Lab Guide: Getting Started with Generative AI in watsonx.ai

Author: Elena Lowery, elowery@us.ibm.com

Contents

Overview 3

Required software, access, and files 3

Generative AI and Large Language Models 4

Part 1: Understand LLM capabilities 6

Part 2: Integrate LLMs with applications 23

Conclusion 31

Overview

In this lab you will learn how to implement generative AI use cases in **watsonx.ai**. *Watsonx.ai* is an AI platform which can be used to implement both traditional machine learning use cases and use cases that utilize *Large Language Models* (LLMs).

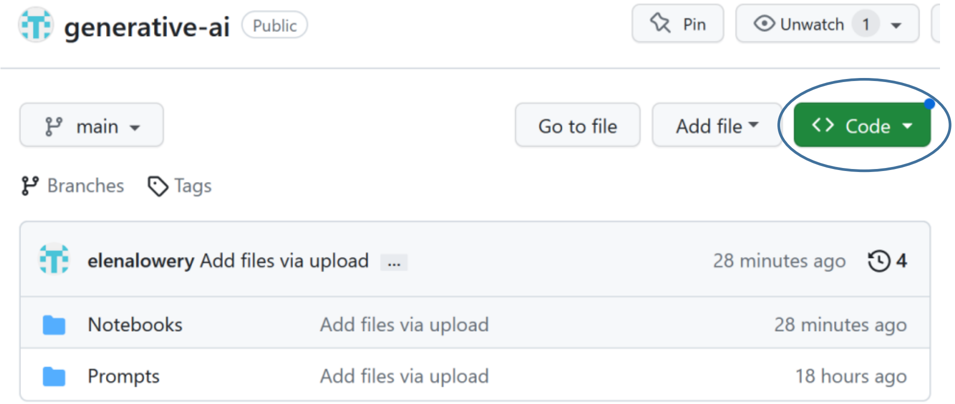
We will take a closer look at the following use cases:

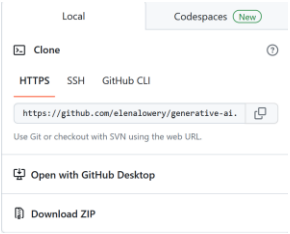
- Generation
- Summarization
- Classification

Note: LLMs are a type of a foundation model. In IBM tools and documentation, the terms LLM and foundation models are used interchangeably.

Required software, access, and files

- To complete this lab, you will need access to *watsonx.ai*. You can get access by signing up for an IBM Cloud account and provisioning watsonx.ai service.
- A Python IDE with Python 3.10 or 3.11 environment (*Visual Studio Code* or *PyCharm*)
- You will also need to download and unzip this GitHub repository: *https://github.com/elenalowery/generative-ai*
- Click the **Code** button and select **Download ZIP**





```
https://github.com/elenalowery/generative-ai.
```

Use Git or checkout with SVN using the web URL.

- Unzip the downloaded zip file. In the lab we will refer to this folder as the *git repo* folder.

Generative AI and Large Language Models

Generative AI is a new domain in AI which allows users to interact with models using natural language. A user sends requests (*prompts*) to a model, and the model generates a response. To an end user generative AI may look like a chatbot or a search engine, but implementation of generative AI is different from legacy chatbots that rely on hardcoded business rules and search engines that use indexing.

Generative AI uses Large Language Models (LLMs) to generate response to a prompt.

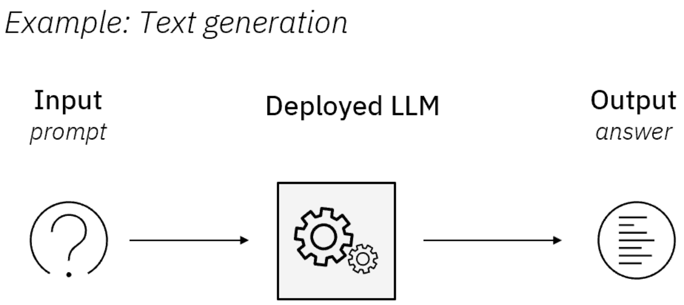
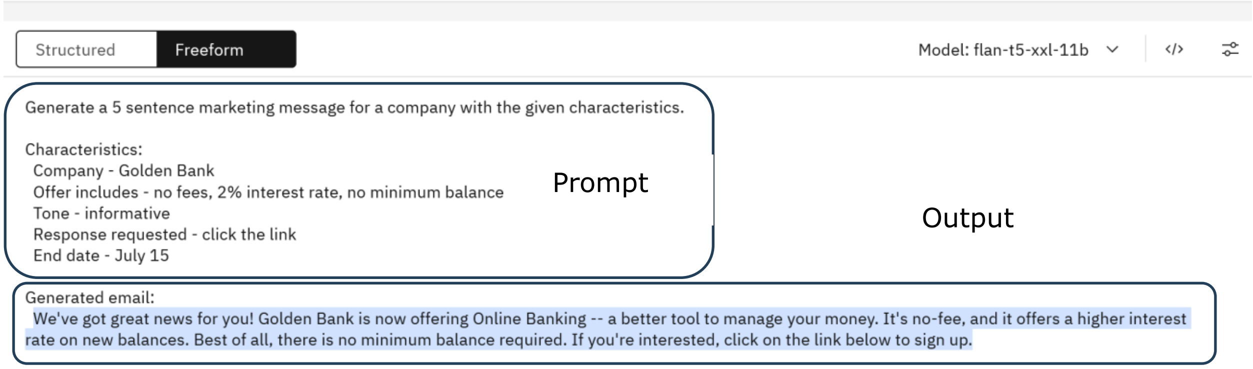


Figure 1: Prompt Lab in watsonx.ai



Generate a 5 sentence marketing message for a company with the given characteristics.

Characteristics:
Company - Golden Bank
Offer includes - no fees, 2% interest rate, no minimum balance
Tone - informative
Response requested - click the link
End date - July 15

Generated email:
We've got great news for you! Golden Bank is now offering Online Banking -- a better tool to manage your money. It's no-fee, and it offers a higher interest rate on new balances. Best of all, there is no minimum balance required. If you're interested, click on the link below to sign up.

Unlike traditional machine learning models, which always require training, LLMs are pretrained on a very large dataset. There are dozens of LLMs, which are developed by different companies. Some companies contribute their models to open source, and many of them are available on the LLM community site *Hugging Face*. It's up to the LLM developer to publish information about the model and the dataset that the model has been trained on. For example, see the model card for one of the popular open source models, *Flan-T5-xxl*. In general, all LLMs are trained on publicly available data. IBM is one of the few companies that publishes detailed information about the data that was used to train the model. This information can be found in Research Papers published by IBM.

You can think of the data that the model has been trained on as its "knowledge". For example, if the model was not trained on a dataset that contained *2022 Soccer World Cup* results, it will not be able to generate valid/correct answers related to this event. Of course, this applies to all business use cases in which we need models to interact with proprietary enterprise data. We will explain how to solve this problem later in the lab.

Two more factors influence LLM capabilities: size and instruction-tuning. Larger models have been trained on more data and have more parameters. In the context of LLMs, the number of parameters refers to the number of adjustable weights in the neural network that the model uses to generate responses. Parameters are the internal variables of the model that are learned during the training process and represent the knowledge the model has acquired.

While it may seem obvious that a larger model will produce better results, in a production implementation we may need to consider smaller models that meet our use case requirements because of the hosting and inferencing cost.

Instruction-tuned models are models that have been specifically trained for tasks such as classification or summarization. See IBM documentation for other considerations when choosing a model.

As we work through the lab, we will introduce a few more important LLM concepts.

Part 1: Understand LLM capabilities

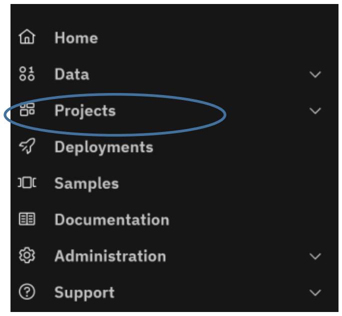
In this lab we will use the **Prompt Lab** in *watsonx.ai* to interact with LLMs included in the platform.

Typically, users (prompt engineers or data scientists) have three goals in this phase of the LLM lifecycle:

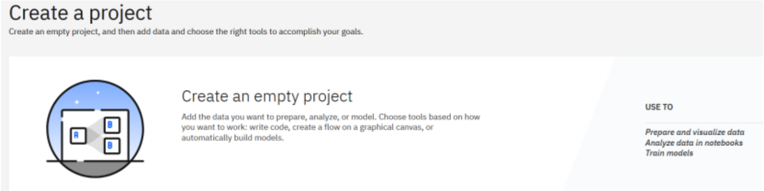
- Find if LLMs can be used for the proposed use case
- Identify the best model and parameters
- Create prompts for the use case.

1. Log in to your **IBM Cloud account**. Your lab instructor will provide the URL and userid/password.

2. From the main menu in the top left corner select **Projects -> View All Projects**.



3. Click on the **New Project** button. Select **Empty Project** and add your initials to the project name. For example, *LLM-workshop-YI*.



Create a project

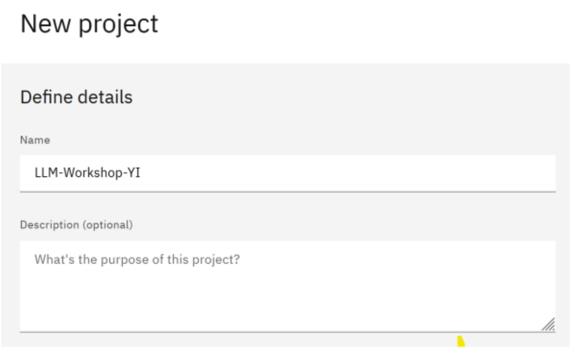
Create an empty project, and then add data and choose the right tools to accomplish your goals.

Create an empty project

Add the data you want to prepare, analyze, or model. Choose tools based on how you want to work: write code, create a flow on a graphical canvas, or automatically build models.

USE TO

Prepare and visualize data
Analyze data in notebooks
Train models



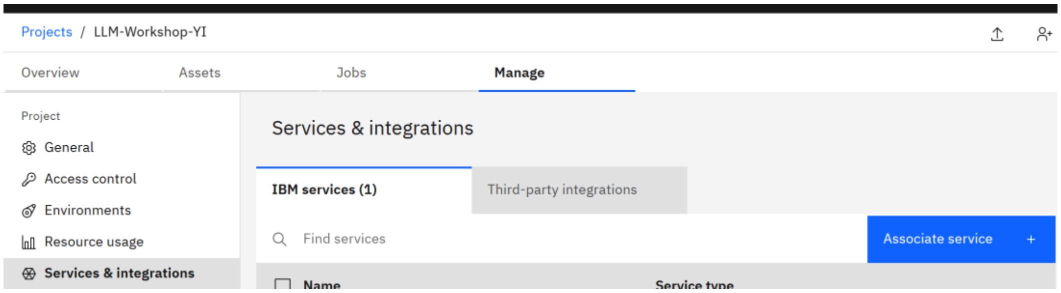
New project

Define details

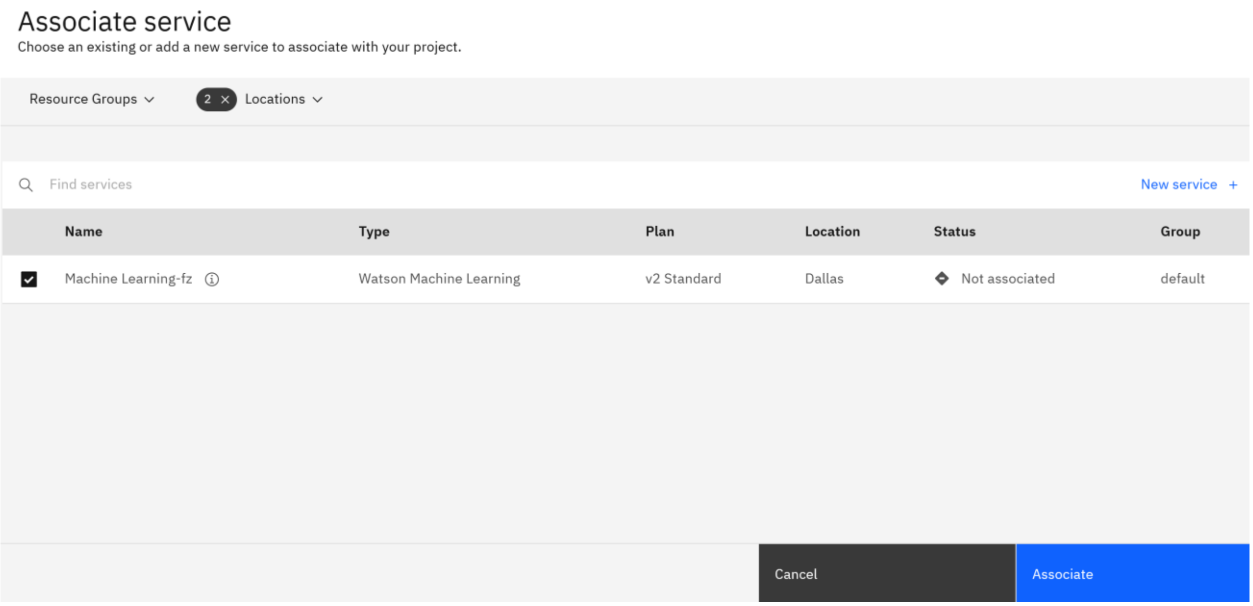
Name: LLM-Workshop-YI

Description (optional): What's the purpose of this project?

