specific branch. Each remote repository can contain multiple branches, but the remote configuration just points to the repository itself.

So, make sure that the remote path is correctly configured. In this case, I remove the default "origin" remote and I add "my_origin", pointing to my repo (no branch is needed here) and making sure that I'm using the SSH path (**not the HTTP PATH!**) provided by github

```
azureuser:openaivbd20231215$ git remote -v
origin  https://github.com/maurominella/Azure-Cognitive-Search-Azure-OpenAI-Accelerator.git (fetch)
origin  https://github.com/maurominella/Azure-Cognitive-Search-Azure-OpenAI-Accelerator.git (push)

azureuser:openaivbd20231215$ git remote remove origin

azureuser:openaivbd20231215$ git remote -v
(empty)

Provide the SSH path, not the HTTPS path provided by github!!:
azureuser:openaivbd20231215$ git remote add my_origin git@github.com:maurominella/Azure-Cognitive-Search-Azure-
OpenAI-Accelerator.git

azureuser:openaivbd20231215$ git remote -v
my_origin       git@github.com:maurominella/Azure-Cognitive-Search-Azure-OpenAI-Accelerator.git (fetch)
my_origin       git@github.com:maurominella/Azure-Cognitive-Search-Azure-OpenAI-Accelerator.git (push)
```

If you need to commit and push changes to a branch (part 3)

First of all, check the remote branches.
In fact, when you push changes, you specify both the remote name and the branch name to ensure you're pushing to the correct branch. Here's how you can ensure you're not pushing to the remote's default branch (often the main or master branch) unless you intend to

```
azureuser:openaivbd20231215$ git branch -vv
  main     d84b779 Merge pull request #75 from MSUSAzureAccelerators/main
* 20231215 d84b779 Merge pull request #75 from MSUSAzureAccelerators/main

azureuser:openaivbd20231215$ git push -u my_origin 20231215
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 24.14 KiB | 588.00 KiB/s, done.
Total 9 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), completed with 5 local objects.
remote:
remote: Create a pull request for '20231215' on GitHub by visiting:
remote:      https://github.com/maurominella/Azure-Cognitive-Search-Azure-OpenAI-Accelerator/pull/new/20231215
remote:
To github.com:maurominella/Azure-Cognitive-Search-Azure-OpenAI-Accelerator.git
 * [new branch]      openaivbd_azuresearchapi2023-11-01 -> 20231215
branch '20231215' set up to track 'my_origin/20231215'.
```

Check (git remote -v) remote locations and remove the ones (like origin) you don't need.
Anyway, remove (git remote remove) ALL of them since with Linux we need SSH rather than HTTPS paths

```
azureuser:openaivbd20231124$ git remote -v
openaivbd       https://github.com/maurominella/Azure-Cognitive-Search-Azure-OpenAI-Accelerator.git (fetch)
openaivbd       https://github.com/maurominella/Azure-Cognitive-Search-Azure-OpenAI-Accelerator.git (push)
origin  https://github.com/maurominella/Azure-Cognitive-Search-Azure-OpenAI-Accelerator.git (fetch)
origin  https://github.com/maurominella/Azure-Cognitive-Search-Azure-OpenAI-Accelerator.git (push)

azureuser:openaivbd20231124$ git remote -v
openaivbd       https://github.com/maurominella/Azure-Cognitive-Search-Azure-OpenAI-Accelerator.git (fetch)
openaivbd       https://github.com/maurominella/Azure-Cognitive-Search-Azure-OpenAI-Accelerator.git (push)
origin  https://github.com/maurominella/Azure-Cognitive-Search-Azure-OpenAI-Accelerator.git (fetch)
origin  https://github.com/maurominella/Azure-Cognitive-Search-Azure-OpenAI-Accelerator.git (push)
azureuser:openaivbd20231124$ git remote remove openaivbd
azureuser:openaivbd20231124$ git remote remove origin
azureuser:openaivbd20231124$ git remote -v
```

Create Conda Environment

In brief:

- notepad openaivbd_env.yml. The file does not exist so accept the option to created it, then paste the following text:

```
name: openaivbd_20231215
dependencies:
  - python=3.10
```

```
  - pip
```

