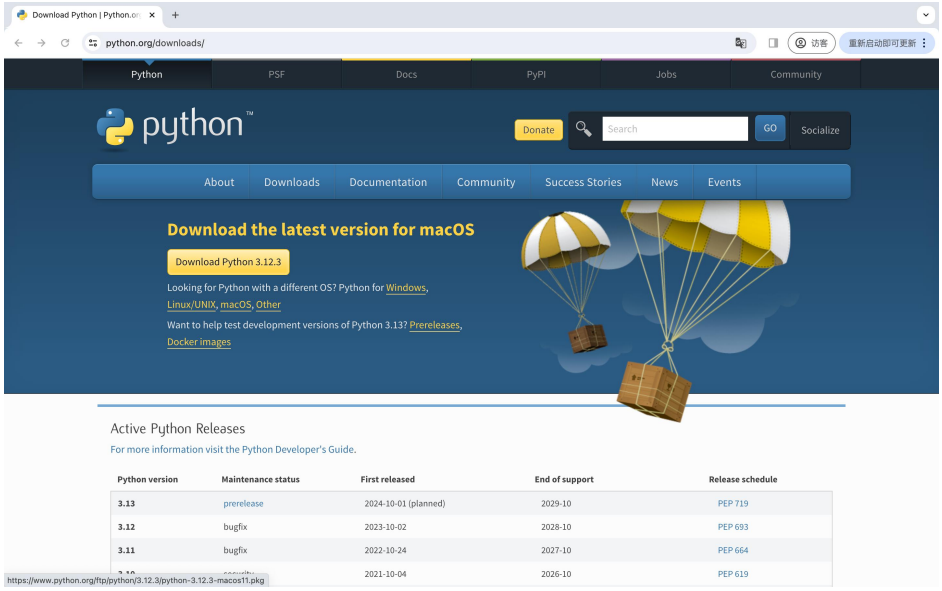
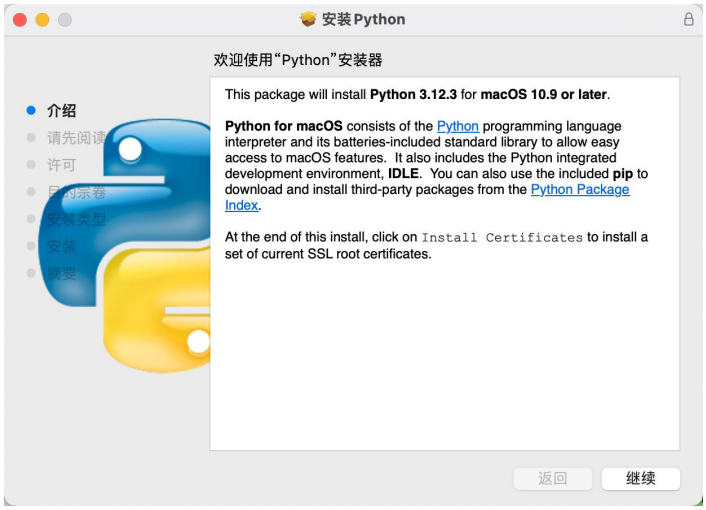
Installation(all base on Macbook M1):

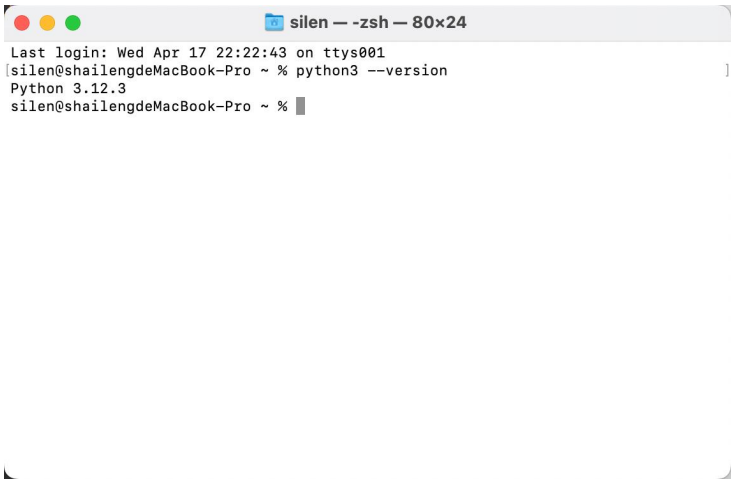
1. python

Download from website then install following the guidelines.



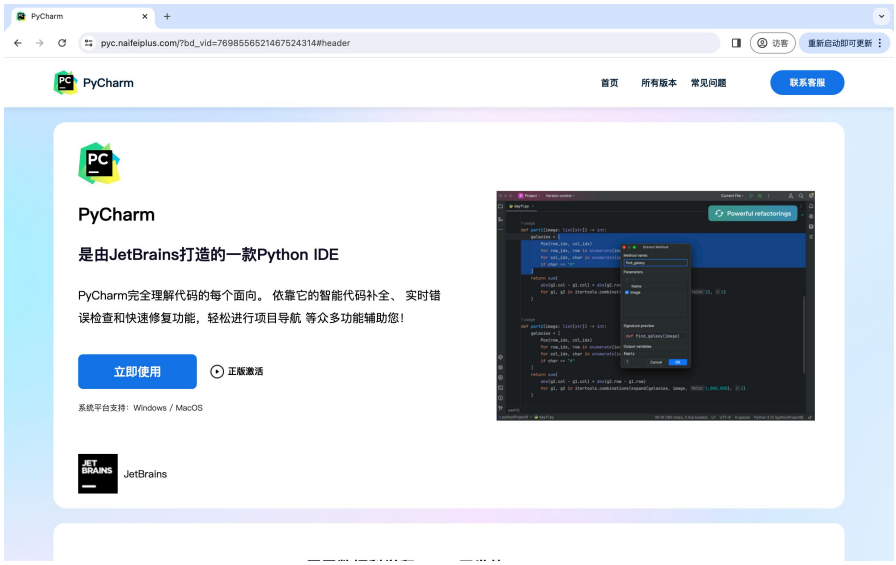


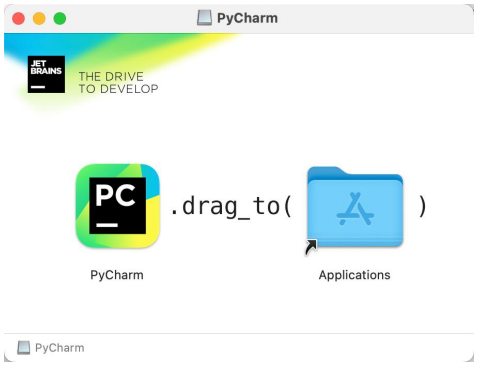
Can check python version by ‘python --version’ after installation in terminal.