4. Switch to the **Services and Integrations** tab, then click **Associate Service**.



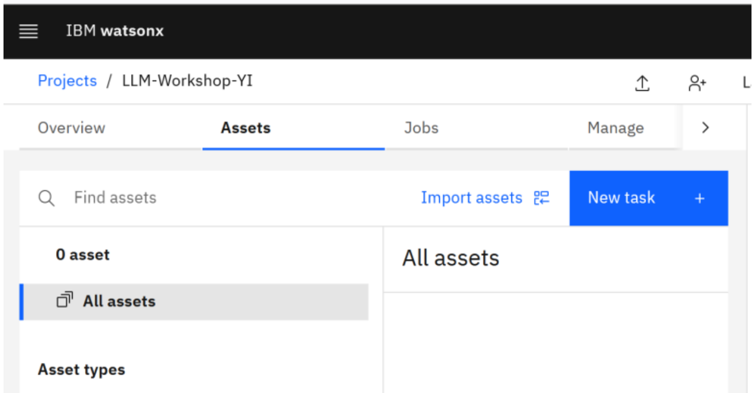
5. Select the displayed **Machine Learning** service and click **Associate**.


Associate service

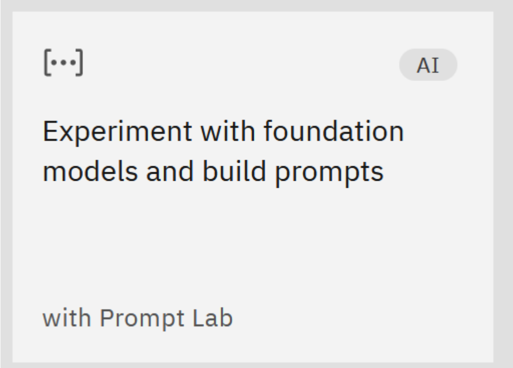
Choose an existing or add a new service to associate with your project.

	Name	Type	Plan	Location	Status	Group
☑	Machine Learning-fz	Watson Machine Learning	v2 Standard	Dallas	Not associated	default

6. Switch to the **Assets** tab, then click the **New task** button.



7. Click on the **Experiment with foundation models…** tile.


[···] AI

Experiment with foundation models and build prompts

with Prompt Lab

Before we start experimenting with prompting, let’s review some key concepts about **model selection**.

As you try different models, you will notice that some models return better results with zero-shot prompting (providing instructions without examples) than others. Usually models that have gone through *fine-tuning*, *instruction-tuning*, and *RLHF* generate significantly better output.

- *Fine-tuning* means that the original LLM was trained with high quality labeled data for specific use cases. For example, if our goal for the model was to “Act as an IT architect” when generating output, during the fine-tuning process we provided labeled data examples of a writing style for IT architecture.
- If the model goes through the instruction-tuning process, then it will be able to generate output without explicit instructions, such as *“Can you answer this question?”* The model will understand that you’re asking a question from the context and sentence structure.
- *RLHF* (*Reinforcement Learning from Human Feedback*) is a technique that’s used to improve model output based on feedback provided by testers, usually domain experts (for example, lawyers for generation of legal documents). Before the model is released, it’s updated based on testing results.

While all vendors can say that their model has been fine-tuned, instruction-tuned, and has gone through RLHF, the industry benchmarks for LLMs are not mature. Even benchmarks that may eventually become the industry standard (for example, TruthfulQA) test only certain aspects of model output.

A potential solution to this issue is the research done by the broader LLM community. The LLM community is very active; information about the quality of models usually becomes widely known through various community resources such as articles, blogs, and YouTube videos. This is especially true for open source vs. proprietary models. For example, search for “llama vs. ChatGPT” and review results. You can also review the leaderboard on Hugging Face, keeping in mind that you will need to understand the evaluation criteria used for the leaderboard (see the About page of the leaderboard).

At this time, *llama-70b-2-chat* model, is one of the best models for zero-shot prompting. While it may seem like an obvious choice to always use *llama-70b-2-chat* in watsonx.ai, it may not be possible for several reasons:

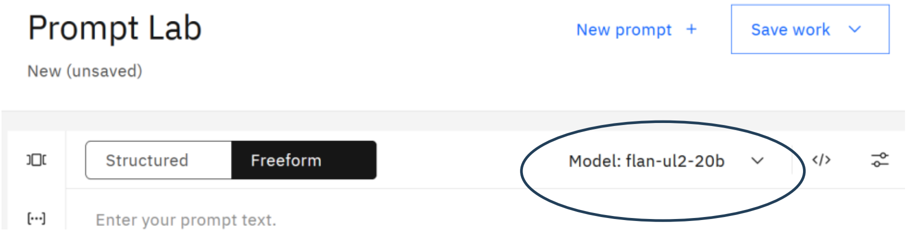
- Model availability in the data center (due to resources or licensing)
- Inference cost
- Hosting cost (for on-premises or hybrid cloud deployments).

It may be possible to achieve similar results with other models or with smaller versions of *llama* by using few-shot prompting or fine-tuning, that's why it's important to experiment with multiple models and understand prompt/turning techniques.

Note: Instructions in this lab are written for the flan and llama models, which are available in all IBM Cloud data centers where watsonx.ai is hosted and in the on-premises version of watsonx.ai. We encourage you to try other models (for example, granite and mpt-7b), if they're available in your workshop environment.

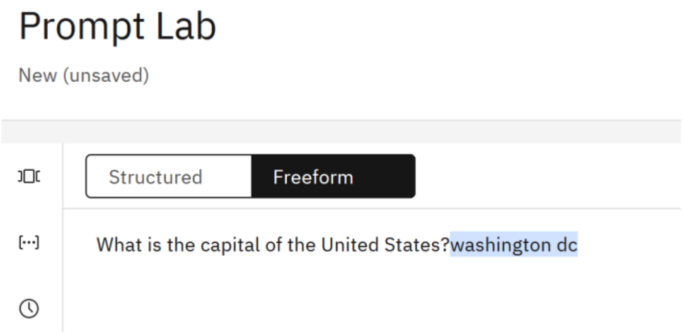
8. After the **Prompt Lab** UI opens, switch to the **Freeform** tab.

   Select the *flan-ul2-20b* model. We will review model settings later in the lab.



Since most LLMs, including the selected *flan* model were trained on publicly available data, we can ask it some general questions.

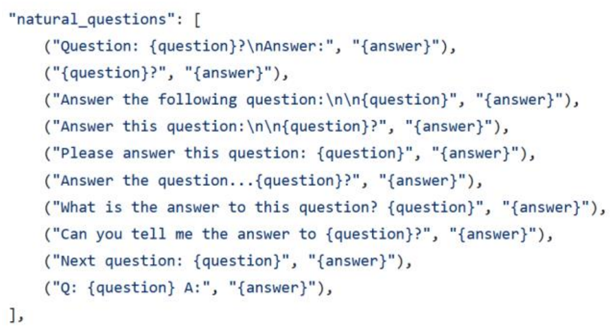
9. Type in the question: *"What is the capital of the United States?"* and click **Generate**. The generated answer is highlighted in blue.



What is the capital of the United States?washington dc

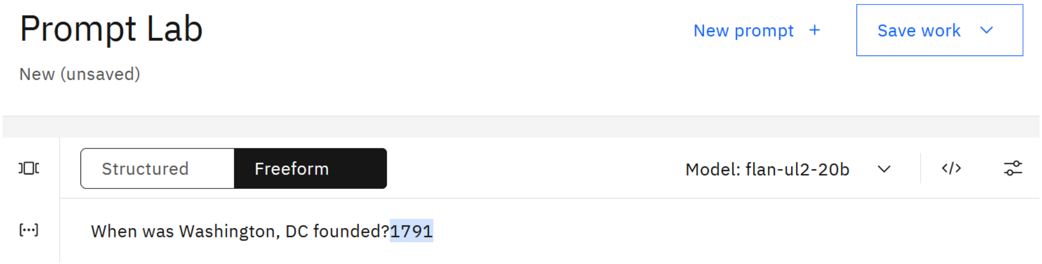
We got the answer to our question without instructing the model to do it because the *flan* model was *instruction-tuned* to answer questions. *Google*, the creator of this model, published the training instructions that were used for the model in this git repository. As you can see in documentation, the instructions are often shown in a "technical format", but they are still helpful for understanding the best prompting options for this model.

Scroll down to the *natural questions* section of the git page. Here we can see the various phrases we can use with the model when asking questions.



```
"natural_questions": [
    ("Question: {question}?\nAnswer:", "{answer}"),
    ("{question}?", "{answer}"),
    ("Answer the following question:\n\n{question}", "{answer}"),
    ("Answer this question:\n\n{question}?", "{answer}"),
    ("Please answer this question: {question}", "{answer}"),
    ("Answer the question...{question}?", "{answer}"),
    ("What is the answer to this question? {question}", "{answer}"),
    ("Can you tell me the answer to {question}?", "{answer}"),
    ("Next question: {question}", "{answer}"),
    ("Q: {question} A:", "{answer}"),
],
```

Next, we will ask a different question: "*When was Washington, DC founded?"*

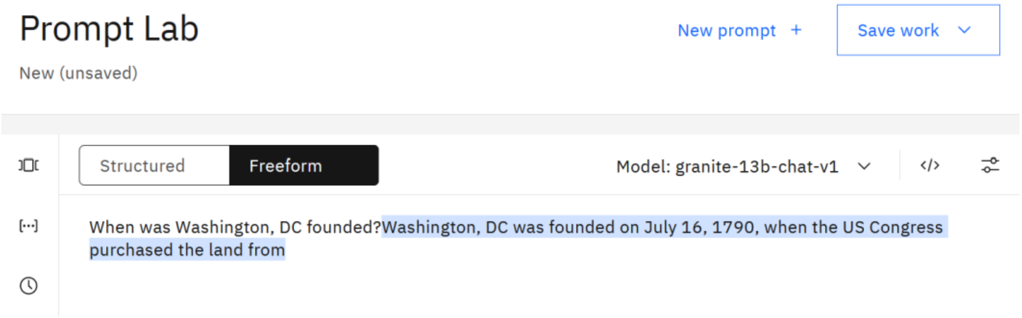


Model: flan-ul2-20b

When was Washington, DC founded?1791

Double check if this is a correct answer by doing a traditional Internet search. You will find out that the correct answer is *July 16th, 1790*.

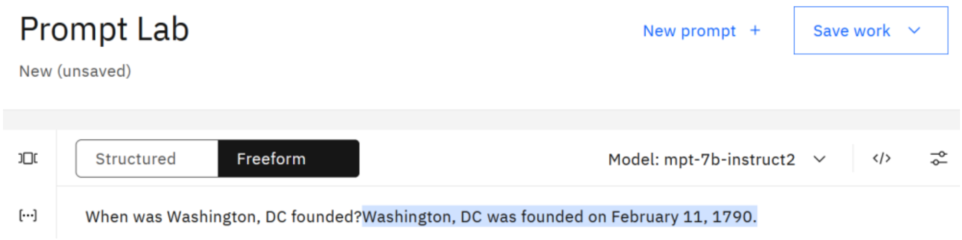
Next, switch the model to *granite*, then *llama2-70b-chat* and ask the question again. This time we get the correct answer.



