1309 words | 4 minutes

Environment Configuration

This chapter mainly provides some necessary environment configuration guides, including code environment configuration, Python environment configuration of VSCODE code editor, and some other resource configurations used.

1. Code Environment Configuration Guide

Here we introduce each step of code environment configuration in detail, which is divided into two parts: basic environment configuration and general environment configuration to meet the needs of different users and environments.

- **Basic environment configuration** section: Suitable for **beginners** of environment configuration or **new server environments (such as Alibaba Cloud)** . This section introduces how to generate an SSH key and add it to GitHub, as well as how to install and initialize the conda environment.

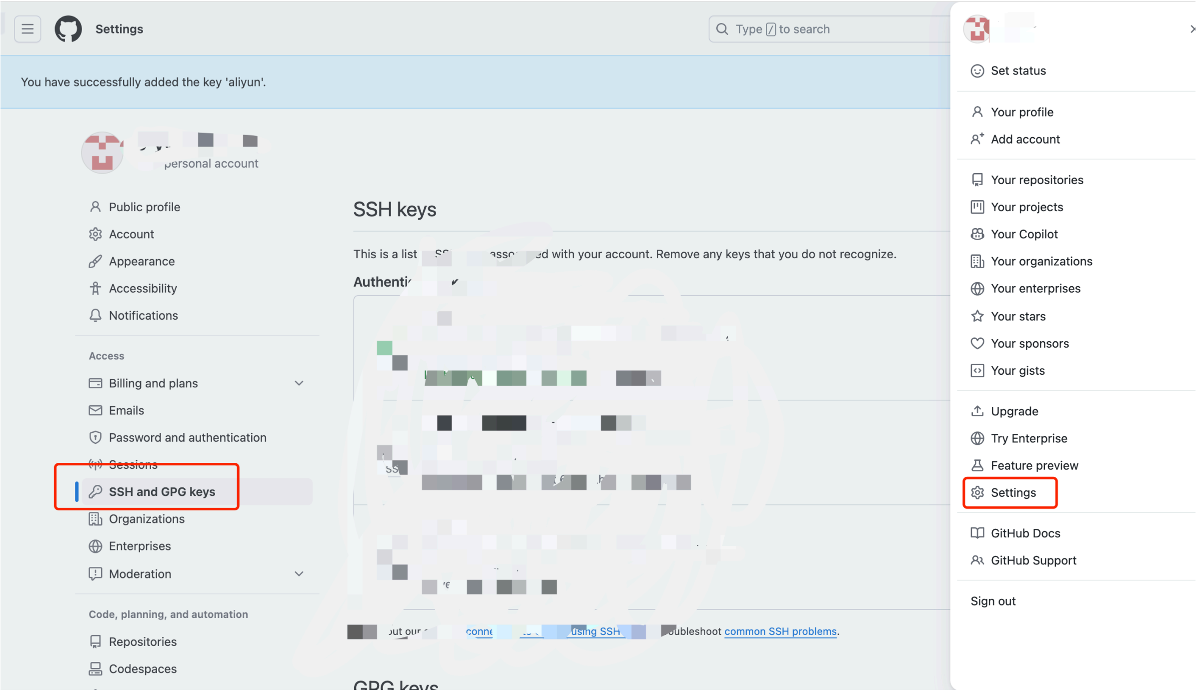
- **General environment configuration** section: suitable for **users with some experience** , local installations with **existing environment foundations , or completely independent environments (such as GitHub Codespace)** . This section describes how to create and activate a conda virtual environment, clone a project repository, switch to the project directory, and install the required Python packages. To speed up the installation of Python packages, we also provide some domestic mirror sources. *For completely independent environments, you can skip the first two steps about virtual environment (conda) configuration* .

1.1 Basic environment configuration (configure git and conda)

1. Generate SSH key `ssh-keygen -t rsa -C "youremail@example.com"`

2. Add the public key to GitHub `cat ~/.ssh/id_rsa.pub` Copy the output content, open GitHub, click the avatar in the upper right corner, select `settings` -> `SSH and GPG keys` -

> `New SSH key` , paste the copied content into key, and click `Add SSH key` .



You have successfully added the key 'aliyun'.

[redacted] personal account

SSH keys

This is a list [redacted] associated with your account. Remove any keys that you do not recognize.

Authenti [redacted]

[redacted] troubleshoot common SSH problems.

GPG keys

3. Install conda environment

    1. Linux environment (usually using Linux environment)

        1. Install:

```shell
mkdir -p ~/miniconda3
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux
bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
rm -rf ~/miniconda3/miniconda.sh
```

        2. initialization:

```shell
~/miniconda3/bin/conda init bash
~/miniconda3/bin/conda init zsh
```

        3. Create a new terminal and check whether conda is installed successfully `conda --version`

    2. macOS environment

        1. Install

```shell
mkdir -p ~/miniconda3
curl https://repo.anaconda.com/miniconda/Miniconda3-latest-MacOS
bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
rm -rf ~/miniconda3/miniconda.sh
```

    2. initialization:

    ```shell
    ~/miniconda3/bin/conda init bash
    ~/miniconda3/bin/conda init zsh
    ```