2. pycharm

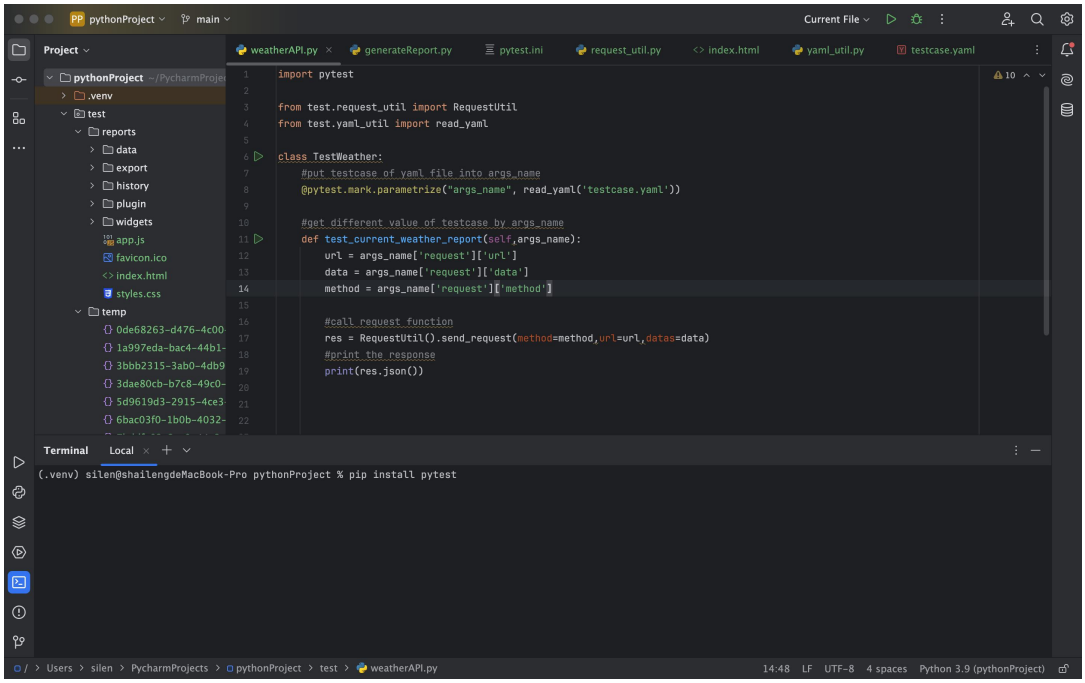
Download from website then install following the guidelines.



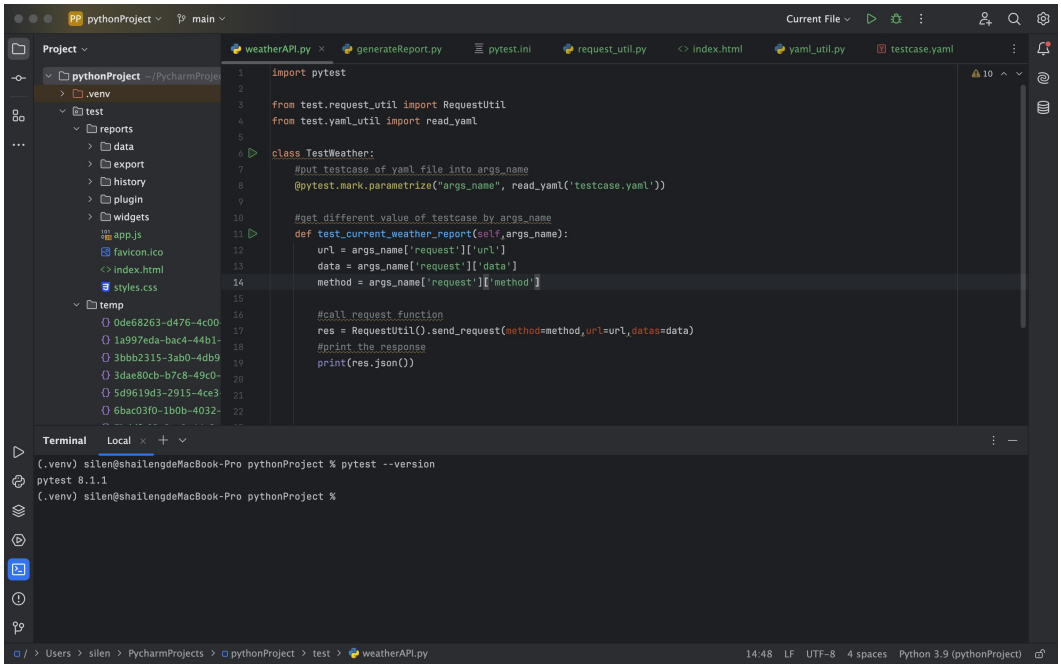


3. pytest

Can install pytest by ‘pip install pytest’ in terminal of pycharm.

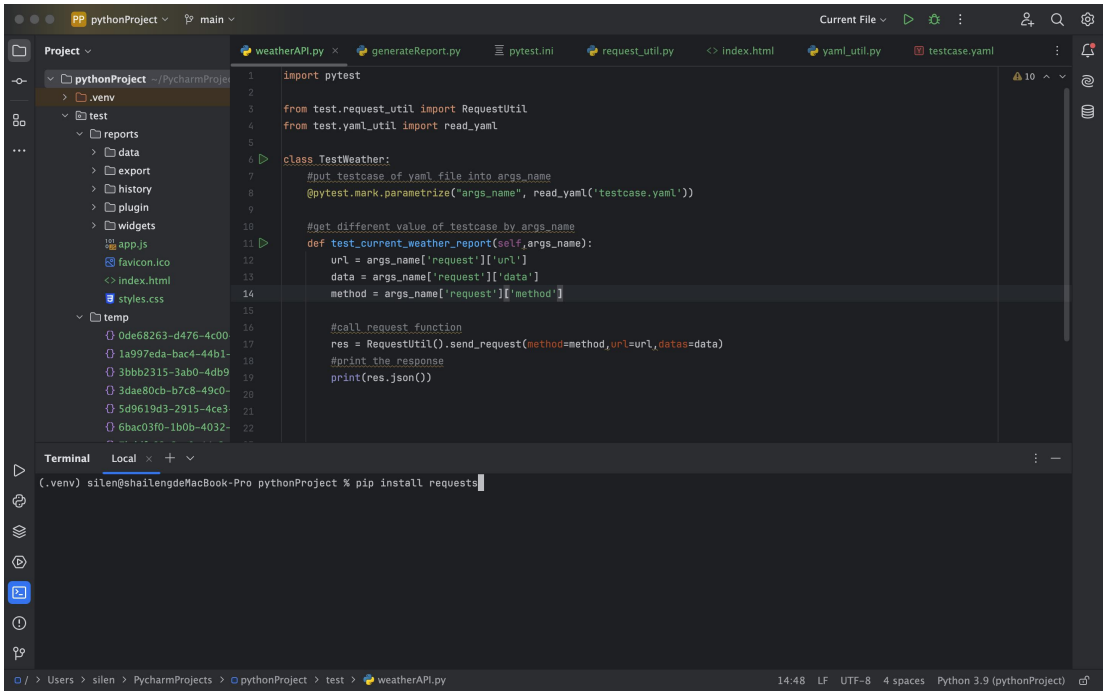


Can check pytest version by ‘pytest --version’ after installation.