Model: granite-13b-chat-v1

When was Washington, DC founded?Washington, DC was founded on July 16, 1790, when the US Congress purchased the land from

Next, switch the model to *mpt* and ask the question again - we get another incorrect answer.



Model: mpt-7b-instruct2

When was Washington, DC founded?Washington, DC was founded on February 11, 1790.

In general, this was not a simple question because at the time of writing conflicting dates are listed on the *Library of Congress* website and *Wikipedia*. *Library of Congress* is a more credible source, and in this example the *Wikipedia* page, which may have been used for model training, has the incorrect date.

We provided this example to highlight the fact that the primary usage of LLMs should not be *general knowledge question and answer*. The quality of LLM output depends on the knowledge base that it was trained on. If we asked another question, it’s possible that *flan* would outperform other models.

We should think of LLMs as an “engine” that can work with unstructured data rather than a “knowledge base”.

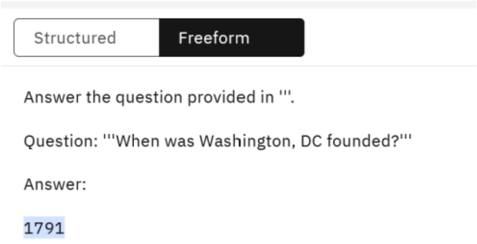
When you first start working with LLMs, you may think that some models are not returning the correct response because of the prompt format. Let’s test this theory with the *flan* model.

Enter this prompt in the **Prompt Lab**:

> *Answer the question provided in '''.*
>
> *Question: '''When was Washington, DC founded?'''*
>
> *Answer:*

Let’s review why we constructed the prompt in this format:

- Triple single quotes (‘’’) are often used to identify a question or text that we want the LLM to use. You can choose other characters, but avoid “ (double quotes) because they may already be in the provided text
- Notice that we provided the word “*Answer*:” at the end. Remember that LLMs *generate the next probable word*”, and providing the word “Answer” is a “hint” for the model.



Answer the question provided in '''.

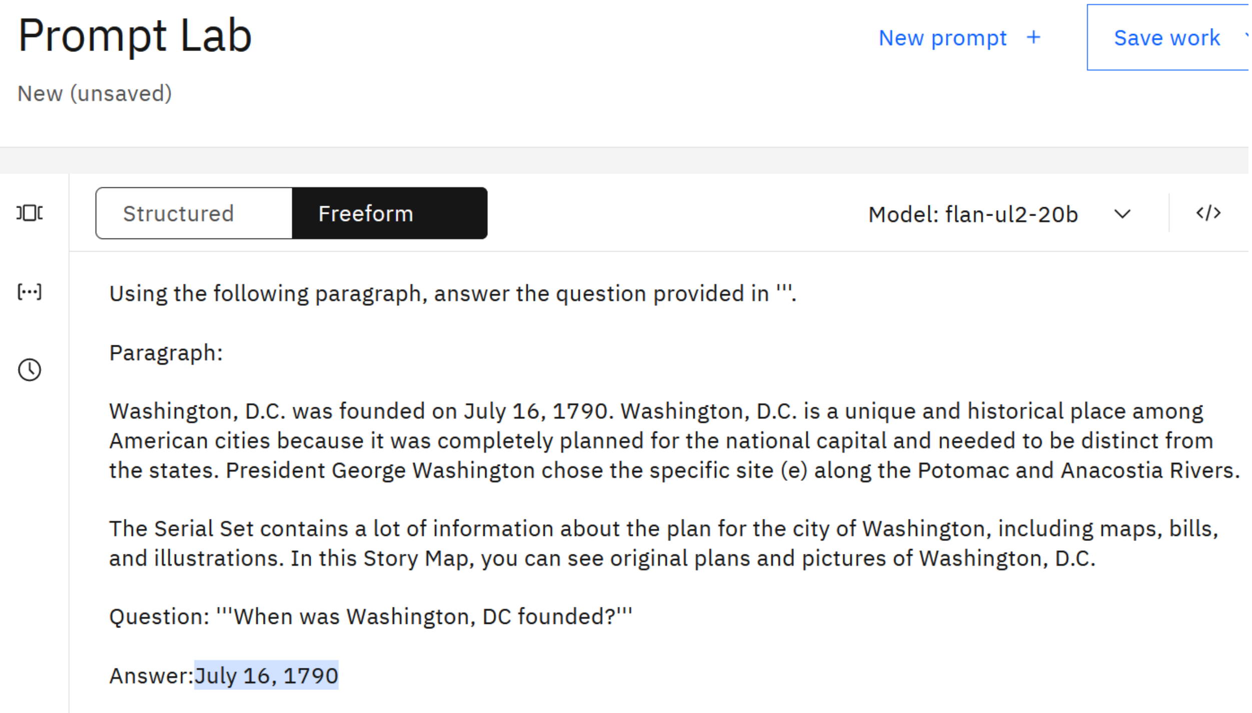
Question: '''When was Washington, DC founded?'''

Answer:

1791

Unfortunately, we did not get a more accurate result from the *flan* model.

We will try one more approach, this time with a different prompt, which you can copy from *Washington_DC_prompts.txt* in the *git repo /Prompts* folder.



```
Using the following paragraph, answer the question provided in '''.

Paragraph:

Washington, D.C. was founded on July 16, 1790. Washington, D.C. is a unique and historical place among American cities because it was completely planned for the national capital and needed to be distinct from the states. President George Washington chose the specific site (e) along the Potomac and Anacostia Rivers.

The Serial Set contains a lot of information about the plan for the city of Washington, including maps, bills, and illustrations. In this Story Map, you can see original plans and pictures of Washington, D.C.

Question: '''When was Washington, DC founded?'''

Answer:July 16, 1790
```

While this example may seem simple because we provide the answer in our prompt, it demonstrates one of the key use cases for LLMs, which is called *Retrieval Augmented Generation (RAG)*. With RAG, we ask LLMs to answer questions or generate content based on the information that we provide. In our example we hardcoded content in the prompt, but it's also possible to implement RAG with automatic information retrieval from various knowledge bases, such as Websites, documents, emails, etc. In this case, the main feature of an LLM that we are interested in is "understanding" and not "knowledge".

We used a simple example to ask a trivia question, but think about topics that may be relevant to your business for which "general information" may also exist, for example:

- What are the steps to get a driver's license?
- What are the steps for submitting a car insurance claim?
- How can I close a credit card?
- How can I improve my credit score?
- Will an airline reimburse me for a canceled flight?

Most LLMs will be able to answer these questions because they were trained on data available on the Internet, but if you want the correct answer to your question, in most cases you will need to use *RAG*, i.e. provide information from your company's data sources.

Watsonx.ai is supports several implementations of RAG. We will cover it in more detail in one of the other labs.

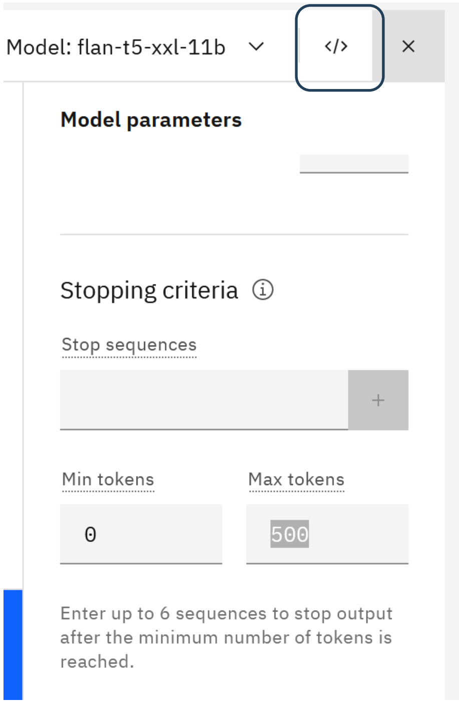
Next, we'll test prompts that generate output.

10. Click on model parameters icon in the top right corner.

    If you would like to learn more about each input in the Model parameters panel, you can review documentation.

    Change the **Max tokens** to *500*. When LLMs process instructions and generate output, they convert words to tokens (a sequence of characters). While there isn't a static ratio for letter to token conversion, we can use *10 words = 15 to 20* tokens as a rule of thumb for conversion.

    Use the *flan* model for your first test.



Model: flan-t5-xxl-11b

Min tokens: 0

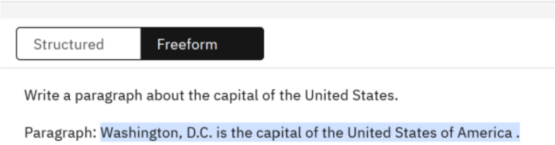
Max tokens: 500

Enter up to 6 sequences to stop output after the minimum number of tokens is reached.

11. Change the prompt to write a paragraph:

    > *Write a paragraph about the capital of the United States.*
    >
    > *Paragraph:*

    Notice that our output is rather brief.



Write a paragraph about the capital of the United States.

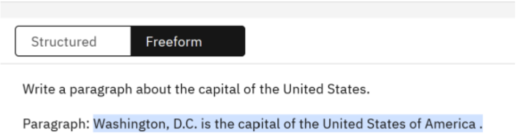
Paragraph: Washington, D.C. is the capital of the United States of America .

Next, we will try different model parameters and models to see if we can get better results.

12. In the model settings switch **Decoding** from *Greedy* to *Sampling*. Sampling will produce greater variability/creativity in generated content (see documentation for more information).

    *Important: Make sure to delete the generated text after the word "Paragraph"**:** before clicking **Generate** again because the model will continue generating after any given text in a prompt, which may result in repetition.*

    Click the **Generate** button. It looks like we're not getting better results with this model, so let's try another one.

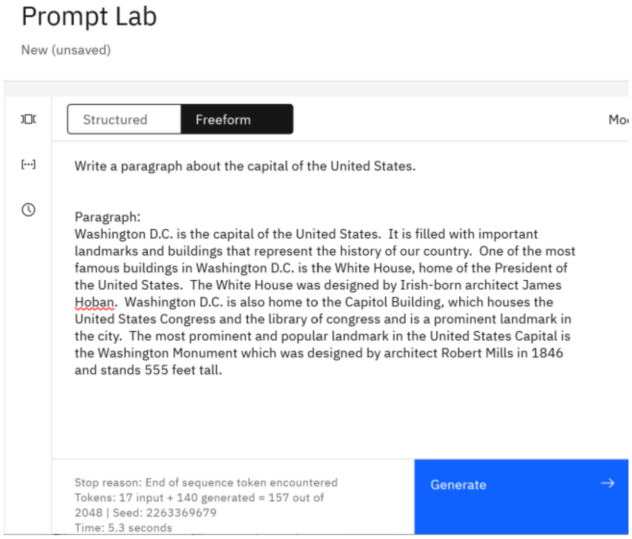


Write a paragraph about the capital of the United States.

Paragraph: Washington, D.C. is the capital of the United States of America .

13. Test the same prompt with the *granite* and *mpt-7b-instruct-2* models. Delete the generated text and test again.

    In our testing, we get better results with this model. Notice that every time you click **Generate** (after deleting the generated text), you get different results. We're seeing this because we set the *Decoding* option to *Sampling*. You can also try *Sampling* with different temperature (higher value will result in more variability).



Write a paragraph about the capital of the United States.

Paragraph:
Washington D.C. is the capital of the United States. It is filled with important landmarks and buildings that represent the history of our country. One of the most famous buildings in Washington D.C. is the White House, home of the President of the United States. The White House was designed by Irish-born architect James Hoban. Washington D.C. is also home to the Capitol Building, which houses the United States Congress and the library of congress and is a prominent landmark in the city. The most prominent and popular landmark in the United States Capital is the Washington Monument which was designed by architect Robert Mills in 1846 and stands 555 feet tall.

Stop reason: End of sequence token encountered
Tokens: 17 input + 140 generated = 157 out of 2048 | Seed: 2263369679
Time: 5.3 seconds

    While it may seem unusual that the model generates a different output each time, it's what we're instructing the model to do by both giving it instructions ("*write*") and setting model parameters ("*sampling*"). We would not use the same instructions/parameters for a classification use case which needs to provide consistent output (for example, *positive*, *negative*, or *neutral* sentiment).