    3. Create a new terminal and check whether conda is installed successfully `conda --version`

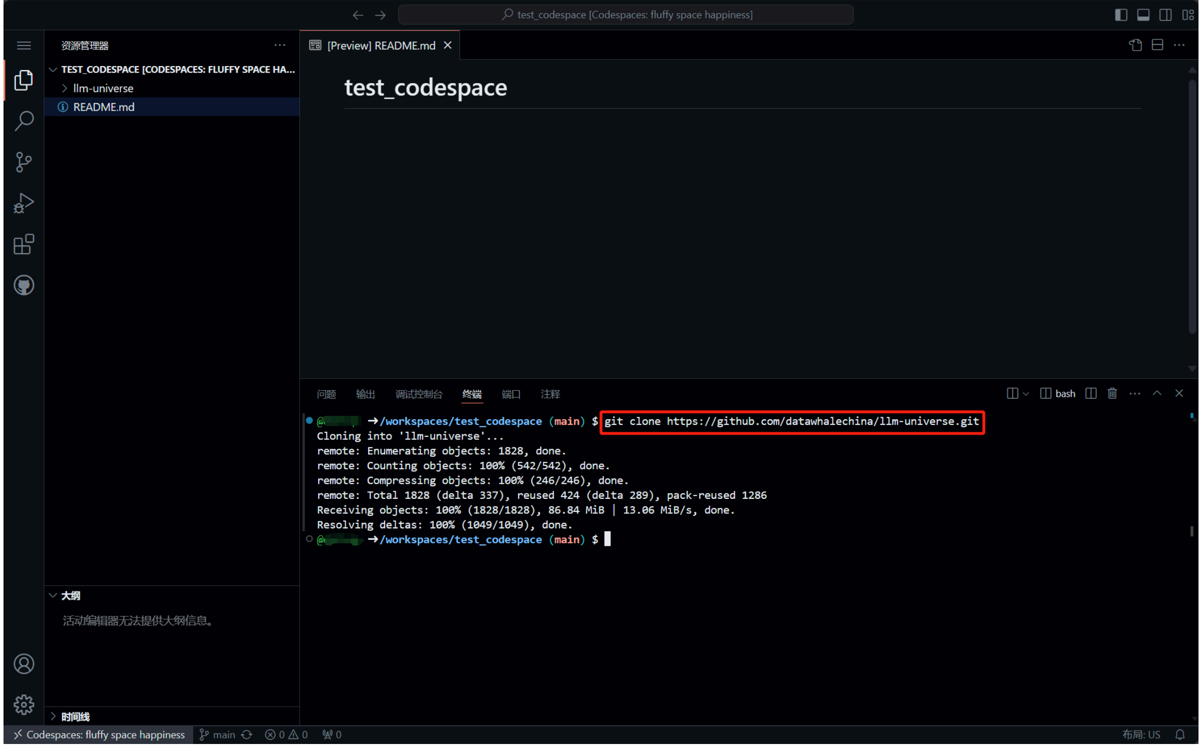
3. Windows environment

    1. download: `curl https://repo.anaconda.com/miniconda/Miniconda3-latest-Windows-x86_64.exe -o miniconda.exe`
    2. Installation: Click on the downloaded file `miniconda.exe` and follow the installation instructions to install it
    3. Open Anaconda Prompt in the menu to check whether conda is installed successfully `conda --version`
    4. Delete the installation package: `del miniconda.exe`

4. Please refer to the following 通用环境配置 section for subsequent configuration

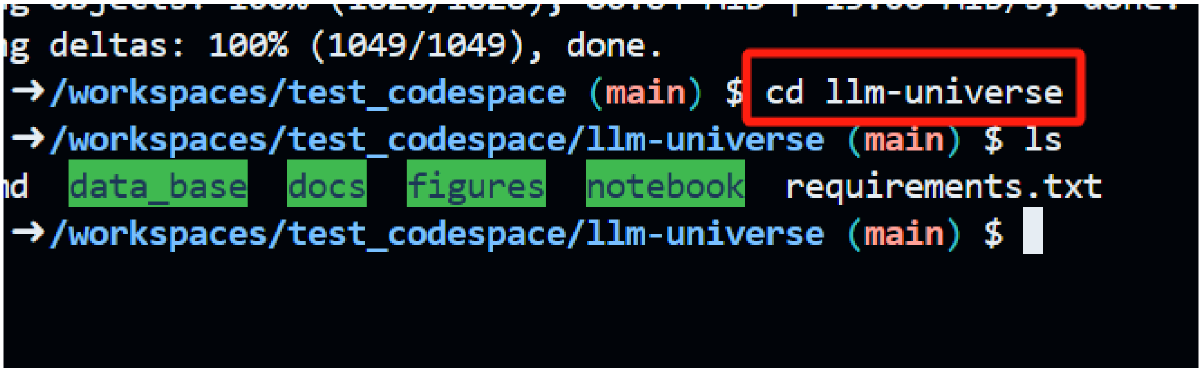
1.2 General environment configuration

1. Create a new virtual environment `conda create -n llm-universe python=3.10`
2. Activate the virtual environment `conda activate llm-universe`
3. Clone the current repository in the path where you want to store the project `git clone git@github.com:datawhalechina/llm-universe.git`


test_codespace [Codespaces: fluffy space happiness]
资源管理器
TEST_CODESPACE [CODESPACES: FLUFFY SPACE HA...
llm-universe
README.md
[Preview] README.md
test_codespace
问题
输出
调试控制台
终端
端口
注释
bash
大纲
活动编辑器无法提供大纲信息。
时间线
Codespaces: fluffy space happiness
main
0
0
0
布局: US