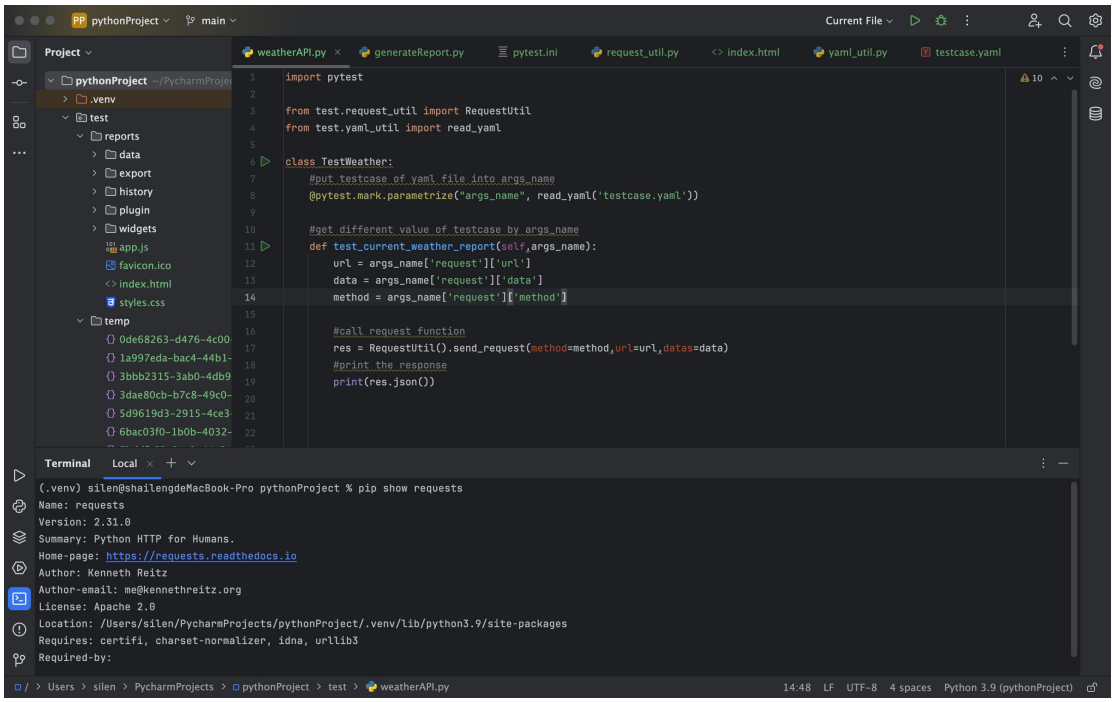


4. requests

Can install requests by ‘pip install requests’ in terminal of pycharm.



Can check requests version by ‘pip show requests’ after installation.



5. homebrew

Can install homebrew by following command in terminal one by one, when see the green info means homebrew installed successfully.

echo export PATH=/opt/homebrew/bin:$PATH >> ~/.zshrc

source ~/.zshrc

HOMEBREW_CORE_GIT_REMOTE=https://mirrors.ustc.edu.cn/homebrew-core.git

/bin/bash -c "$(curl -fsSL https://cdn.jsdelivr.net/gh/ineo6/homebrew-install/install.sh)"

Then can update homebrew by following command in terminal one by one.

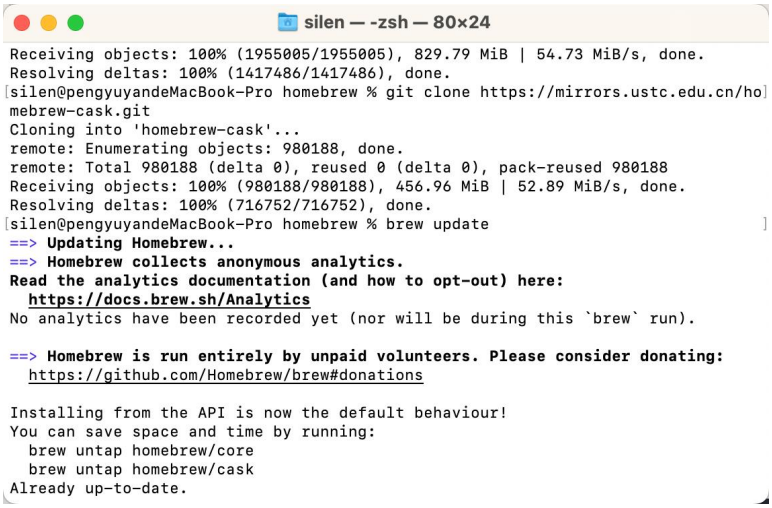
cd "$(brew --repo)"
git clone https://mirrors.ustc.edu.cn/brew.git

cd "$(brew --repo)/Library/Taps/homebrew"

git clone https://mirrors.ustc.edu.cn/homebrew-core.git

git clone https://mirrors.ustc.edu.cn/homebrew-cask.git

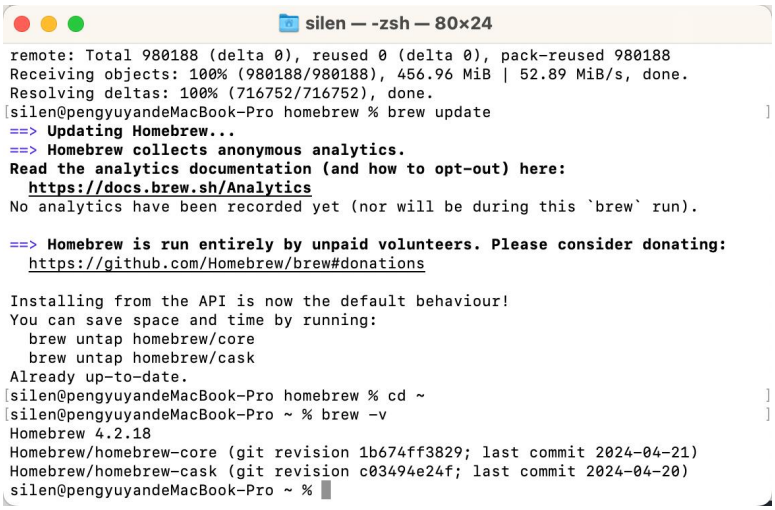
brew update



After updating, check installationg of homebrew by following command in terminal one by one.