14. Finally, try the *llama* model. For "creative output" use cases, *llama* usually produces the best output.

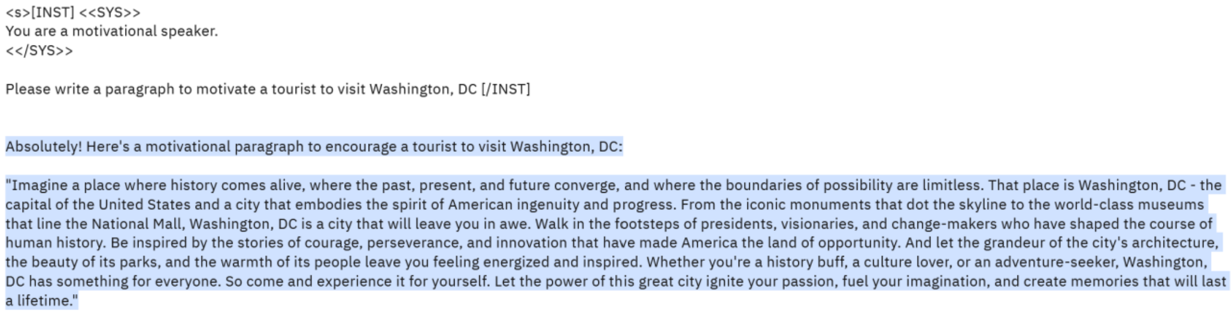
While you can you use the same prompt, we should also be familiar with *system prompt format* in *llama*, which is one of the few models that accepts prompts in a specific format.

Enter this prompt, which you can copy from the from *Washington_DC_prompts.txt* in the *git repo /Prompts* folder.

<s>[INST] <<SYS>>
You are a motivational speaker.
<</SYS>>

Please write a paragraph to motivate a tourist to visit Washington, DC [/INST]

Important note: If you decide to use this format for llama, make sure not to put anything after the [/INST] In this format you do not need to give the model a "hint" (Answer: to generate the answer, etc.), like you've done in other examples.



<s>[INST] <<SYS>>
You are a motivational speaker.
<</SYS>>

Please write a paragraph to motivate a tourist to visit Washington, DC [/INST]

Absolutely! Here's a motivational paragraph to encourage a tourist to visit Washington, DC:

"Imagine a place where history comes alive, where the past, present, and future converge, and where the boundaries of possibility are limitless. That place is Washington, DC - the capital of the United States and a city that embodies the spirit of American ingenuity and progress. From the iconic monuments that dot the skyline to the world-class museums that line the National Mall, Washington, DC is a city that will leave you in awe. Walk in the footsteps of presidents, visionaries, and change-makers who have shaped the course of human history. Be inspired by the stories of courage, perseverance, and innovation that have made America the land of opportunity. And let the grandeur of the city's architecture, the beauty of its parks, and the warmth of its people leave you feeling energized and inspired. Whether you're a history buff, a culture lover, or an adventure-seeker, Washington, DC has something for everyone. So come and experience it for yourself. Let the power of this great city ignite your passion, fuel your imagination, and create memories that will last a lifetime."

Similar to the first prompting exercise, we started with LLMs' general knowledge for generating output. In a business use case scenario, we would give LLMs a few short bullet points and ask it to generate output.

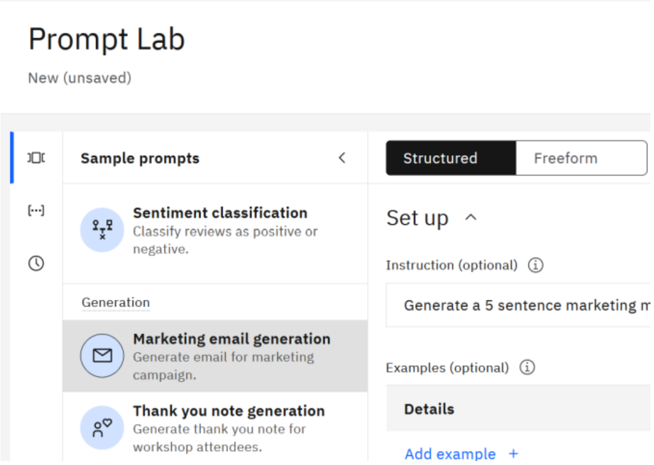
If you wish to continue with the Washington, DC example, you can use this prompt from the sample prompt file:

<s>[INST] <<SYS>>
You are a marketing consultant.
<</SYS>>

Please generate a promotional email to visit the following attractions in Washington, DC:
1. The National Mall
2. The Smithsonian Museums
3. The White House
4. The U.S. Capitol
5. The National Gallery of Art
[/INST]

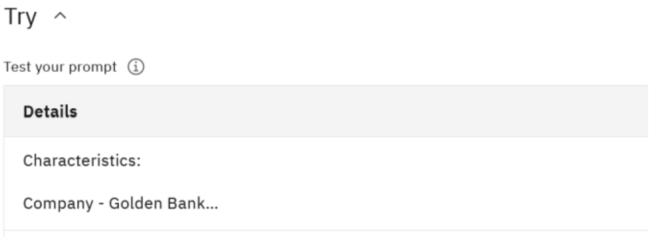
Another example of generation is provided in *Sample Prompts* included with the **Prompt Lab**.

15. Click on the *Sample Prompts* icon and scroll down to the *Marketing email generation* example.


Prompt Lab
New (unsaved)
Sample prompts
Structured
Freeform
Sentiment classification
Classify reviews as positive or negative.
Set up
Instruction (optional)
Generation
Marketing email generation
Generate email for marketing campaign.
Examples (optional)
Details
Thank you note generation
Generate thank you note for workshop attendees.
Add example +

Generate a 5 sentence marketing m

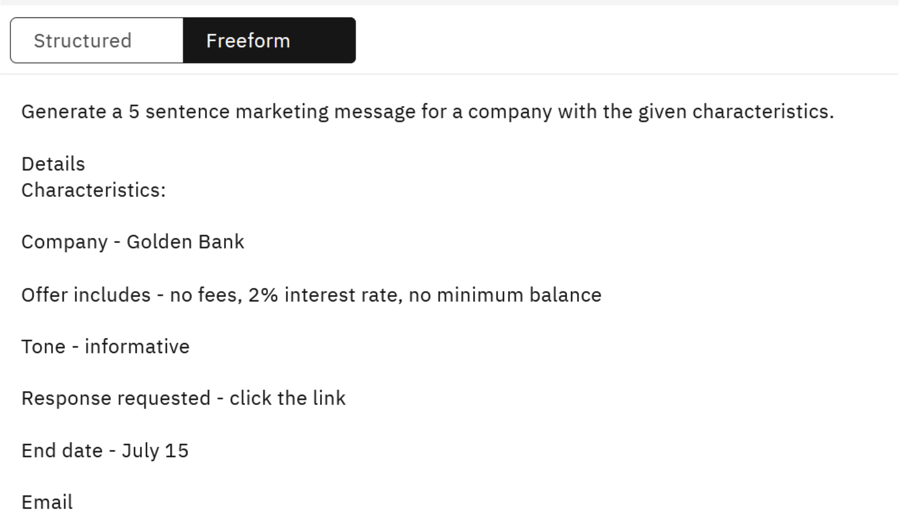
You can run this example from either the *Structured* or *Freeform* view. If you use the *Structured* view, the input for generation is shown under *Details*.



Characteristics:

Company - Golden Bank...

If you wish, change the input for the generated email and test the prompt.



Generate a 5 sentence marketing message for a company with the given characteristics.

Details
Characteristics:

Company - Golden Bank

Offer includes - no fees, 2% interest rate, no minimum balance

Tone - informative

Response requested - click the link

End date - July 15

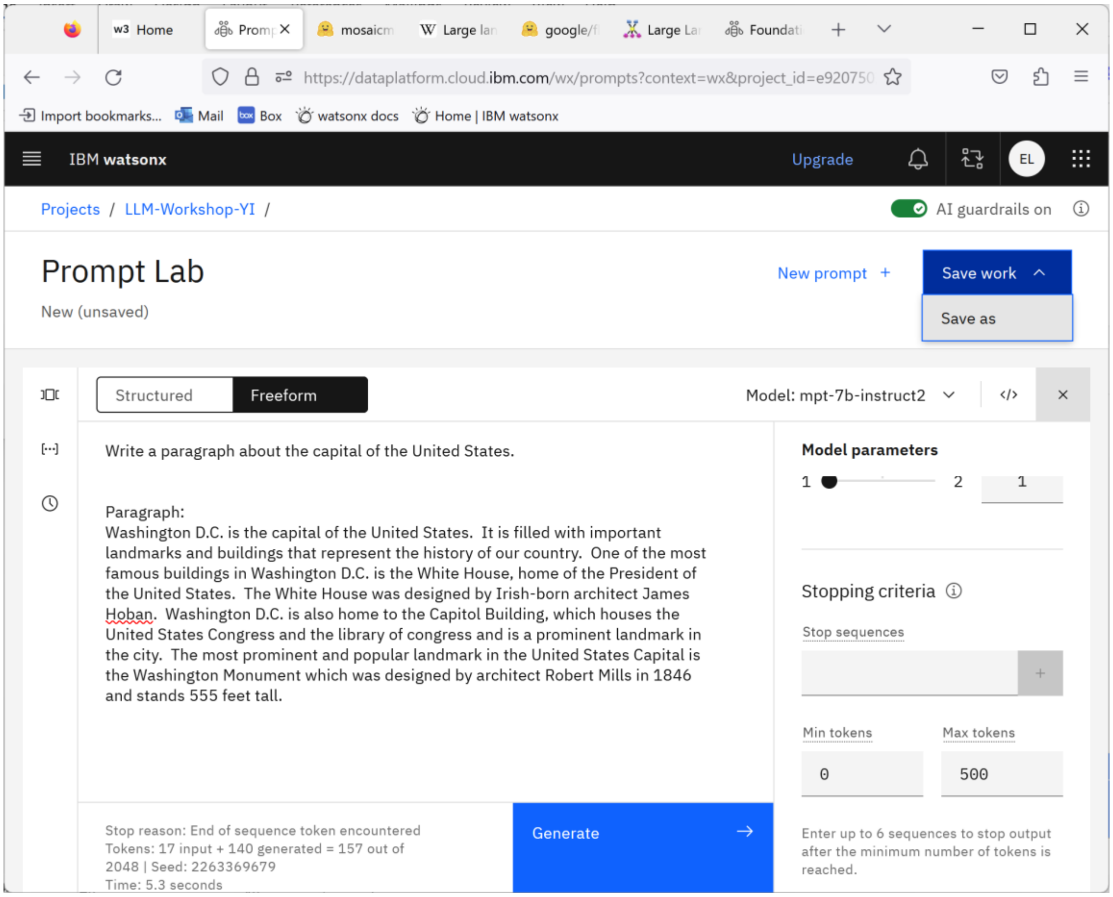
Email

You may have already noticed that working with LLMs requires experimentation. In the **Prompt Lab** we can save the results of our experimentation with prompts

- As a *notebook*
- As a *prompt*
- As a *prompt session*.

If we save our experimentation as a prompt session, we will be able to access various prompts and the output that was generated.

In the **Prompt Lab**, select **Save work -> Save as -> Prompt session**.



Write a paragraph about the capital of the United States.

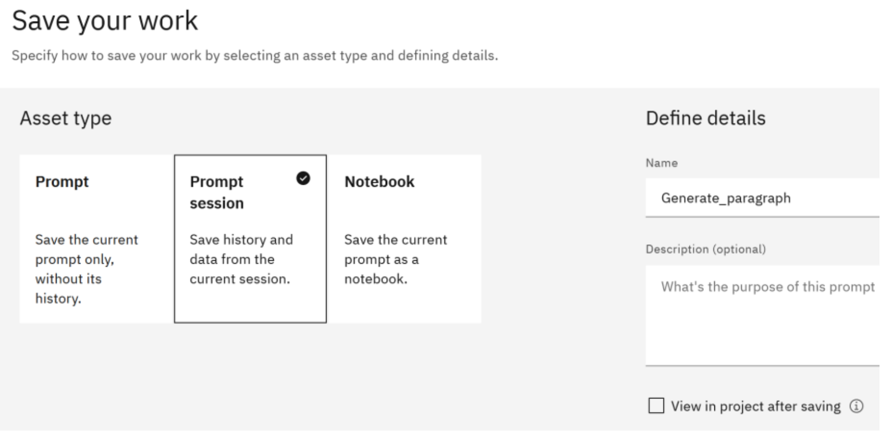
Paragraph:
Washington D.C. is the capital of the United States. It is filled with important landmarks and buildings that represent the history of our country. One of the most famous buildings in Washington D.C. is the White House, home of the President of the United States. The White House was designed by Irish-born architect James Hoban. Washington D.C. is also home to the Capitol Building, which houses the United States Congress and the library of congress and is a prominent landmark in the city. The most prominent and popular landmark in the United States Capital is the Washington Monument which was designed by architect Robert Mills in 1846 and stands 555 feet tall.