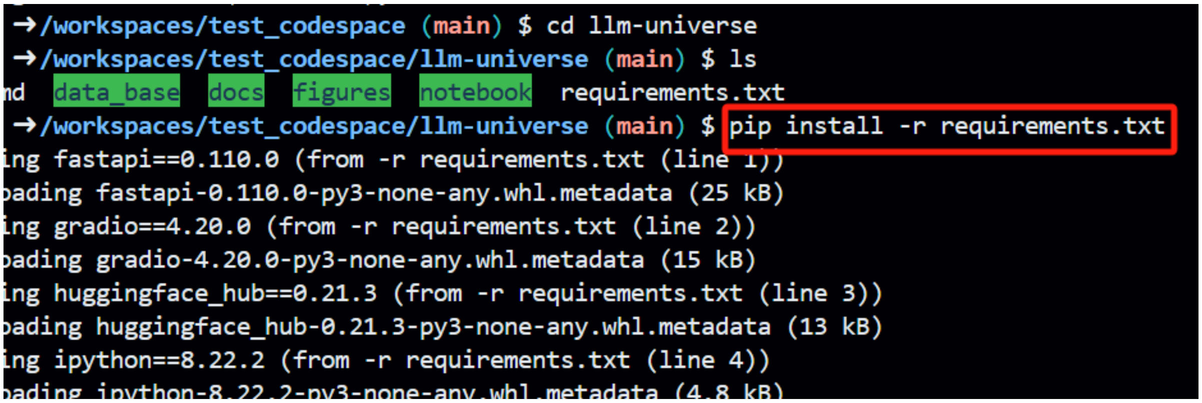
```
@[redacted] →/workspaces/test_codespace (main) $ git clone https://github.com/datawhalechina/llm-universe.git
Cloning into 'llm-universe'...
remote: Enumerating objects: 1828, done.
remote: Counting objects: 100% (542/542), done.
remote: Compressing objects: 100% (246/246), done.
remote: Total 1828 (delta 337), reused 424 (delta 289), pack-reused 1286
Receiving objects: 100% (1828/1828), 86.84 MiB | 13.06 MiB/s, done.
Resolving deltas: 100% (1049/1049), done.
@[redacted] →/workspaces/test_codespace (main) $ 
```

4. Change directory to llm-universe `cd llm-universe`



```
g deltas: 100% (1049/1049), done.
→/workspaces/test_codespace (main) $ cd llm-universe
→/workspaces/test_codespace/llm-universe (main) $ ls
d  data_base  docs  figures  notebook  requirements.txt
→/workspaces/test_codespace/llm-universe (main) $ 
```

5. Install the required packages. `pip install -r requirements.txt`



```
→/workspaces/test_codespace (main) $ cd llm-universe
→/workspaces/test_codespace/llm-universe (main) $ ls
d  data_base  docs  figures  notebook  requirements.txt
→/workspaces/test_codespace/llm-universe (main) $ pip install -r requirements.txt
ing fastapi==0.110.0 (from -r requirements.txt (line 1))
oading fastapi-0.110.0-py3-none-any.whl.metadata (25 kB)
ing gradio==4.20.0 (from -r requirements.txt (line 2))
oading gradio-4.20.0-py3-none-any.whl.metadata (15 kB)
ing huggingface_hub==0.21.3 (from -r requirements.txt (line 3))
oading huggingface_hub-0.21.3-py3-none-any.whl.metadata (13 kB)
ing ipython==8.22.2 (from -r requirements.txt (line 4))
oading ipython-8.22.2-py3-none-any.whl.metadata (4.8 kB)
```

Usually you can speed up the installation through Tsinghua source `pip install -r requirements.txt -i https://pypi.tuna.tsinghua.edu.cn/simple`

> Here is a list of commonly used domestic mirror sources. When the mirror source is not stable, you can switch it as needed: Tsinghua University: https://pypi.tuna.tsinghua.edu.cn/simple/Alibaba Cloud: http://mirrors.aliyun.com/pypi/simple/University of Science and Technology of China: https://pypi.mirrors.ustc.edu.cn/simple/Huazhong University of Science and Technology:

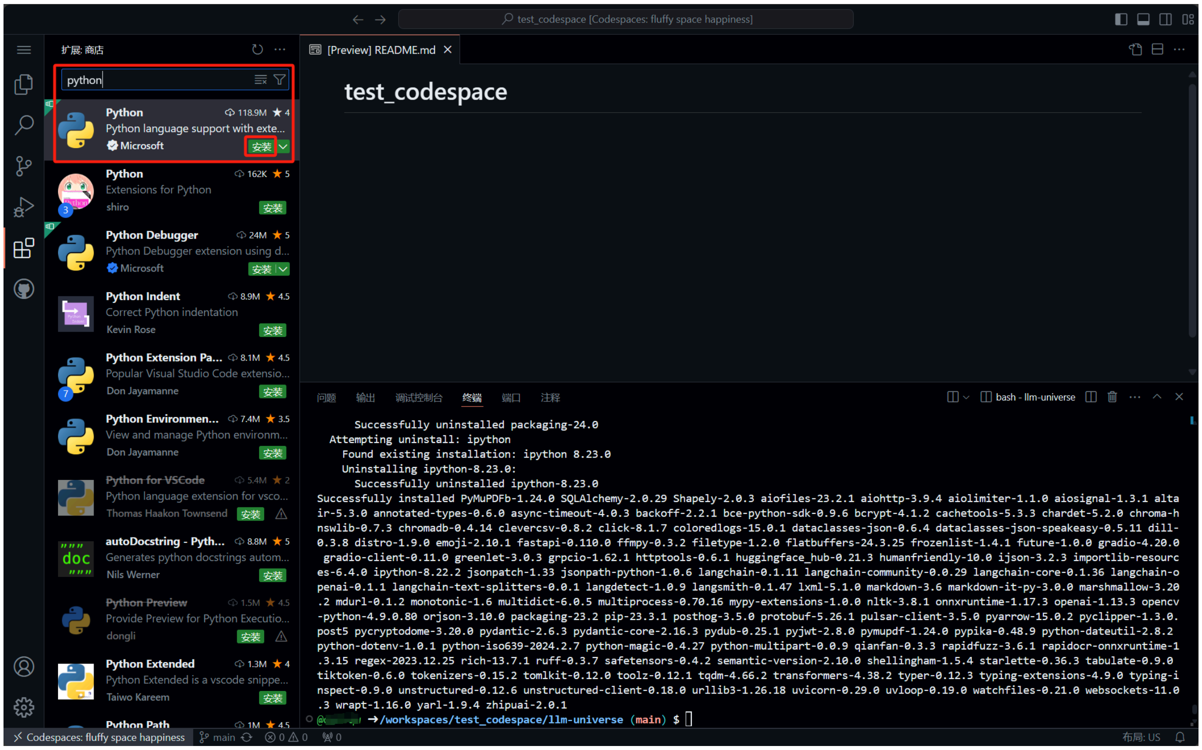
> http://pypi.hustunique.com/simple/Shanghai Jiaotong University:
> https://mirror.sjtu.edu.cn/pypi/web/simple/Douban : http://pypi.douban.com/simple

2. VSCode configures Python environment

1. Installing the Python plugin

   This tutorial is developed based on Python language. For a better development experience, we need to install the Python plug-in.

   Search in the plugin market `Python` , find `Python` the plugin and install it.