- `conda env remove -n openaivbd_20231215`
- `conda env create -f openaivbd_env.yml`
- `conda activate openaivbd_20231215`
- On windows:
  - `pip install -r .\common\requirements.txt` (if you get an error on windows, please install C++ runtime as explained here)
- On Linux:
  - `pip install -r ./common/requirements.txt`
- `pip install jupyter` (only on windows) OR `pip install ipykernel` (only on Linux)
- `jupyter kernelspec uninstall openaivbd_20231215`
- `python -m ipykernel install --name openaivbd_20231215 --user`
- `jupyter kernelspec list`
- `jupyter notebook` (only on windows)
- `pip install semantic-kernel` (only for testing purposes on Semantic Kernel)

Prerequisites

Prerequisites Client 3-5 Days POC

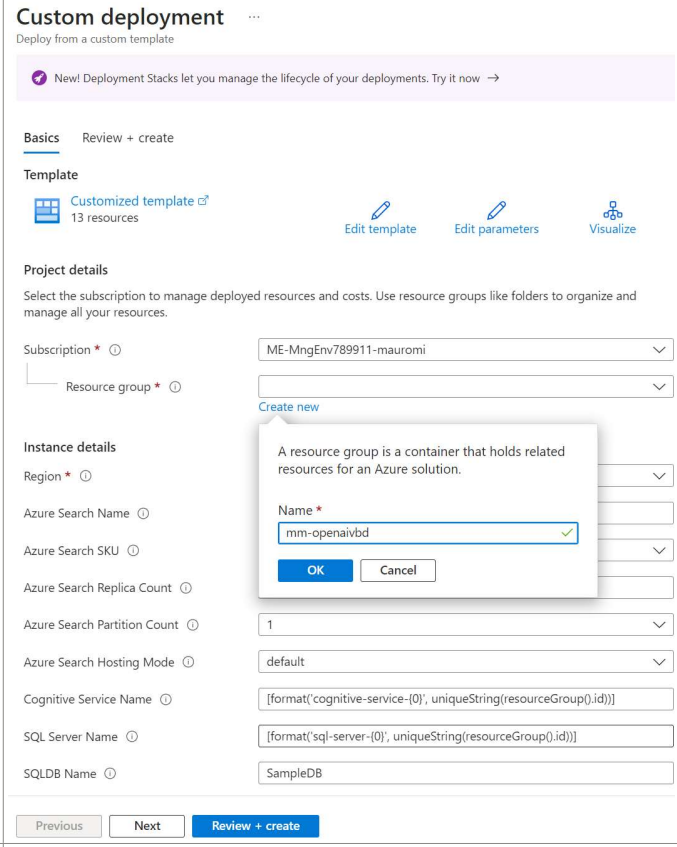
- For Windows machines: Microsoft Visual C++ 14.0 or greater as explained above
- Azure subscription
- Accepted Application to Azure Open AI, including GPT-4 (mandatory)
- Microsoft members need to be added as Guests in clients Azure AD
- A Resource Group (RG) needs to be set for this Workshop POC, in the customer Azure tenant
- The customer team and the Microsoft team must have Contributor permissions to this resource group so they can set everything up 2 weeks prior to the workshop
- A storage account must be set in place in the RG.
- Data/Documents must be uploaded to the blob storage account, at least two weeks prior to the workshop date
- For IDE collaboration during workshop, Jupyper Lab will be used, for this, Azure Machine Learning Workspace must be deployed in the RG
  - Note: Please ensure you have enough core compute quota in your Azure Machine Learning workspace
- Deploy these models in you Azure OpenAI Studio (it won't be automatically deployed by the arm template). **Make sure that the deployment name is the same as the model name.**
  - "gpt-35-turbo"
  - "gpt-35-turbo-16k"
  - "gpt-4"
  - "gpt-4-32k"
  - "text-embedding-ada-002"
- Deploy to Azure as explained in the home page, or click on https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2Fpablomarin%2FGPT-Azure-Search-Engine%2Fmain%2Fazuredeploy.json

Deployment!

Here it is, on my subscription.