Min tokens: 0
Max tokens: 500

Enter up to 6 sequences to stop output after the minimum number of tokens is reached.

Stop reason: End of sequence token encountered
Tokens: 17 input + 140 generated = 157 out of 2048 | Seed: 2263369679
Time: 5.3 seconds

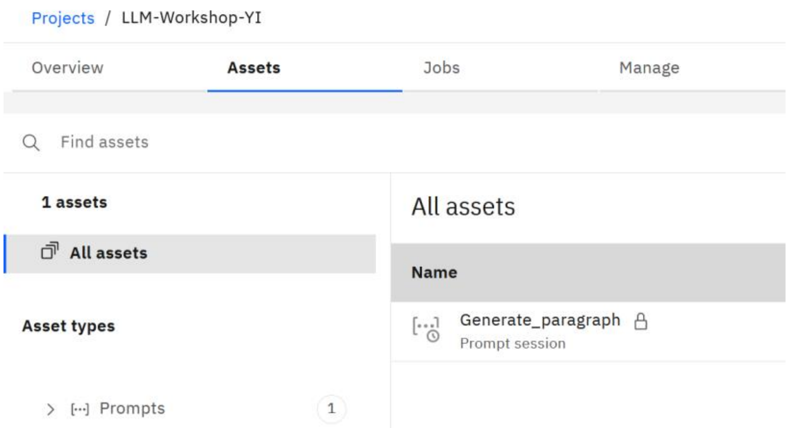
Name the prompt session *Generate_paragraph*.


Save your work
Specify how to save your work by selecting an asset type and defining details.
Asset type
Prompt
Save the current prompt only, without its history.
Prompt session
Save history and data from the current session.
Notebook
Save the current prompt as a notebook.
Define details
Description (optional)
What's the purpose of this prompt
View in project after saving

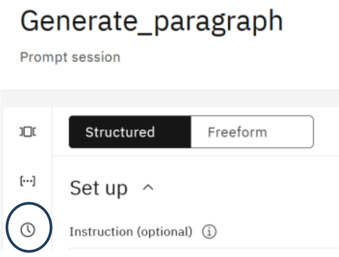
Name
Generate_paragraph

16. Open **watsonx.ai** in another browser window and navigate to your project.

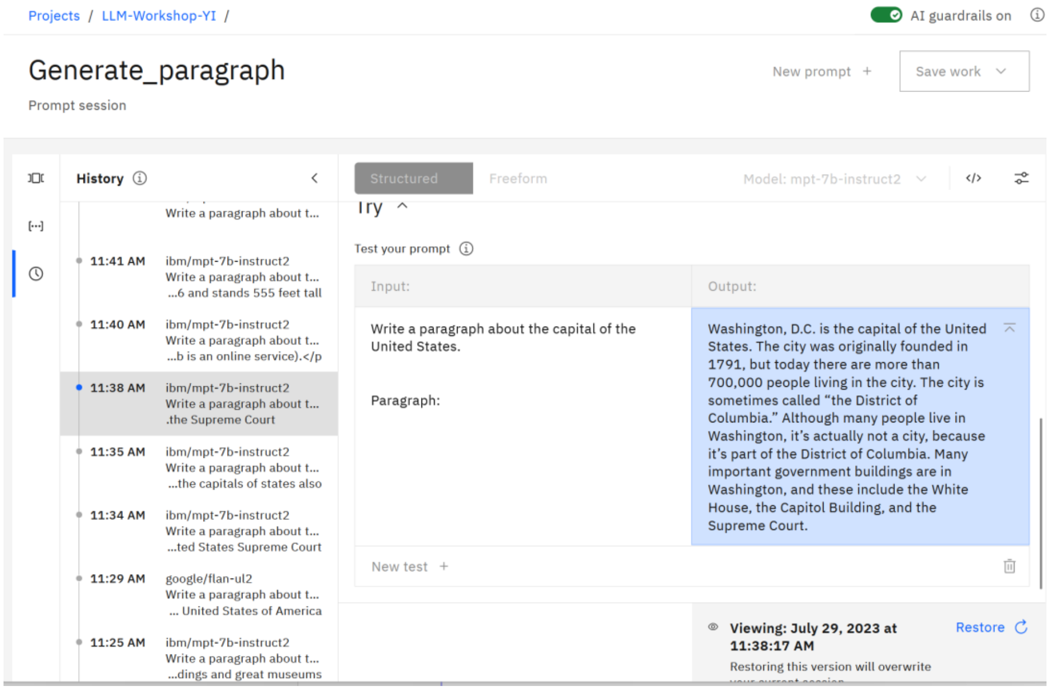
17. Click on the **Assets** tab and open the prompt session asset that you created.



18. In the **Prompt Lab**, click on the *History* icon.



Notice that you can click on various prompts that you tested and view the output in the **Test your promp**t section of the **Prompt Lab**.



Generate_paragraph

Prompt session

Time	Entry
	Write a paragraph about t...
11:41 AM	ibm/mpt-7b-instruct2 Write a paragraph about t... ...6 and stands 555 feet tall
11:40 AM	ibm/mpt-7b-instruct2 Write a paragraph about t... ...b is an online service).</p
11:38 AM	ibm/mpt-7b-instruct2 Write a paragraph about t... .the Supreme Court
11:35 AM	ibm/mpt-7b-instruct2 Write a paragraph about t... ...the capitals of states also
11:34 AM	ibm/mpt-7b-instruct2 Write a paragraph about t... ...ted States Supreme Court
11:29 AM	google/flan-ul2 Write a paragraph about t... ... United States of America
11:25 AM	ibm/mpt-7b-instruct2 Write a paragraph about t... ...dings and great museums

Input:	Output:
Write a paragraph about the capital of the United States. Paragraph:	Washington, D.C. is the capital of the United States. The city was originally founded in 1791, but today there are more than 700,000 people living in the city. The city is sometimes called "the District of Columbia." Although many people live in Washington, it's actually not a city, because it's part of the District of Columbia. Many important government buildings are in Washington, and these include the White House, the Capitol Building, and the Supreme Court.

Viewing: July 29, 2023 at 11:38:17 AM

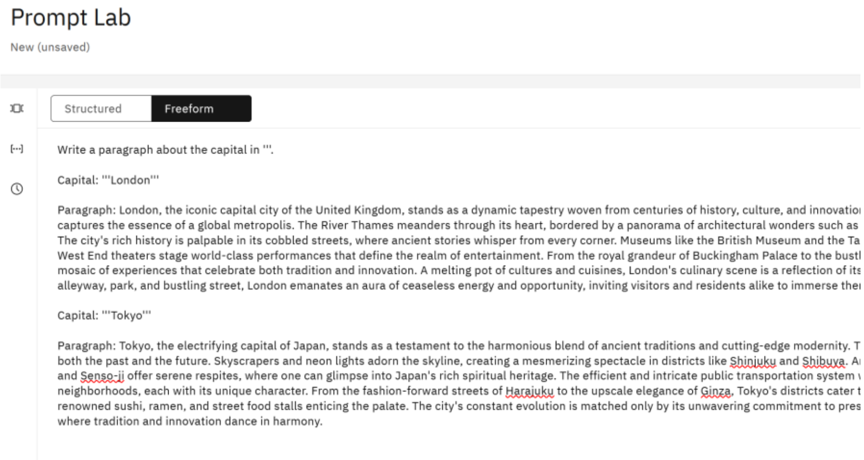
Restoring this version will overwrite

Close this browser tab and return to the **Prompt Lab**.

What you have tried so far is a "*question and answer and generation use case with zero-shot prompting*" – you've asked the LLM to generate output without providing any examples. The majority of LLMs produce better output when they're given a few examples. This technique is called "few-shot prompting". The examples are provided in the prompt after the instruction.

Let's test *few-shot prompting* for various use cases.

19. In the **Prompt Lab**, create a new prompt. Switch to **Freeform**, and paste the following prompt (this prompt is available in the *Paragraph_few_shot.txt* file in the */Prompts* folder).



Prompt Lab

New (unsaved)

Write a paragraph about the capital in '''.

Capital: '''London'''

Paragraph: London, the iconic capital city of the United Kingdom, stands as a dynamic tapestry woven from centuries of history, culture, and innovatio
captures the essence of a global metropolis. The River Thames meanders through its heart, bordered by a panorama of architectural wonders such as
The city's rich history is palpable in its cobbled streets, where ancient stories whisper from every corner. Museums like the British Museum and the Ta
West End theaters stage world-class performances that define the realm of entertainment. From the royal grandeur of Buckingham Palace to the bustl
mosaic of experiences that celebrate both tradition and innovation. A melting pot of cultures and cuisines, London's culinary scene is a reflection of its
alleyway, park, and bustling street, London emanates an aura of ceaseless energy and opportunity, inviting visitors and residents alike to immerse the

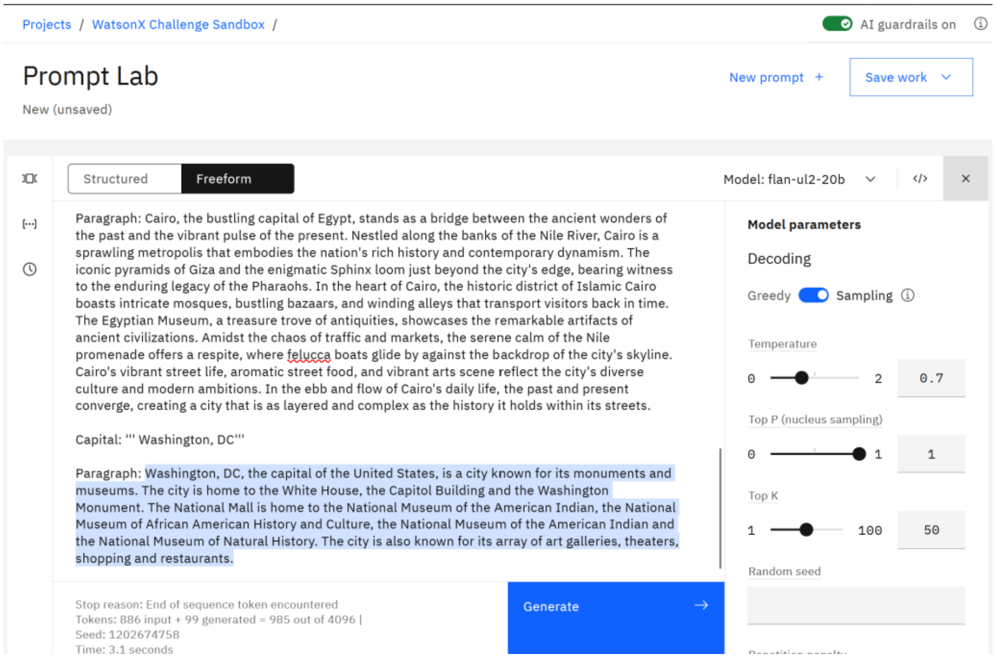
Capital: '''Tokyo'''

Paragraph: Tokyo, the electrifying capital of Japan, stands as a testament to the harmonious blend of ancient traditions and cutting-edge modernity. T
both the past and the future. Skyscrapers and neon lights adorn the skyline, creating a mesmerizing spectacle in districts like Shinjuku and Shibuya. A
and Senso-ji offer serene respites, where one can glimpse into Japan's rich spiritual heritage. The efficient and intricate public transportation system
neighborhoods, each with its unique character. From the fashion-forward streets of Harajuku to the upscale elegance of Ginza, Tokyo's districts cater t
renowned sushi, ramen, and street food stalls enticing the palate. The city's constant evolution is matched only by its unwavering commitment to pres
where tradition and innovation dance in harmony.