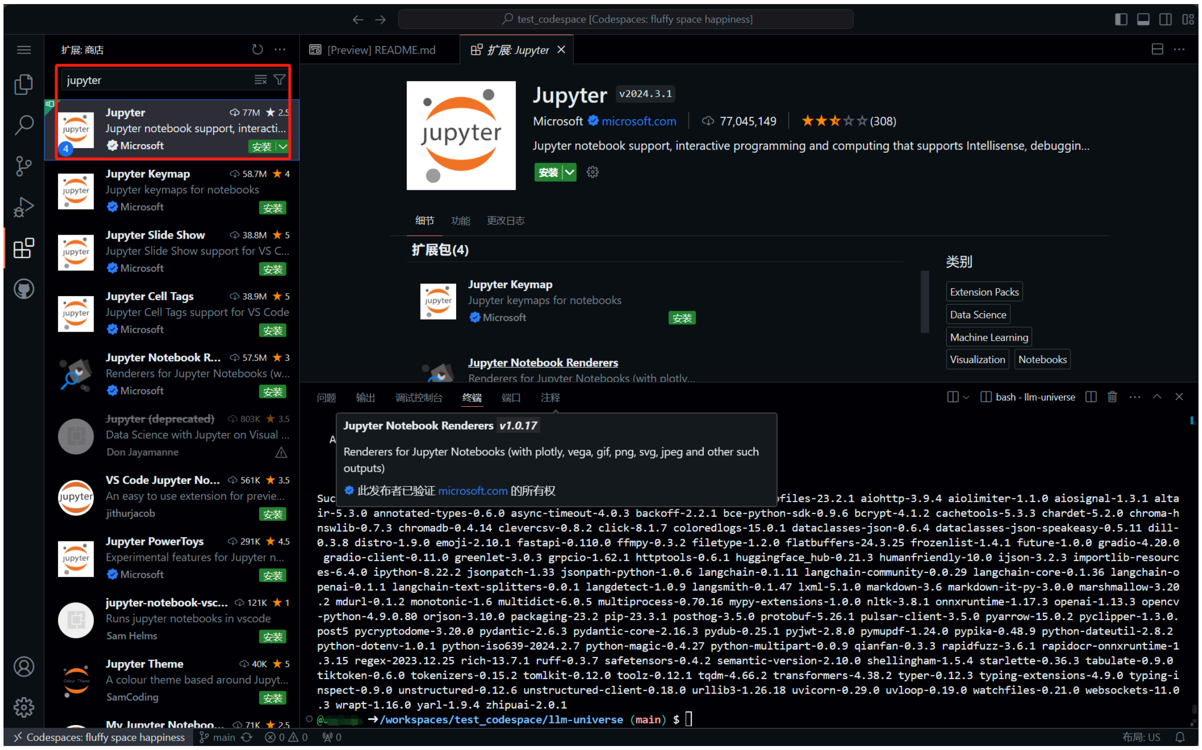
   
   test_codespace [Codespaces: fluffy space happiness]
   扩展: 商店
   python
   Python 118.9M ★4
   Python language support with exte...
   Microsoft 安装
   Python 162K ★5
   Extensions for Python
   shiro 安装
   Python Debugger 24M ★5
   Python Debugger extension using d...
   Microsoft 安装
   Python Indent 8.9M ★4.5
   Correct Python indentation
   Kevin Rose 安装
   Python Extension Pa... 8.1M ★4.5
   Popular Visual Studio Code extensio...
   Don Jayamanne 安装
   Python Environmen... 7.4M ★3.5
   View and manage Python environm...
   Don Jayamanne 安装
   Python for VSCode 5.4M ★2
   Python language extension for vsco...
   Thomas Haakon Townsend 安装
   autoDocstring - Pyth... 8.8M ★5
   Generates python docstrings autom...
   Nils Werner 安装
   Python Preview 1.5M ★4.5
   Provide Preview for Python Executio...
   dongli 安装
   Python Extended 1.3M ★4
   Python Extended is a vscode snippe...
   Taiwo Kareem 安装
   Python Path 1M ★4.5
   [Preview] README.md
   test_codespace
   问题 输出 调试控制台 终端 端口 注释
   bash - llm-universe
   Codespaces: fluffy space happiness
   main
   0 0 0
   布局: US

   ```
         Successfully uninstalled packaging-24.0
     Attempting uninstall: ipython
       Found existing installation: ipython 8.23.0
       Uninstalling ipython-8.23.0:
         Successfully uninstalled ipython-8.23.0
   Successfully installed PyMuPDFb-1.24.0 SQLAlchemy-2.0.29 Shapely-2.0.3 aiofiles-23.2.1 aiohttp-3.9.4 aiolimiter-1.1.0 aiosignal-1.3.1 alta
   ir-5.3.0 annotated-types-0.6.0 async-timeout-4.0.3 backoff-2.2.1 bce-python-sdk-0.9.6 bcrypt-4.1.2 cachetools-5.3.3 chardet-5.2.0 chroma-h
   nswlib-0.7.3 chromadb-0.4.14 clevercsv-0.8.2 click-8.1.7 coloredlogs-15.0.1 dataclasses-json-0.6.4 dataclasses-json-speakeasy-0.5.11 dill-
   0.3.8 distro-1.9.0 emoji-2.10.1 fastapi-0.110.0 ffmpy-0.3.2 filetype-1.2.0 flatbuffers-24.3.25 frozenlist-1.4.1 future-1.0.0 gradio-4.20.0
    gradio-client-0.11.0 greenlet-3.0.3 grpcio-1.62.1 httptools-0.6.1 huggingface_hub-0.21.3 humanfriendly-10.0 ijson-3.2.3 importlib-resourc
   es-6.4.0 ipython-8.22.2 jsonpatch-1.33 jsonpath-python-1.0.6 langchain-0.1.11 langchain-community-0.0.29 langchain-core-0.1.36 langchain-o
   penai-0.1.1 langchain-text-splitters-0.0.1 langdetect-1.0.9 langsmith-0.1.47 lxml-5.1.0 markdown-3.6 markdown-it-py-3.0.0 marshmallow-3.20
   .2 mdurl-0.1.2 monotonic-1.6 multidict-6.0.5 multiprocess-0.70.16 mypy-extensions-1.0.0 nltk-3.8.1 onnxruntime-1.17.3 openai-1.13.3 opencv
   -python-4.9.0.80 orjson-3.10.0 packaging-23.2 pip-23.3.1 posthog-3.5.0 protobuf-5.26.1 pulsar-client-3.5.0 pyarrow-15.0.2 pyclipper-1.3.0.
   post5 pycryptodome-3.20.0 pydantic-2.6.3 pydantic-core-2.16.3 pydub-0.25.1 pyjwt-2.8.0 pymupdf-1.24.0 pypika-0.48.9 python-dateutil-2.8.2
   python-dotenv-1.0.1 python-iso639-2024.2.7 python-magic-0.4.27 python-multipart-0.0.9 qianfan-0.3.3 rapidfuzz-3.6.1 rapidocr-onnxruntime-1
   .3.15 regex-2023.12.25 rich-13.7.1 ruff-0.3.7 safetensors-0.4.2 semantic-version-2.10.0 shellingham-1.5.4 starlette-0.36.3 tabulate-0.9.0
   tiktoken-0.6.0 tokenizers-0.15.2 tomlkit-0.12.0 toolz-0.12.1 tqdm-4.66.2 transformers-4.38.2 typer-0.12.3 typing-extensions-4.9.0 typing-i
   nspect-0.9.0 unstructured-0.12.6 unstructured-client-0.18.0 urllib3-1.26.18 uvicorn-0.29.0 uvloop-0.19.0 watchfiles-0.21.0 websockets-11.0
   .3 wrapt-1.16.0 yarl-1.9.4 zhipuai-2.0.1
   @[redacted] →/workspaces/test_codespace/llm-universe (main) $
   ```

   When we execute Python code, it will automatically recognize our Python environment and provide functions such as code completion to facilitate our development.

2. Install Jupyter plugin In this tutorial, we use Jupyter Notebook for development, so we need to install the Jupyter plugin. Search in the plugin market `Jupyter` , find `Jupyter` the

plugin and install it.