Manual information to fill:

- Regione (e.g. Weste Europe)
- Resource group (e.g. mm01-openaivbdgrp)
- SQL administrator login (e.g. mauromi)
- SQL Administrator Login Password (P@ssw0rd!1)



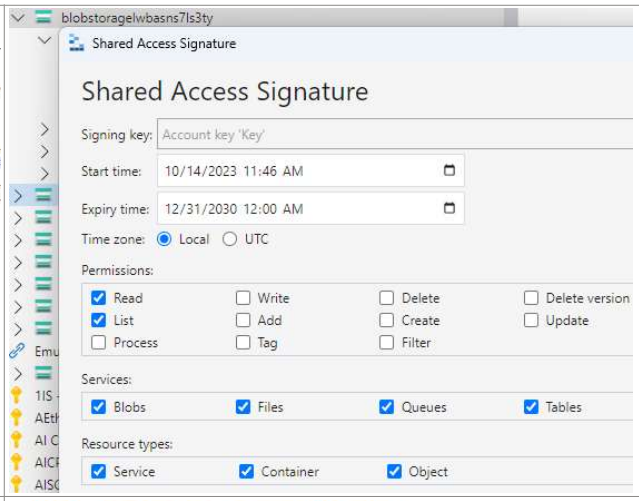
Right after the installation, the RG includes the following 8 services. In this case, we don't have nor

- Open AI
- , neither Azure ML

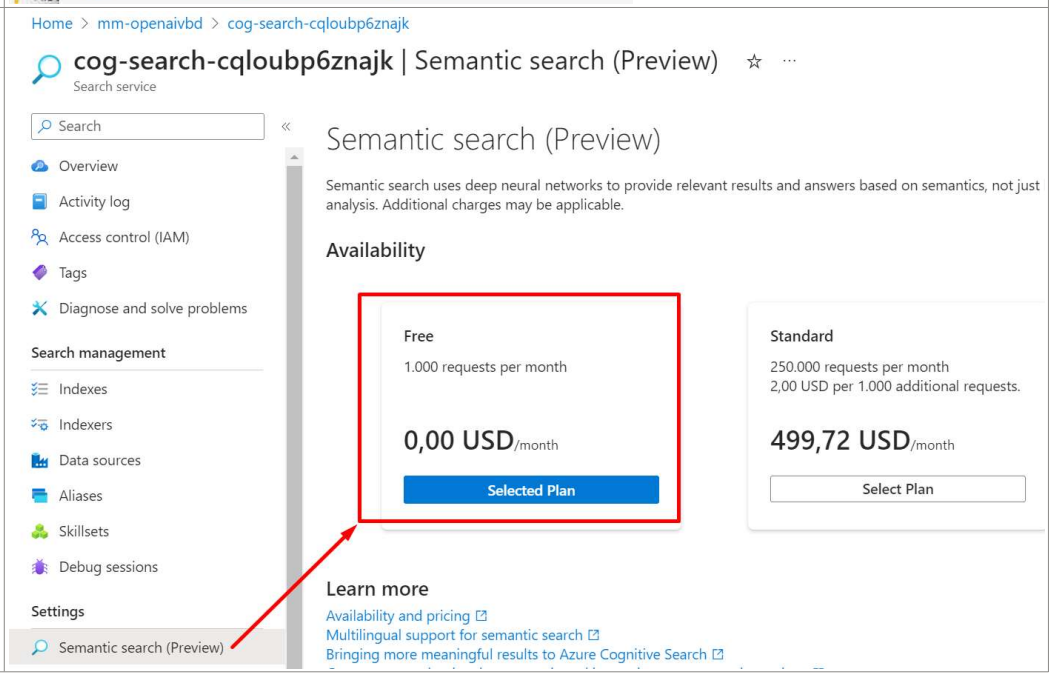
Name	Type	Location
bing-search-yhyufoqh6yjr6	Bing Resource	Global
blobstorageyhyufoqh6yjr6	Storage account	West Europe
cog-search-yhyufoqh6yjr6	Search service	West Europe
cognitive-service-yhyufoqh6yjr6	Azure AI services multi-service account	West Europe
cosmosdb-account-yhyufoqh6yjr6	Azure Cosmos DB account	West Europe
form-recognizer-yhyufoqh6yjr6	Document intelligence	West Europe
SampleDB (sql-server-yhyufoqh6yjr6/SampleDB)	SQL database	West Europe
sql-server-yhyufoqh6yjr6	SQL server	West Europe