20. Modify model parameters:

    - Change *decoding* to *sampling* (for more creative output)
    - Change the min and max number of tokens to the output you would like to see (for example, *50 min* and *500 max*)
    - If you wish, you can test different models.

    Test the model and review the output.



Model: flan-ul2-20b

Paragraph: Cairo, the bustling capital of Egypt, stands as a bridge between the ancient wonders of the past and the vibrant pulse of the present. Nestled along the banks of the Nile River, Cairo is a sprawling metropolis that embodies the nation's rich history and contemporary dynamism. The iconic pyramids of Giza and the enigmatic Sphinx loom just beyond the city's edge, bearing witness to the enduring legacy of the Pharaohs. In the heart of Cairo, the historic district of Islamic Cairo boasts intricate mosques, bustling bazaars, and winding alleys that transport visitors back in time. The Egyptian Museum, a treasure trove of antiquities, showcases the remarkable artifacts of ancient civilizations. Amidst the chaos of traffic and markets, the serene calm of the Nile promenade offers a respite, where felucca boats glide by against the backdrop of the city's skyline. Cairo's vibrant street life, aromatic street food, and vibrant arts scene reflect the city's diverse culture and modern ambitions. In the ebb and flow of Cairo's daily life, the past and present converge, creating a city that is as layered and complex as the history it holds within its streets.

Capital: ''' Washington, DC'''

Paragraph: Washington, DC, the capital of the United States, is a city known for its monuments and museums. The city is home to the White House, the Capitol Building and the Washington Monument. The National Mall is home to the National Museum of the American Indian, the National Museum of African American History and Culture, the National Museum of the American Indian and the National Museum of Natural History. The city is also known for its array of art galleries, theaters, shopping and restaurants.

Stop reason: End of sequence token encountered
Tokens: 886 input + 99 generated = 985 out of 4096 |
Seed: 1202674758
Time: 3.1 seconds

Model parameters

Decoding

Greedy ● Sampling

Temperature 0 — 2: 0.7

Top P (nucleus sampling) 0 — 1: 1

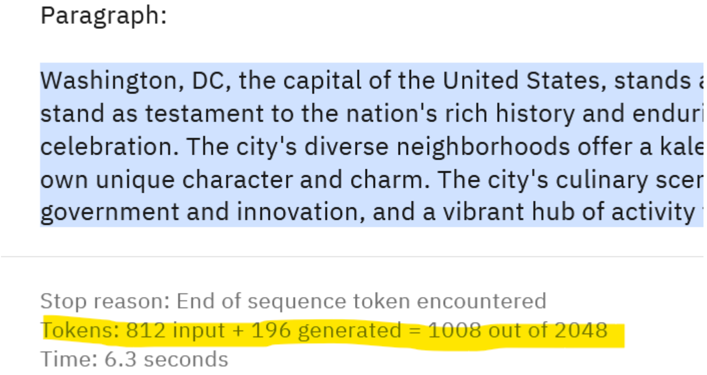
Top K 1 — 100: 50

Random seed

Next, we will review the concept of *tokens*.

21. Notice the token count that's shown on the bottom of the model output.

    In this screenshot of the *mpt* model output, the "*out of*" number (*2048*) shows the maximum number of tokens that can be processed by a model. If you test with a different model, the maximum number of tokens will be different.


Paragraph:

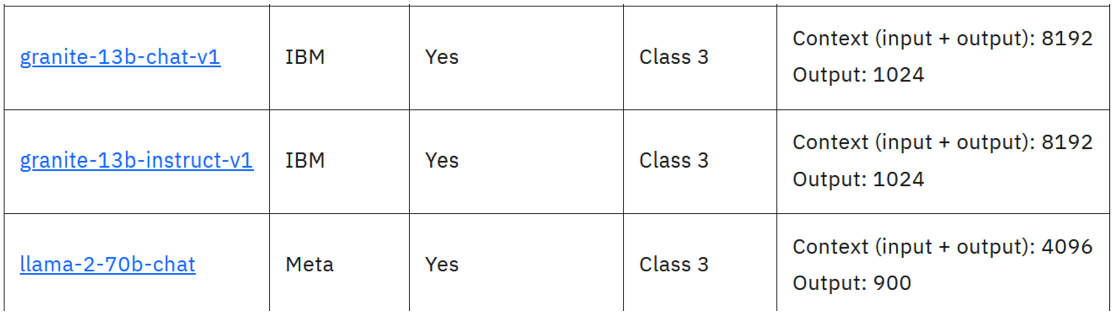
Washington, DC, the capital of the United States, stands
stand as testament to the nation's rich history and endur
celebration. The city's diverse neighborhoods offer a kale
own unique character and charm. The city's culinary sce
government and innovation, and a vibrant hub of activity

Stop reason: End of sequence token encountered
Tokens: 812 input + 196 generated = 1008 out of 2048
Time: 6.3 seconds

It’s important to understand the following facts about tokens:

- All LLMs have a limit for the number of supported tokens. The maximum number of tokens is usually captured in documentation or in the UI, as you’ve seen in the **Prompt Lab.**
- The maximum number of tokens includes both input and output tokens. This means that you can’t provide an unlimited number of examples in the prompt. In addition to that, each model has the maximum number of output tokens (see documentation).
- Some vendors have daily/monthly token limits for different plans, which should be considered when selecting an LLM platform.

Example of token limits (from documentation):



granite-13b-chat-v1	IBM	Yes	Class 3	Context (input + output): 8192 Output: 1024
granite-13b-instruct-v1	IBM	Yes	Class 3	Context (input + output): 8192 Output: 1024
llama-2-70b-chat	Meta	Yes	Class 3	Context (input + output): 4096 Output: 900

Understanding token constraints is especially important for summarization, generation, and Q&A use cases because they may require more tokens than classification, extraction, or sentiment analysis use cases.

The token constraint limitation can be solved with several approaches. If we need to provide more examples to the model, we can use an approach called *Multitask Prompt Tuning (MPT)* or *fine tuning*. We are not covering these advanced approaches in this introductory lab.

Up to this point we reviewed question & answer and generation examples. We started with these examples because for most users they are the “first experience” with generative AI. Many people are familiar with *ChatGPT*, a popular personal assistant application developed by *OpenAI*. Sometimes the terms *generative AI* and even *LLM* are used interchangeably with *ChatGPT*, but *ChatGPT* is more than an LLM, it’s a complex application that uses LLMs.

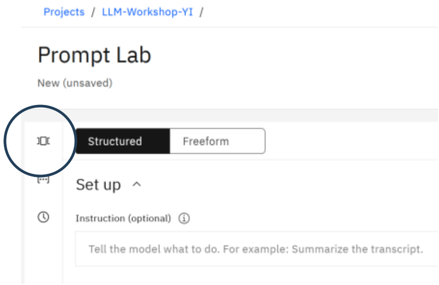
LLMs are building blocks or components of an application, and by themselves they can rarely be used by a business user. *ChatGPT* is a tool that focuses on personal productivity tasks for many types of users. Companies that want to build AI-driven applications need an AI development and deployment platform, such as *watsonx.ai*.

In our experience of working with clients, some of the top generative AI use cases are:

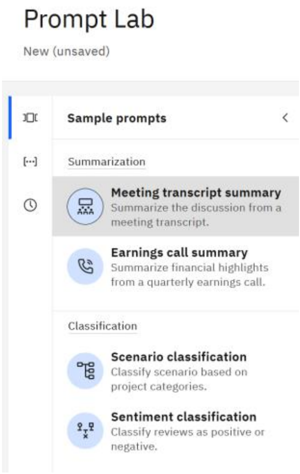
- *Content summarization*
- *Content classification*
- *Content generation*
- *Content extraction, NER (Named Entity Recognition)*
- *Sentiment analysis*
- *Question answering with RAG*

Let’s review sample prompts for a few more use cases.

22. In the **Prompt Lab**, create a new prompt and click on the **Sample prompts** icon.


Projects / LLM-Workshop-YI /
Prompt Lab
New (unsaved)
Structured
Freeform
Set up
Instruction (optional)
Tell the model what to do. For example: Summarize the transcript.

23. Explore/test the different prompts, which are organized by use cases.



Meeting transcript summary Summarize the discussion from a meeting transcript.

Earnings call summary Summarize financial highlights from a quarterly earnings call.

Scenario classification Classify scenario based on project categories.

Sentiment classification Classify reviews as positive or negative.

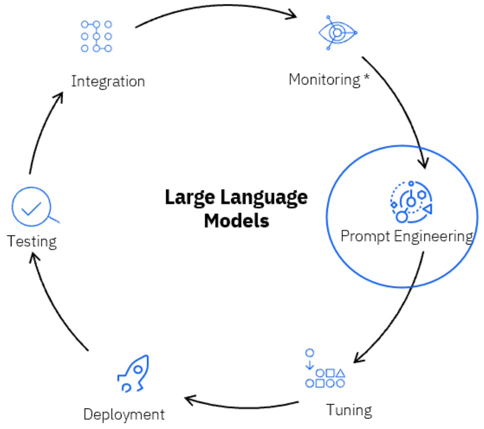
Notice that some examples use few-shot prompting (*Meeting transcript summar*y), while others (*Sentiment classification*) use zero-shot prompting.

If you wish, try a few sample prompts that we provided in the downloaded *git repo /Prompts* folder and sample prompts from documentation.

Now that you have reviewed and created prompts, we will test the integration of LLMs with client applications.

Part 2: Integrate LLMs with applications

Up to this point you have completed a simple prompt engineering exercise. Prompt engineering is just one of the steps in the process of integrating LLMs into business applications.



Let's review other steps:

- There are several types of tuning, which is usually applied to improve model output. In many use cases tuning won't be a required step.
- LLMs are pre-deployed (available for invocation out-of-the-box) in *watsonx.ai*. The only time deployment may be required is for tuned models.
- Testing and integration is done with either the REST API or the Python SDK.

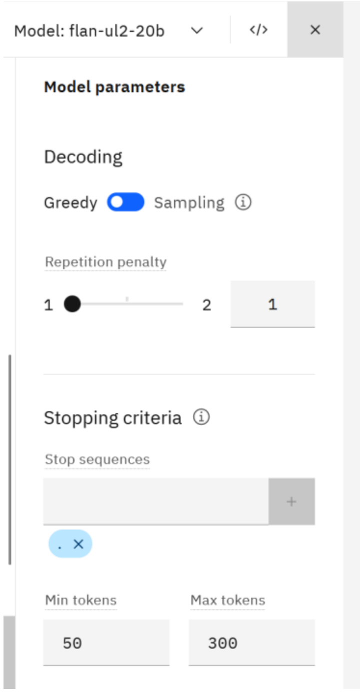
In this section we will review the testing and integration steps.

1. Navigate to the **Prompt lab** and open one of the prompts you previously created or one of the sample prompts.

   In this lab we will be using the loan summary prompt which you can find in */Prompts/Loan_few_shot_summary.txt.*

2. Generate a response using this prompt.

   - You can use either the *flan* of the *mpt* model
   - Keep the decoding method as *Greedy*
   - Add a stop sequence of "." to prevent output that ends with mid-sentence.
   - Make sure to set the min and max tokens to 50 and 300.



Model parameters

Decoding

Greedy Sampling

Repetition penalty

1 2 | 1

Stopping criteria

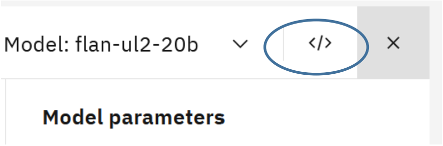
Stop sequences

. ×

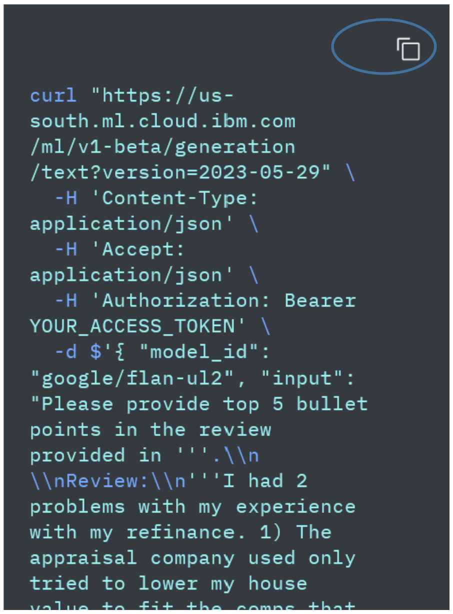
Min tokens 50

Max tokens 300

After testing the prompt (click **Generate**), click on the **View code** icon.