扩展: 商店

jupyter

Jupyter — 77M ★ 2.5 — Jupyter notebook support, interacti... — Microsoft — 安装
Jupyter Keymap — 58.7M ★ 4 — Jupyter keymaps for notebooks — Microsoft — 安装
Jupyter Slide Show — 38.8M ★ 5 — Jupyter Slide Show support for VS C... — Microsoft — 安装
Jupyter Cell Tags — 38.9M ★ 5 — Jupyter Cell Tags support for VS Code — Microsoft — 安装
Jupyter Notebook R... — 57.5M ★ 3 — Renderers for Jupyter Notebooks (w... — Microsoft — 安装
~~Jupyter (deprecated)~~ — 803K ★ 3.5 — Data Science with Jupyter on Visual ... — Don Jayamanne
VS Code Jupyter No... — 561K ★ 3.5 — An easy to use extension for previe... — jithurjacob — 安装
Jupyter PowerToys — 291K ★ 4.5 — Experimental features for Jupyter n... — Microsoft — 安装
jupyter-notebook-vsc... — 121K ★ 1 — Runs jupyter notebooks in vscode — Sam Helms — 安装
Jupyter Theme — 40K ★ 5 — A colour theme based around Jupyt... — SamCoding — 安装
My Jupyter Noteboo... — 71K ★ 2.5

Jupyter v2024.3.1

Microsoft microsoft.com | 77,045,149 | ★★★☆☆ (308)

Jupyter notebook support, interactive programming and computing that supports Intellisense, debuggin...

安装

细节 功能 更改日志

扩展包(4)

Jupyter Keymap — Jupyter keymaps for notebooks — Microsoft — 安装

Jupyter Notebook Renderers — Renderers for Jupyter Notebooks (with plotly...

类别: Extension Packs, Data Science, Machine Learning, Visualization, Notebooks

问题 输出 调试控制台 终端 端口 注释 — bash - llm-universe

Jupyter Notebook Renderers *v1.0.17*

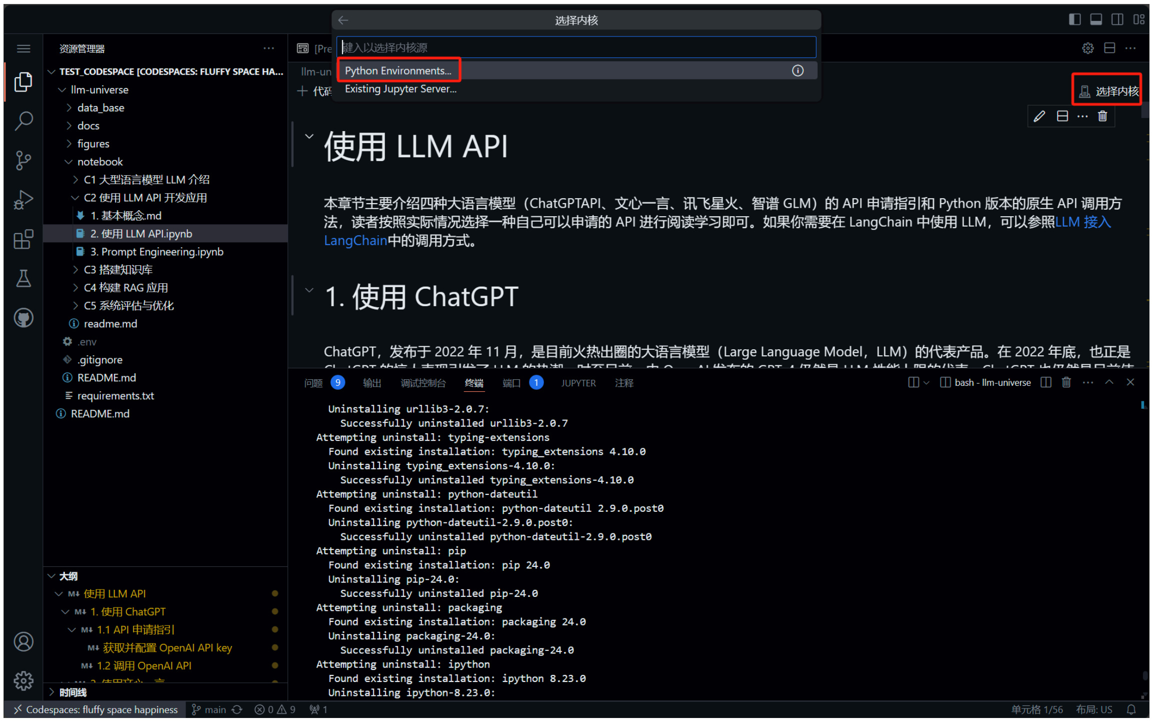
Renderers for Jupyter Notebooks (with plotly, vega, gif, png, svg, jpeg and other such outputs)