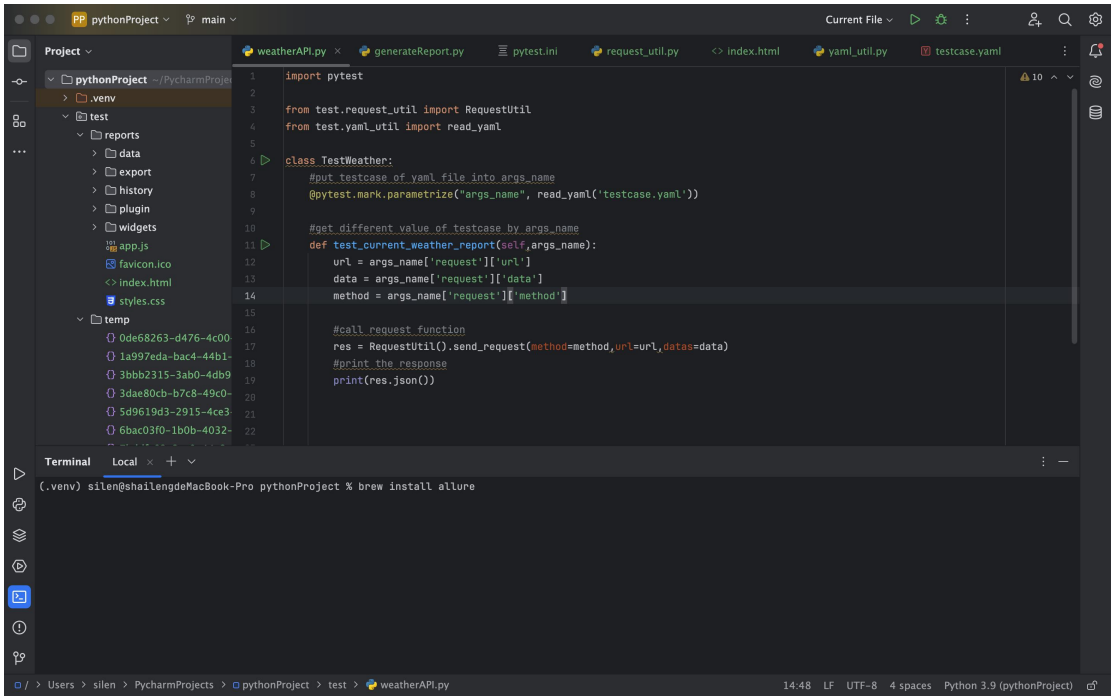
cd ~

brew -v

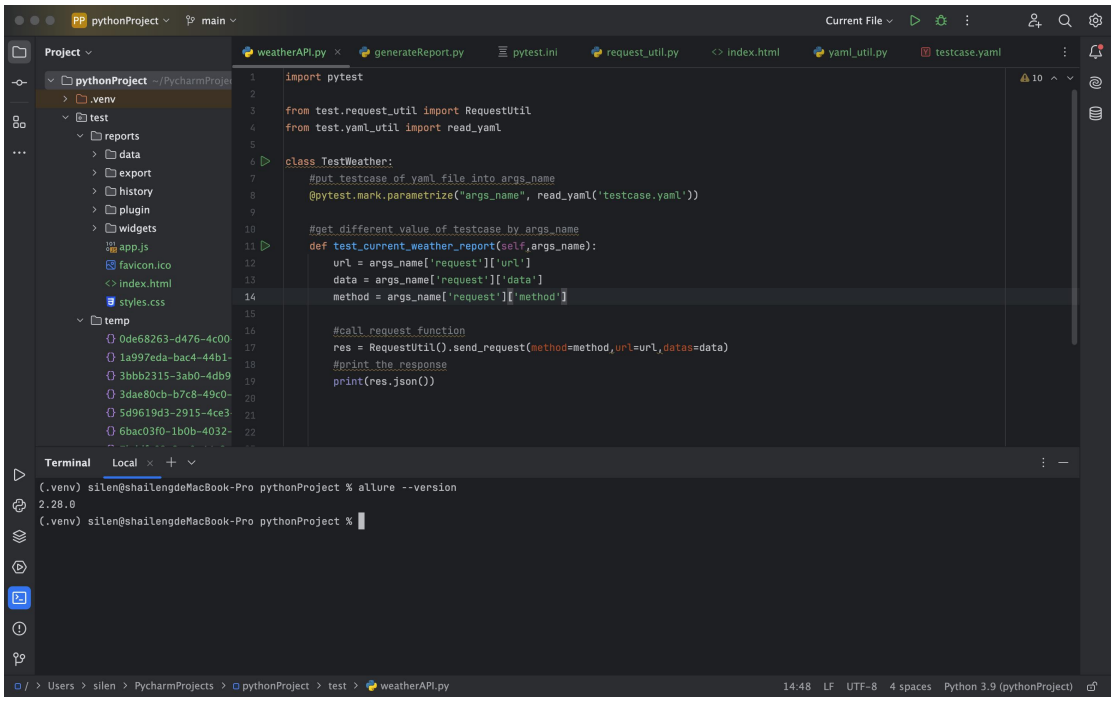


6. allure

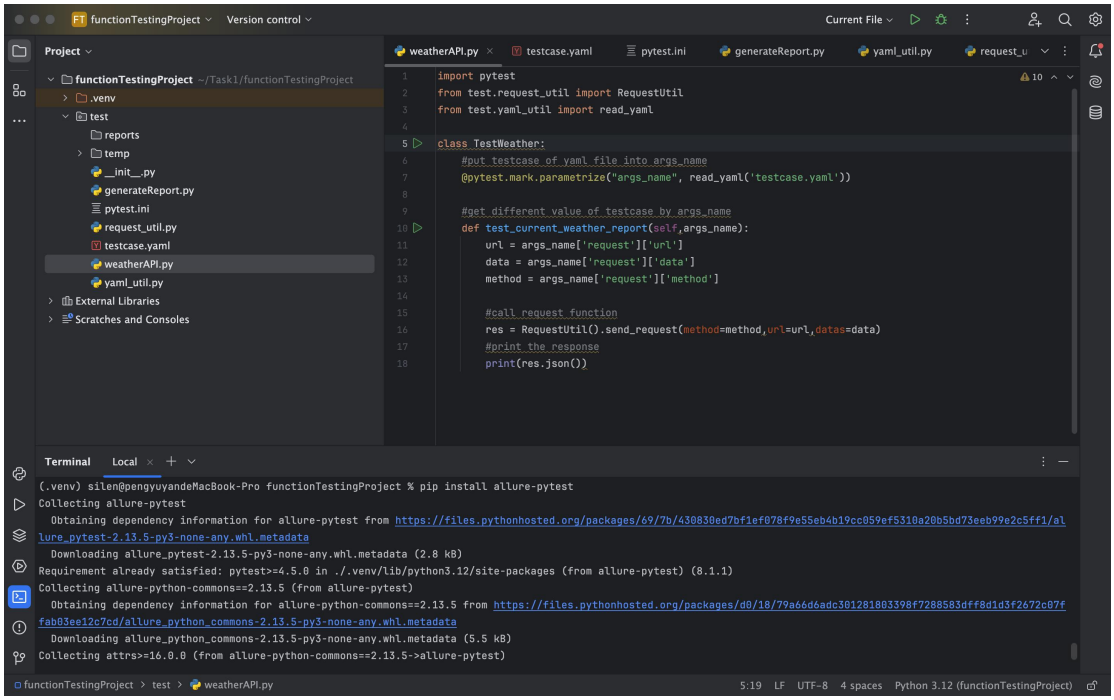
Can install allure by ‘brew install allure’ in terminal of pycharm.