Copy the code to a notepad.



```
curl "https://us-
south.ml.cloud.ibm.com
/ml/v1-beta/generation
/text?version=2023-05-29" \
  -H 'Content-Type:
application/json' \
  -H 'Accept:
application/json' \
  -H 'Authorization: Bearer
YOUR_ACCESS_TOKEN' \
  -d $'{ "model_id":
"google/flan-ul2", "input":
"Please provide top 5 bullet
points in the review
provided in '''.\\n
\\nReview:\\n'''I had 2
problems with my experience
with my refinance. 1) The
appraisal company used only
tried to lower my house
value to fit the comps that
```

Let’s review the code.

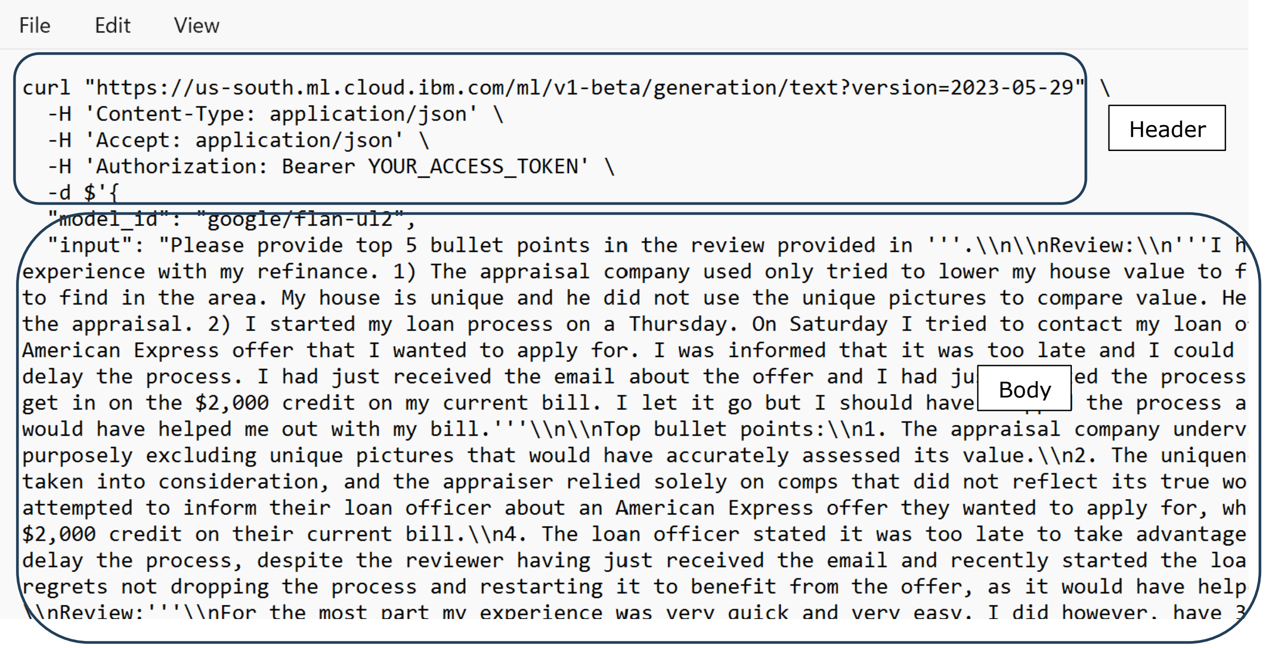
This code is an example of a REST call to invoke the model. **Watsonx.ai** also provides a Python API for model invocation, which we will review later in this lab.

The header of the REST request includes the URL where the model is hosted and a placeholder for the authentication token. At this time all users share a single model inference endpoint. In the future, IBM plans to provide dedicated model endpoints.

Note: IBM does not store model inference input/output data. In the future, users will be able to opt in to storing data.

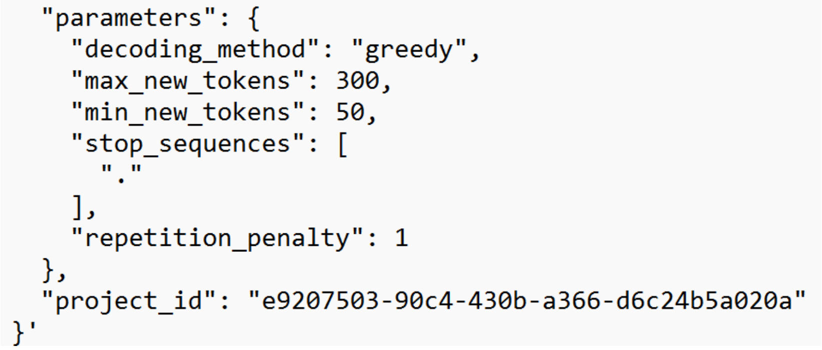
Security is managed by the *IBM Cloud authentication token*. We will get this token shortly.

The body of the request contains the entire prompt.



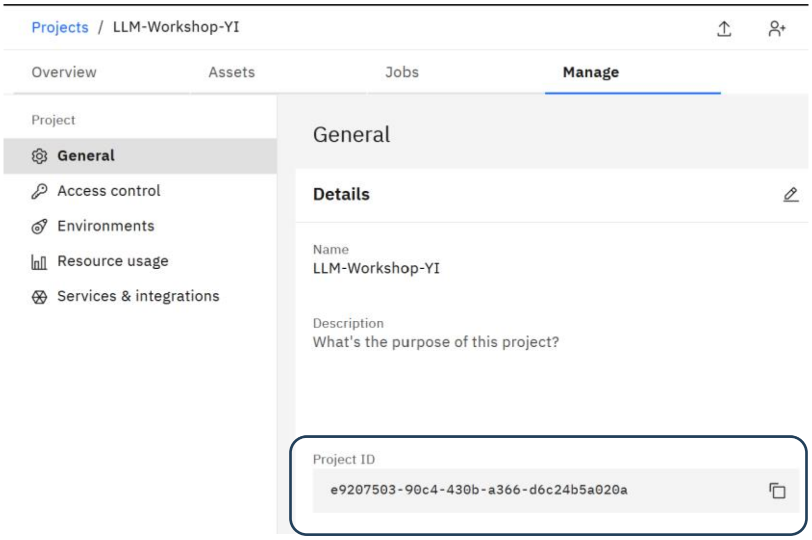
```
curl "https://us-south.ml.cloud.ibm.com/ml/v1-beta/generation/text?version=2023-05-29" \
  -H 'Content-Type: application/json' \
  -H 'Accept: application/json' \
  -H 'Authorization: Bearer YOUR_ACCESS_TOKEN' \
  -d $'{
  "model_id": "google/flan-ul2",
  "input": "Please provide top 5 bullet points in the review provided in '''.\\n\\nReview:\\n'''I h
experience with my refinance. 1) The appraisal company used only tried to lower my house value to f
to find in the area. My house is unique and he did not use the unique pictures to compare value. He
the appraisal. 2) I started my loan process on a Thursday. On Saturday I tried to contact my loan o
American Express offer that I wanted to apply for. I was informed that it was too late and I could
delay the process. I had just received the email about the offer and I had ju       ed the process
get in on the $2,000 credit on my current bill. I let it go but I should have        the process a
would have helped me out with my bill.'''\\n\\nTop bullet points:\\n1. The appraisal company underv
purposely excluding unique pictures that would have accurately assessed its value.\\n2. The uniquen
taken into consideration, and the appraiser relied solely on comps that did not reflect its true wo
attempted to inform their loan officer about an American Express offer they wanted to apply for, wh
$2,000 credit on their current bill.\\n4. The loan officer stated it was too late to take advantage
delay the process, despite the reviewer having just received the email and recently started the loa
regrets not dropping the process and restarting it to benefit from the offer, as it would have help
\\nReview:'''\\nFor the most part my experience was very quick and very easy. I did however, have 3
```

Finally, at the end of the request we specify model parameters and the *project id*.



```
  "parameters": {
    "decoding_method": "greedy",
    "max_new_tokens": 300,
    "min_new_tokens": 50,
    "stop_sequences": [
      "."
    ],
    "repetition_penalty": 1
  },
  "project_id": "e9207503-90c4-430b-a366-d6c24b5a020a"
}'
```

The *project id* can be looked up in the **Project -> Manage view** tab of the watxonx.ai project



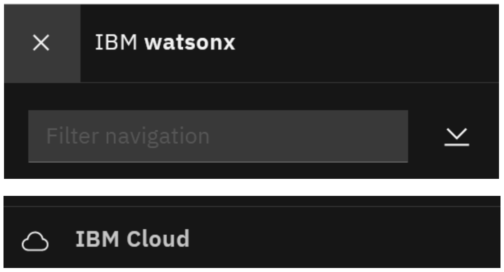
Name
LLM-Workshop-YI

Description
What's the purpose of this project?

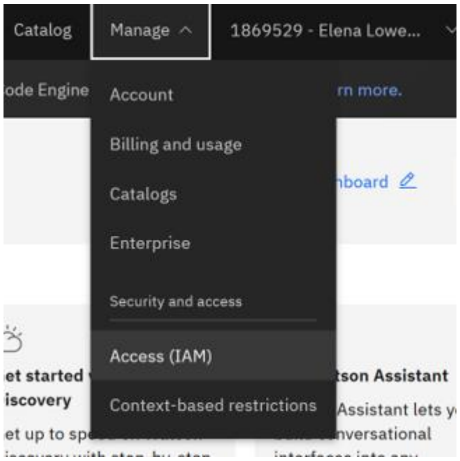
Project ID
`e9207503-90c4-430b-a366-d6c24b5a020a`

Now we will get the authentication token.

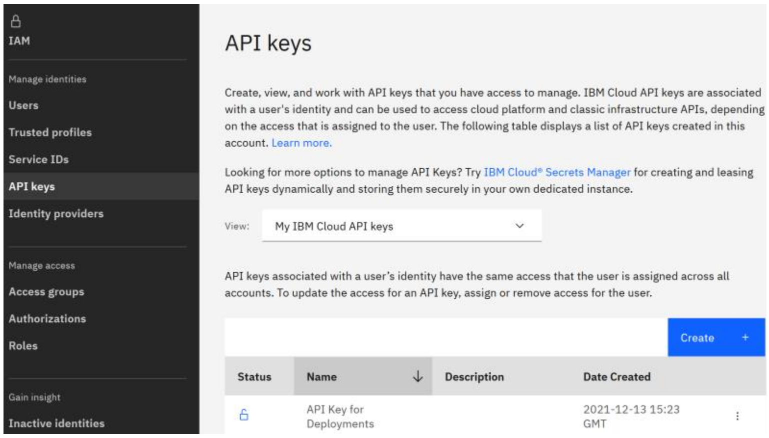
3. Open a new browser window and from the main **watsonx menu** (top right corner), select **IBM Cloud**.



4. Select **Manage -> Access (IAM).**



5. Click **API Keys -> Create**. Give the token a name and save it in a notepad. You will use it in the sample notebook.



API keys

Create, view, and work with API keys that you have access to manage. IBM Cloud API keys are associated with a user's identity and can be used to access cloud platform and classic infrastructure APIs, depending on the access that is assigned to the user. The following table displays a list of API keys created in this account. Learn more.

Looking for more options to manage API Keys? Try IBM Cloud® Secrets Manager for creating and leasing API keys dynamically and storing them securely in your own dedicated instance.

View: My IBM Cloud API keys

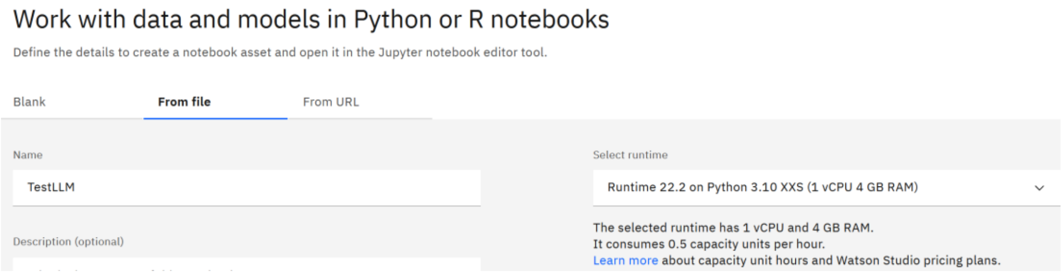
API keys associated with a user's identity have the same access that the user is assigned across all accounts. To update the access for an API key, assign or remove access for the user.