此发布者已验证 microsoft.com 的所有权

```
Suc                                                  ofiles-23.2.1 aiohttp-3.9.4 aiolimiter-1.1.0 aiosignal-1.3.1 alta
ir-5.3.0 annotated-types-0.6.0 async-timeout-4.0.3 backoff-2.2.1 bce-python-sdk-0.9.6 bcrypt-4.1.2 cachetools-5.3.3 chardet-5.2.0 chroma-h
nswlib-0.7.3 chromadb-0.4.14 clevercsv-0.8.2 click-8.1.7 coloredlogs-15.0.1 dataclasses-json-0.6.4 dataclasses-json-speakeasy-0.5.11 dill-
0.3.8 distro-1.9.0 emoji-2.10.1 fastapi-0.110.0 ffmpy-0.3.2 filetype-1.2.0 flatbuffers-24.3.25 frozenlist-1.4.1 future-1.0.0 gradio-4.20.0
 gradio-client-0.11.0 greenlet-3.0.3 grpcio-1.62.1 httptools-0.6.1 huggingface_hub-0.21.3 humanfriendly-10.0 ijson-3.2.3 importlib-resourc
es-6.4.0 ipython-8.22.2 jsonpatch-1.33 jsonpath-python-1.0.6 langchain-0.1.11 langchain-community-0.0.29 langchain-core-0.1.36 langchain-o
penai-0.1.1 langchain-text-splitters-0.0.1 langdetect-1.0.9 langsmith-0.1.47 lxml-5.1.0 markdown-3.6 markdown-it-py-3.0.0 marshmallow-3.20
.2 mdurl-0.1.2 monotonic-1.6 multidict-6.0.5 multiprocess-0.70.16 mypy-extensions-1.0.0 nltk-3.8.1 onnxruntime-1.17.3 openai-1.13.3 opencv
-python-4.9.0.80 orjson-3.10.0 packaging-23.2 pip-23.3.1 posthog-3.5.0 protobuf-5.26.1 pulsar-client-3.5.0 pyarrow-15.0.2 pyclipper-1.3.0.
post5 pycryptodome-3.20.0 pydantic-2.6.3 pydantic-core-2.16.3 pydub-0.25.1 pyjwt-2.8.0 pymupdf-1.24.0 pypika-0.48.9 python-dateutil-2.8.2
python-dotenv-1.0.1 python-iso639-2024.2.7 python-magic-0.4.27 python-multipart-0.0.9 qianfan-0.3.3 rapidfuzz-3.6.1 rapidocr-onnxruntime-1
.3.15 regex-2023.12.25 rich-13.7.1 ruff-0.3.7 safetensors-0.4.2 semantic-version-2.10.0 shellingham-1.5.4 starlette-0.36.3 tabulate-0.9.0
tiktoken-0.6.0 tokenizers-0.15.2 tomlkit-0.12.0 toolz-0.12.1 tqdm-4.66.2 transformers-4.38.2 typer-0.12.3 typing-extensions-4.9.0 typing-i
nspect-0.9.0 unstructured-0.12.6 unstructured-client-0.18.0 urllib3-1.26.18 uvicorn-0.29.0 uvloop-0.19.0 watchfiles-0.21.0 websockets-11.0
.3 wrapt-1.16.0 yarl-1.9.4 zhipuai-2.0.1
@[redacted] ➜ /workspaces/test_codespace/llm-universe (main) $ 
```

3. Configuring the Python environment for Jupyter Notebook

    1. Open a Jupyter Notebook
    2. Click in the upper right corner `选择 Python 解释器（显示内容会根据选择环境的名称变化）` to select the Python environment for the current Jupyter Notebook.



选择内核

键入以选择内核源

Python Environments...

Existing Jupyter Server...

选择内核

资源管理器

TEST_CODESPACE [CODESPACES: FLUFFY SPACE HA...
llm-universe
data_base
docs
figures
notebook
C1 大型语言模型 LLM 介绍
C2 使用 LLM API 开发应用
1. 基本概念.md
2. 使用 LLM API.ipynb
3. Prompt Engineering.ipynb
C3 搭建知识库
C4 构建 RAG 应用
C5 系统评估与优化
readme.md
.env
.gitignore
README.md
requirements.txt
README.md

大纲
使用 LLM API
1. 使用 ChatGPT
1.1 API 申请指引
获取并配置 OpenAI API key
1.2 调用 OpenAI API
2. 使用文心一言

时间线

使用 LLM API

本章节主要介绍四种大语言模型（ChatGPTAPI、文心一言、讯飞星火、智谱 GLM）的 API 申请指引和 Python 版本的原生 API 调用方法，读者按照实际情况选择一种自己可以申请的 API 进行阅读学习即可。如果你需要在 LangChain 中使用 LLM，可以参照LLM 接入 LangChain中的调用方式。