Can check allure version by ‘allure --version’ after installation.

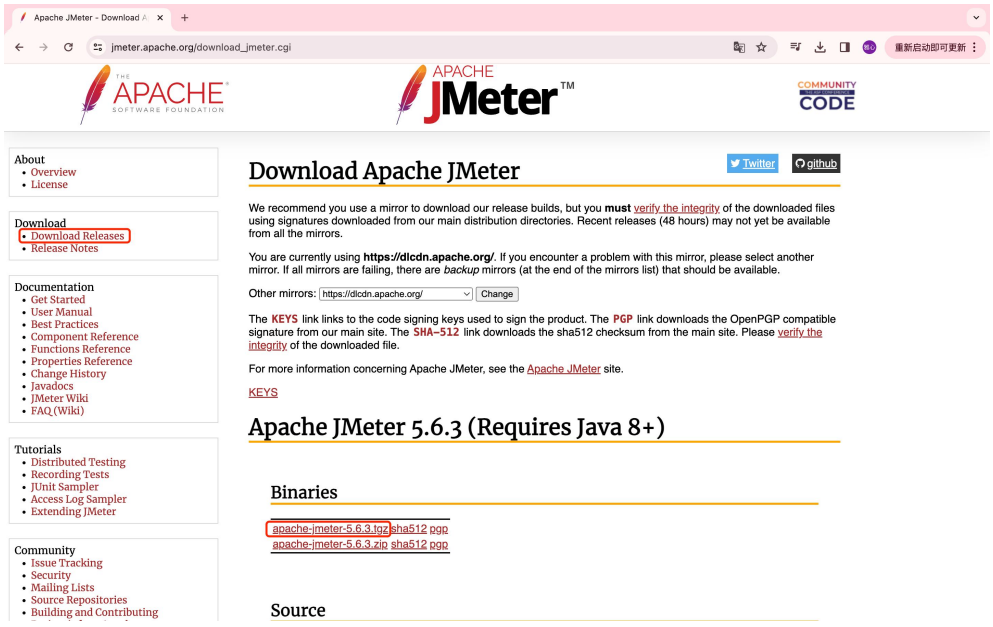


Then also need to install allure-pytest by using “pip install allure pytest” in terminal of pycharm.

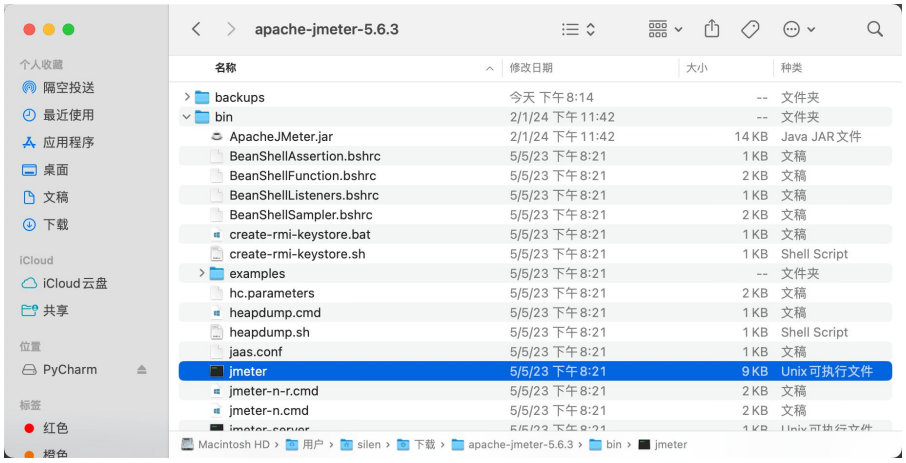


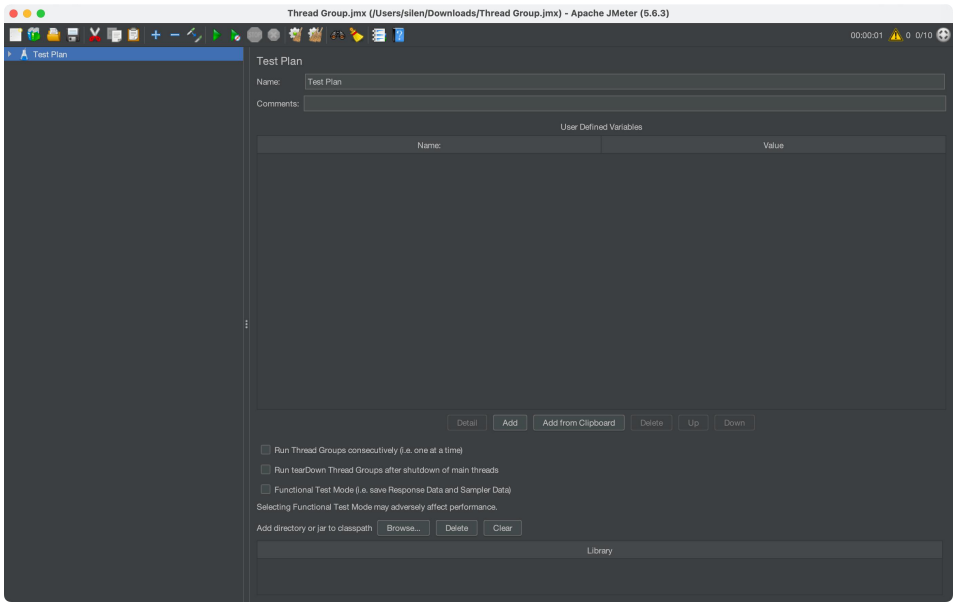
7. jmeter

Download from website then install following the guidelines.