Status	Name	Description	Date Created
	API Key for Deployments		2021-12-13 15:23 GMT

6. In **watsonx.ai** click **New Task -> Work with data and models in Python or R notebooks**.

   Click the **From file** tab and navigate to the downloaded git repo */Notebooks* folder to select the *TestLLM* notebook. Make sure that the *Python 3.10 environment* is selected.

   Click **Create** to import the notebook.



Work with data and models in Python or R notebooks

Define the details to create a notebook asset and open it in the Jupyter notebook editor tool.

Name: TestLLM

Description (optional)

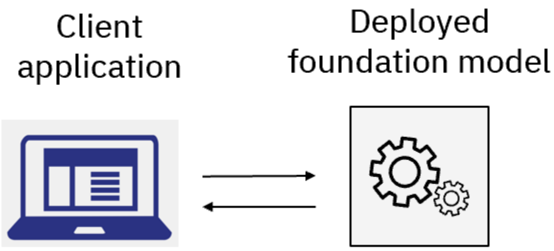
Select runtime: Runtime 22.2 on Python 3.10 XXS (1 vCPU 4 GB RAM)

The selected runtime has 1 vCPU and 4 GB RAM.
It consumes 0.5 capacity units per hour.
Learn more about capacity unit hours and Watson Studio pricing plans.

Let's review the sample notebook.

This notebook acts as a *client application* that invokes the deployed LLM with a Python SDK. We are using the notebook as a client for simplicity of testing during this lab.

Enterprise client applications can be implemented in Python, Java, .Net and many other programming languages. As discussed earlier, LLMs deployed in **watsonx.**ai can be invoked either with REST calls or with the Python SDK.



Run the notebook to test the LLM with your prompts. See specific instructions in the notebook.

Next, we will use a Python IDE, such as **Visual Studio** or **PyCharm** to run the client application.

24. Find the following Python scripts in the downloaded git repo */Applications* folder:

    - *demo_wml_api.py*
    - *demo_wml_api_with_streamlit.py*

    Load these scripts into your Python IDE.

25. In your Python IDE install the *ibm-watson-machine-learning* library. We recommend that you use Python 3.10 environment.

```
pip install ibm-watson-machine-learning
pip install ibm-cloud-sdk-core
```

In non-Anaconda Python environments you may need to install another package:

```
pip install python-dotenv
```

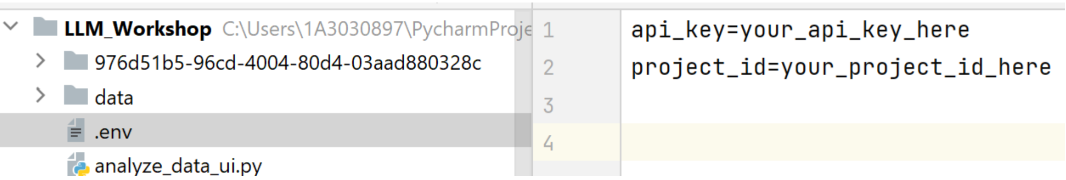
Let’s review the scripts.

demo_wml_api.py is a simple Python script that shows to how invoke an LLM that’s deployed in watsonx.ai. Code in this script can be converted to a module and used in applications that interact with LLMs.

The script has the following functions:

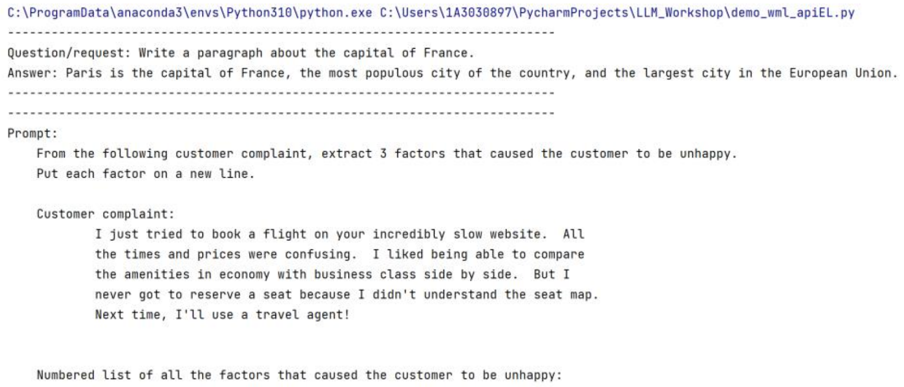
- *get_credentials(): reads the api key and the project id from the .env file (will be used for authentication*
- *get_model()*: creates an LLM model object with the specified parameters
- *answer_questions()*: invokes a model that answers simple questions
- *get_list of_complaints():* generates a list of complaints from a hardcoded customer review
- *invoke_with_REST():* shows how to invoke the LLM using the REST API (other functions use the SDK)
- *get_auth_token():* generates the token that’s required for REST invocation
- *demo_LLM_invocation():* invokes all other functions for testing

Prior to running the script, create a .env file in the root directory of your project and add your *Cloud API key* and *project id*.


LLM_Workshop C:\Users\1A3030897\PycharmProje
976d51b5-96cd-4004-80d4-03aad880328c
data
.env
analyze_data_ui.py

```
api_key=your_api_key_here
project_id=your_project_id_here
```

26. Run the script. The output will be shown in Python terminal.



```
C:\ProgramData\anaconda3\envs\Python310\python.exe C:\Users\1A3030897\PycharmProjects\LLM_Workshop\demo_wml_apiEL.py
---------------------------------------------------------------------------
Question/request: Write a paragraph about the capital of France.
Answer: Paris is the capital of France, the most populous city of the country, and the largest city in the European Union.
---------------------------------------------------------------------------
---------------------------------------------------------------------------
Prompt:
    From the following customer complaint, extract 3 factors that caused the customer to be unhappy.
    Put each factor on a new line.

    Customer complaint:
            I just tried to book a flight on your incredibly slow website.  All
            the times and prices were confusing.  I liked being able to compare
            the amenities in economy with business class side by side.  But I
            never got to reserve a seat because I didn't understand the seat map.
            Next time, I'll use a travel agent!


    Numbered list of all the factors that caused the customer to be unhappy:
```

Let your instructor know if you have questions.

Next, we will invoke the LLM from a UI. We will use a popular web application development framework, *Streamlit*, to create a simple UI.

You can find the video of this application in the *git repo/Reference* folder.

27. To run this script, you will need to install the *streamlit* package in your Python environment.

    ```
    pip install streamlit
    ```

    Important*: If you're running on Windows, you will need to run this script in an Anaconda Python environment because it's the only supported Python environment on Windows. Both VS Code and Pycharm can be configured to use Anaconda. Please check with your instructor if you need help with setup instructions. If you're not able to install Anaconda in your environment, please check with your instructor.*

28. Open the *demo_wml_api_with_streamlit.py script.* This application uses similar code to invoke the LLM as a previous example.

    The application has 4 functions:
    - *get_credentials(): reads the api key and the project id from the .env file (will be used for authentication*
    - *get_model()*: creates an LLM model object with the specified parameters
    - *get_prompt()*: creates a model prompt
    - *answer_questions ():* sets the parameters and invokes the other two functions.

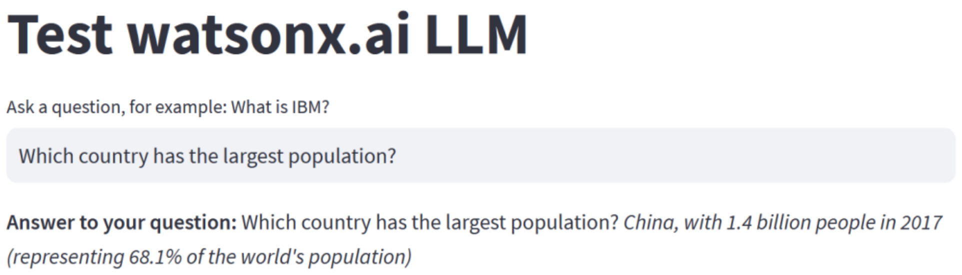
As you can tell by the name of the last function, this is a simple *Question and Answer UI*. You will notice that the prompt is more complicated than the prompt in the previous example: we provide instructions and a few examples (few-shot prompting).

Notice that we are hardcoding the instruction to answer the question. This is just an example, and you can choose to parameterize all components of the prompt.

29. When you run the script, Python will open the *Streamlit UI* in your browser.

If you invoke Python application from a terminal, and not an IDE then use the following command: `streamlit run demo_wml_api_with_streamlit.py`

Note: When testing, ask "general knowledge" questions keeping in mind that our prompt is not sophisticated and that the model was trained on generally available data.

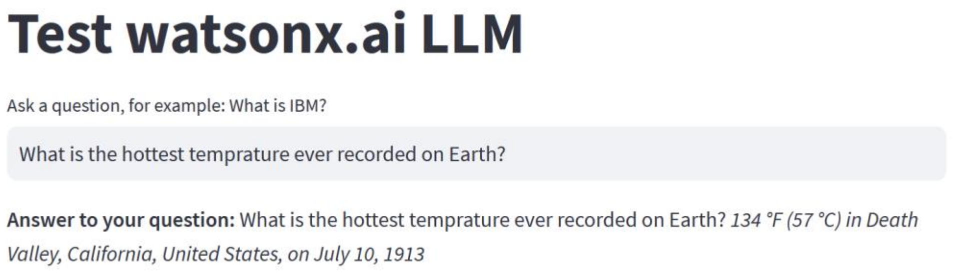


Test watsonx.ai LLM

Ask a question, for example: What is IBM?

Which country has the largest population?

Answer to your question: Which country has the largest population? *China, with 1.4 billion people in 2017 (representing 68.1% of the world's population)*

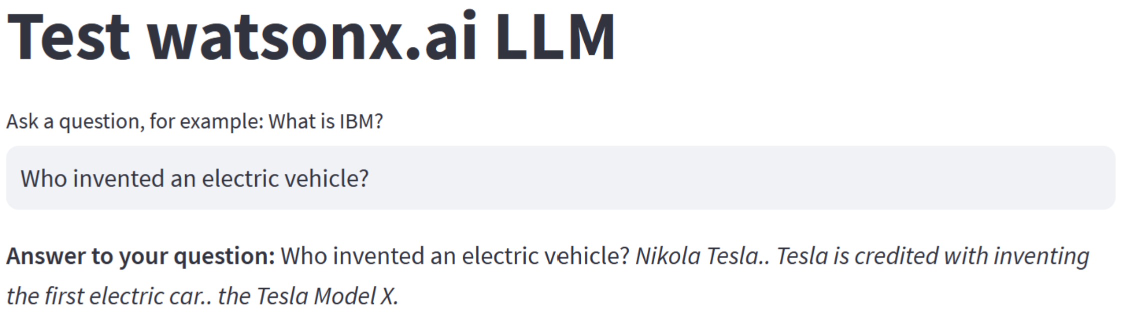


Test watsonx.ai LLM

Ask a question, for example: What is IBM?

What is the hottest temprature ever recorded on Earth?

Answer to your question: What is the hottest temprature ever recorded on Earth? *134 °F (57 °C) in Death Valley, California, United States, on July 10, 1913*



Test watsonx.ai LLM

Ask a question, for example: What is IBM?

Who invented an electric vehicle?

Answer to your question: Who invented an electric vehicle? *Nikola Tesla.. Tesla is credited with inventing the first electric car.. the Tesla Model X.*

Conclusion

You have finished the **Introduction to Generative AI** lab. In this lab you learned:

- Key features of LLMs
- The basics of prompt engineering, including parameter tuning
- Using the **Prompt Lab** to create and test prompts with models available in watsonx.ai
- Testing LLM model inference
- Creating a simple UI to let users interact with LLMs.

What we did not cover in this lab:

- Using LLMs to answer questions on data specific to the enterprise
- Executing multiple instructions with LLMs
- Implementing session management in LLM applications.

If you have questions about these topics, please contact your instructor.