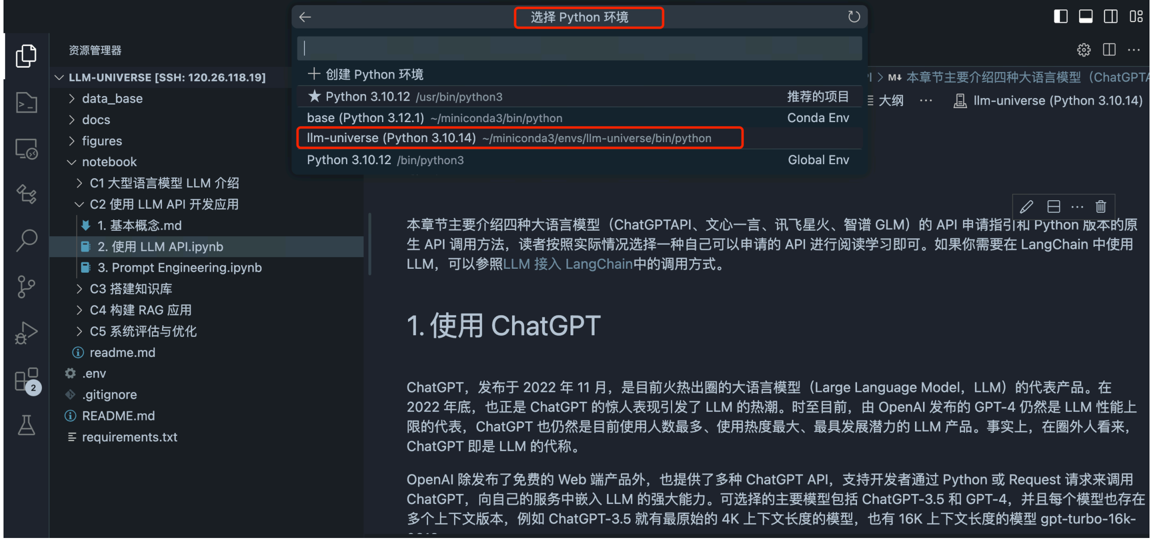
1. 使用 ChatGPT

ChatGPT，发布于 2022 年 11 月，是目前火热出圈的大语言模型（Large Language Model，LLM）的代表产品。在 2022 年底，也正是

问题 9 输出 调试控制台 终端 端口 1 JUPYTER 注释 — bash - llm-universe

```
    Uninstalling urllib3-2.0.7:
      Successfully uninstalled urllib3-2.0.7
  Attempting uninstall: typing-extensions
    Found existing installation: typing_extensions 4.10.0
    Uninstalling typing_extensions-4.10.0:
      Successfully uninstalled typing_extensions-4.10.0
  Attempting uninstall: python-dateutil
    Found existing installation: python-dateutil 2.9.0.post0
    Uninstalling python-dateutil-2.9.0.post0:
      Successfully uninstalled python-dateutil-2.9.0.post0
  Attempting uninstall: pip
    Found existing installation: pip 24.0
    Uninstalling pip-24.0:
      Successfully uninstalled pip-24.0
  Attempting uninstall: packaging
    Found existing installation: packaging 24.0
    Uninstalling packaging-24.0:
      Successfully uninstalled packaging-24.0
  Attempting uninstall: ipython
    Found existing installation: ipython 8.23.0
    Uninstalling ipython-8.23.0:
```

    3. Click `选择 Python` to enter the environment list and select the environment we configured `llm-universe` .


选择 Python 环境
＋ 创建 Python 环境
★ Python 3.10.12 /usr/bin/python3 推荐的项目
base (Python 3.12.1) ~/miniconda3/bin/python Conda Env
llm-universe (Python 3.10.14) ~/miniconda3/envs/llm-universe/bin/python
Python 3.10.12 /bin/python3 Global Env
资源管理器
LLM-UNIVERSE [SSH: 120.26.118.19]
data_base
docs
figures
notebook
C1 大型语言模型 LLM 介绍
C2 使用 LLM API 开发应用
1. 基本概念.md
2. 使用 LLM API.ipynb
3. Prompt Engineering.ipynb
C3 搭建知识库
C4 构建 RAG 应用
C5 系统评估与优化
readme.md
.env
.gitignore
README.md
requirements.txt
2
大纲
llm-universe (Python 3.10.14)

本章节主要介绍四种大语言模型（ChatGPTAPI、文心一言、讯飞星火、智谱 GLM）的 API 申请指引和 Python 版本的原生 API 调用方法，读者按照实际情况选择一种自己可以申请的 API 进行阅读学习即可。如果你需要在 LangChain 中使用 LLM，可以参照LLM 接入 LangChain中的调用方式。

1. 使用 ChatGPT

ChatGPT，发布于 2022 年 11 月，是目前火热出圈的大语言模型（Large Language Model，LLM）的代表产品。在 2022 年底，也正是 ChatGPT 的惊人表现引发了 LLM 的热潮。时至目前，由 OpenAI 发布的 GPT-4 仍然是 LLM 性能上限的代表，ChatGPT 也仍然是目前使用人数最多、使用热度最大、最具发展潜力的 LLM 产品。事实上，在圈外人看来，ChatGPT 即是 LLM 的代称。

OpenAI 除发布了免费的 Web 端产品外，也提供了多种 ChatGPT API，支持开发者通过 Python 或 Request 请求来调用 ChatGPT，向自己的服务中嵌入 LLM 的强大能力。可选择的主要模型包括 ChatGPT-3.5 和 GPT-4，并且每个模型也存在多个上下文版本，例如 ChatGPT-3.5 就有最原始的 4K 上下文长度的模型，也有 16K 上下文长度的模型 gpt-turbo-16k-

After that we can use our Python environment for development in Jupyter Notebook.

3. Download other resources

3.1 Download NLTK related resources

When we use the open source word vector model to build open source word vectors, we need to use some resources of the third-party library nltk. Under normal circumstances, it will be automatically downloaded from the Internet, but the download may be interrupted due to network reasons. When we use nltk, an error will be reported. Here we download relevant resources from the domestic warehouse mirror address.

We use the following command to download the nltk resource and decompress it:

shell

```shell
cd /root
git clone https://gitee.com/yzy0612/nltk_data.git  --branch gh-pages
cd nltk_data
mv packages/*  ./
cd tokenizers
unzip punkt.zip
cd ../taggers
unzip averaged_perceptron_tagger.zip
```

‹ Previous chapter

6. Basic use of GitHub Codespaces (optional)