Unzip the file and open jmeter in bin then can see the home screen of jmeter.

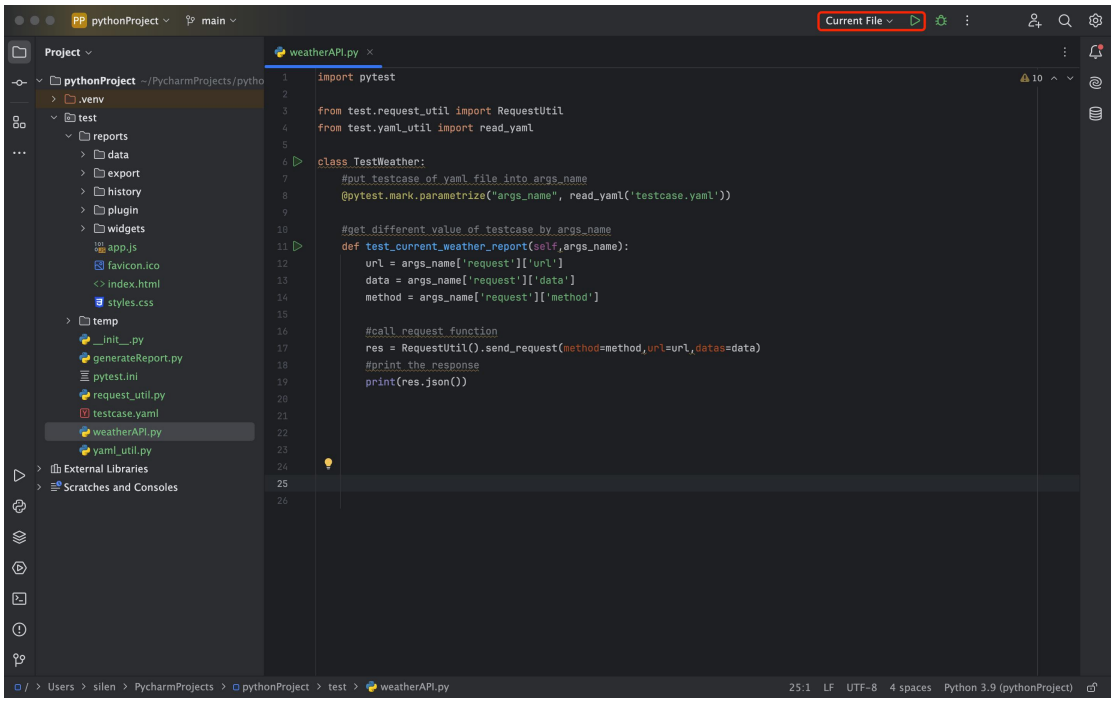




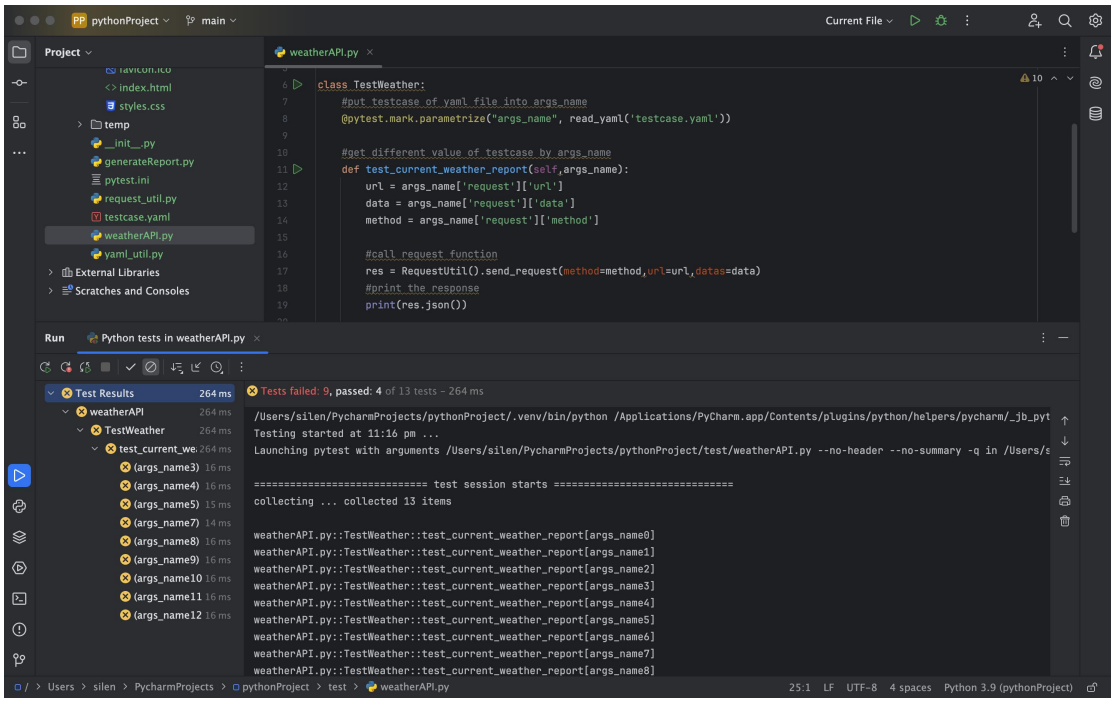
Guidelines:

1. Execute testcase of function testing

After opening python project, find weatherAPI.py file, then click run button.



Then can see the brief running info(responses) of testcases, 4 of 11 testcases are passed, 9 of 11 testcases are failed.



Testcases can be checked in testcase.yaml, including following scenarios:

Testcase1:Language of response-en

Testcase2:Language of response-tc

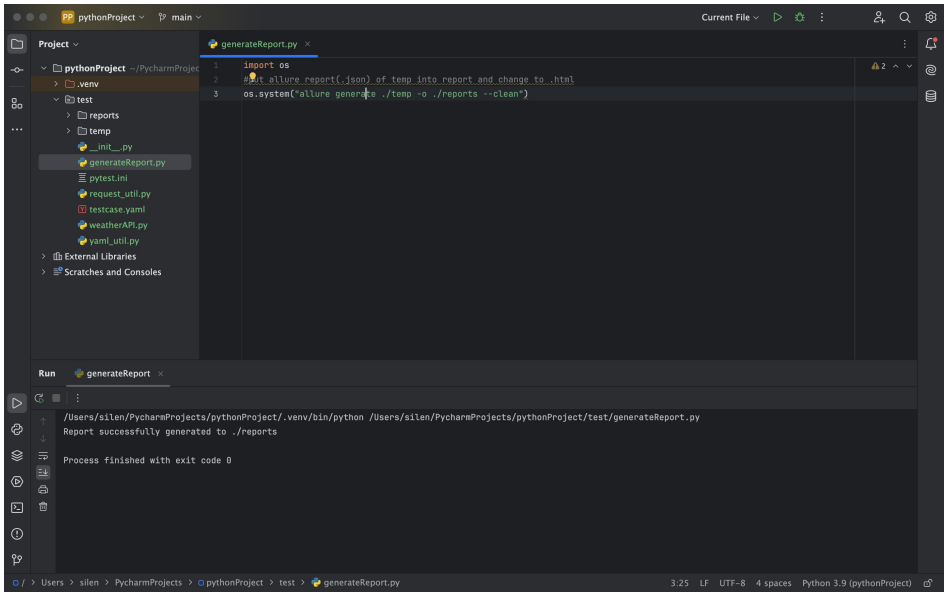
Testcase3:Language of response-sc

Testcase4:Missing dataType-en

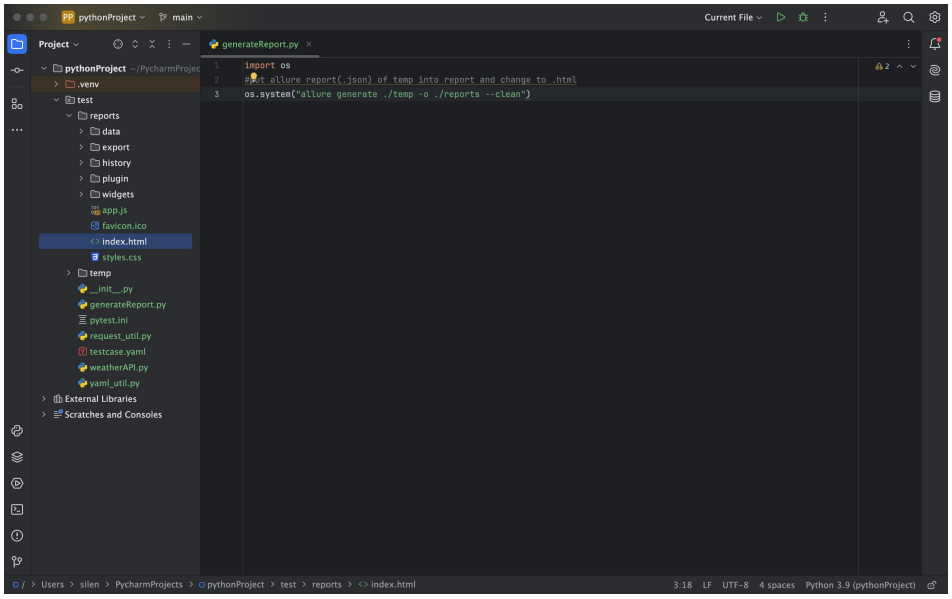
Testcase5:Missing dataType-tc
Testcase6:Missing dataType-sc
Testcase7:Missing lang
Testcase8:Missing dataType and lang
Testcase9:Wrong dataType-en
Testcase10:Wrong dataType-tc
Testcase11:Wrong dataType-sc
Testcase12:Wrong lang
Testcase13:Wrong dataType and lang
So only Testcase1, Testcase2, Testcase3 and Testcase7 should pass, the testing result meet expectation.

2. Generate function testing report

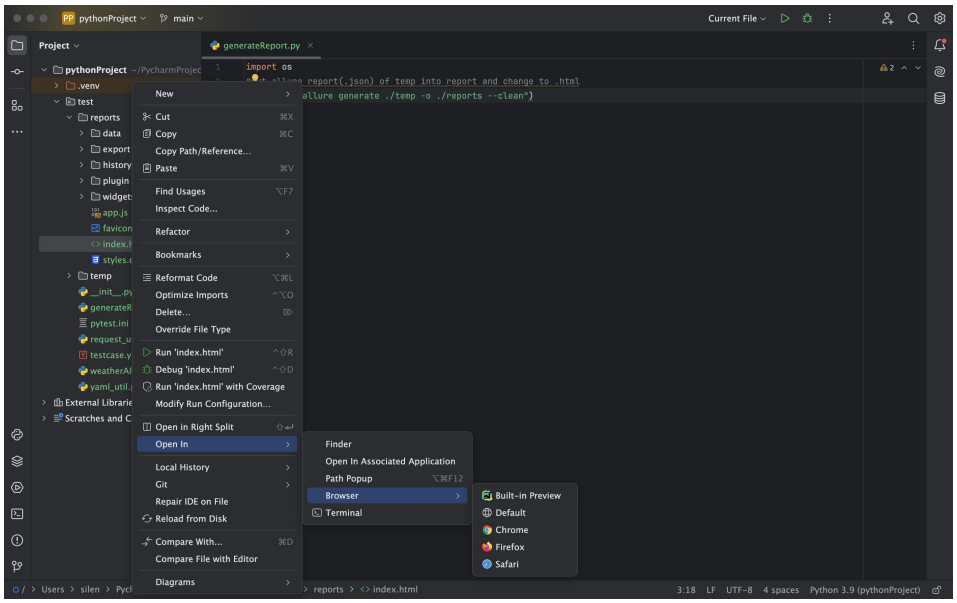
Find generateReport.py file, then click run button.