Create the SAS Token that we'll fill into credentials_my.env
?sv=2021-10-04&ss=btqf&srt=sco&st=2023-10-14*****

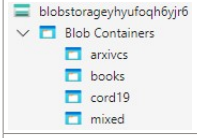
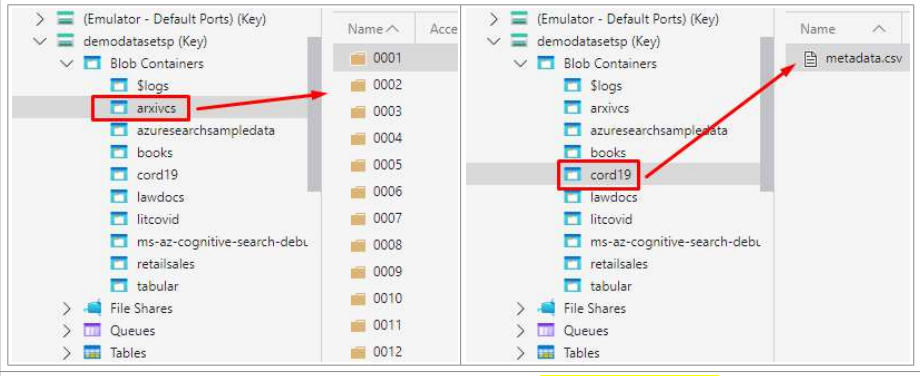
```
# Don't mess with this unless you really know what you are doing
AZURE_SEARCH_API_VERSION = "2023-07-01-Preview"
AZURE_OPENAI_API_VERSION = "2023-05-15"
BING_SEARCH_URL          = "https://api.bing.microsoft.com/v7.0/search"

# Demo Data (edit with your own if you want to use your own data)

# BLOB STORAGE
BLOB_CONNECTION_STRING        = "DefaultEndpointsProtocol=https;Account
BLOB_SAS_TOKEN                = "?sv=2021-10-04&ss=btqf&srt=sco&st=202

BLOB_CONNECTION_STRING_PUBLIC = "DefaultEndpointsProtocol=https;Account
BLOB_SAS_TOKEN_PUBLIC         = "?sv=2022-11-02&ss=bf&srt=sco&sp=rltfx
```



Post deployment: enable semantic search (if it isn't)



Source data

Nel repo originale, i dati da indicizzare si trovano in un blob storage di cui vengono fornite le credenziali nel file credentials.env	• BLOB_CONNECTION_STRING="DefaultEndpointsProtocol=https;AccountName=demodatasetsp;AccountKey=QVFgIKPiWB+8f0mH+F7fidVLG7wq1S3WhtAqXOWaMWtr6fZ4frhVgmUzgBSdkmw4VsjoEAo7C2Hn+ASt2Cc5HA==;EndpointSuffix=core.windows.net" • BLOB_SAS_TOKEN="?sv=2022-11-02&ss=bf&srt=sco&sp=rltfx&se=2024-10-02T01:02:07Z&st=2023-08-03T17:02:07Z&spr=https&sig=gLxStXFSY6X29OPpPDpBEhoQDdtJNDrMVExNYJ%2BhmBQ%3D"
In particolare: • i 9.8K file PDF si trovano nella root del container arxivcs, • mentre il singolo file metadata.csv da 116Mb si trova nella root del container cord19 Per evitare costi eccessivi (specie per il servizio cognitivo che andrebbe ad estrarre tutte le entità), io ho inserito nel MIO storage del repo, all'interno della cartella data\storage account containers, tre sottocartelle con i nomi dei container con un numero minimale dei file necessari (mixed è vuoto) 	
In questo modo posso modificare solamente le due righe con le mie credenziali. **Nota**: il SAS Token l'ho generato dallo Storage Explorer direttamente sul container, e poi ho aggiunto il punto di domanda all'inizio	• BLOB_CONNECTION_STRING="DefaultEndpointsProtocol=https;AccountName=blobstorage5uczmzjbcxd2s;AccountKey=0jk9AA+RrsbTkxgNWXPFF2spMf2sS/t7****CgHfB+AStjDTF6w==;EndpointSuffix=core.windows.net" • BLOB_SAS_TOKEN="?sv=2021-10-04&st=2023-08-29T17%****uO3RA%3D"