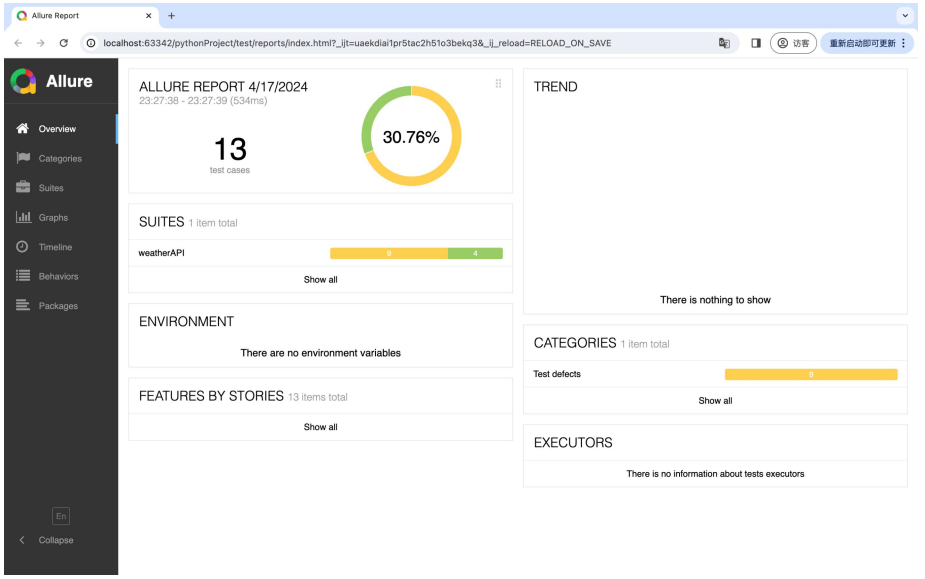


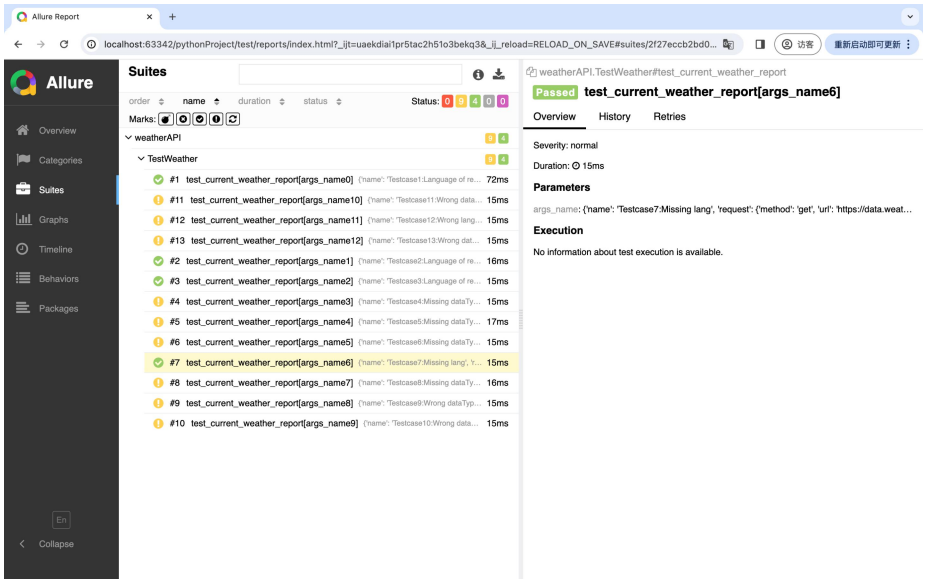
Then can see an index.html report is generated under reports directory.



Then open the index.html report in browser. Can check the detailed info of testcases.

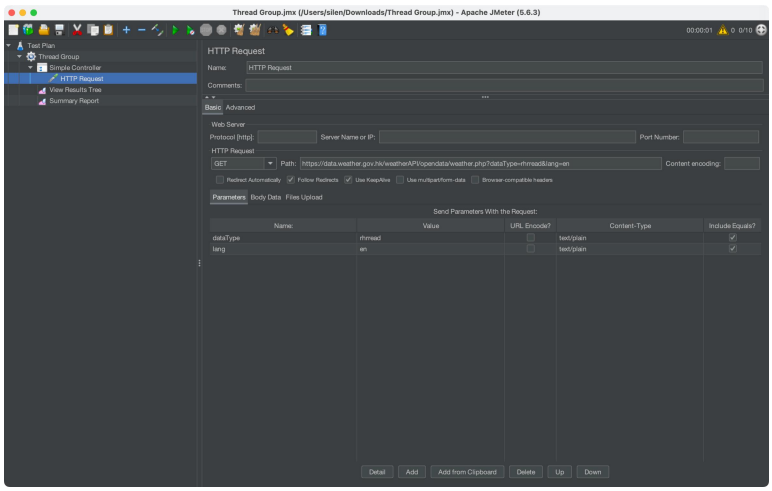




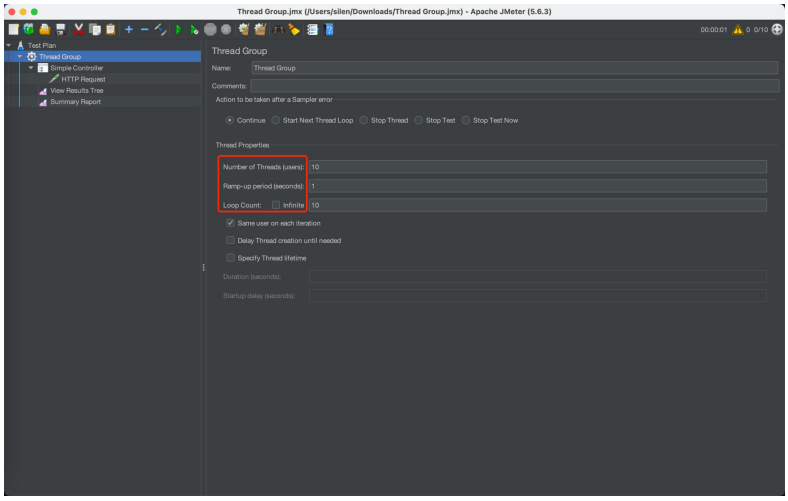


3. Performance testing

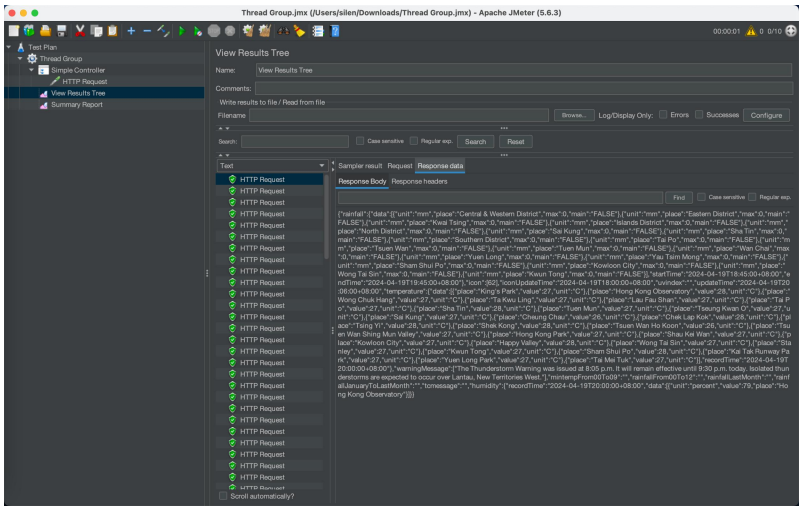
Add HTTP Request module with all request info. Then add View Results Tree module and Summary Report module to check the testing result.



Then can set the Number of Treads, Ramp-up period and Loop Count to simulate high concurrency for performance testing. As following setting, the Thread Group will make 10*10 requests. Then click execute button.



In View Results Tree, it shows all Request, Response Body and Response headers of every requests.



In Summary Report, it shows the general datas like min/max/avg corresponding speed of that 100 requests or fail rate etc. Then compare these datas with them in production environment to output the performance testing report.