1. Lagent Web Demo

1.1 使用 LMDeploy 部署

由于 Lagent 的 Web Demo 需要用到 LMDeploy 所启动的 api\_server，因此我们首先按照下图指示在 vscode terminal 中执行如下代码使用 LMDeploy 启动一个 api\_server。

conda activate agent

lmdeploy serve api\_server /root/share/new\_models/Shanghai\_AI\_Laboratory/internlm2-chat-7b \

--server-name 127.0.0.1 \

--model-name internlm2-chat-7b \

--cache-max-entry-count 0.1

1.2 启动并使用 Lagent Web Demo

接下来我们按照下图指示新建一个 terminal 以启动 Lagent Web Demo。在新建的 terminal 中执行如下指令：

conda activate agent

cd /root/agent/lagent/examples

streamlit run internlm2\_agent\_web\_demo.py --server.address 127.0.0.1 --server.port 7860

在等待 LMDeploy 的 api\_server 与 Lagent Web Demo 完全启动后（如下图所示），在本地进行端口映射，将 LMDeploy api\_server 的23333端口以及 Lagent Web Demo 的7860端口映射到本地。可以执行：

ssh -CNg -L 7860:127.0.0.1:7860 -L 23333:127.0.0.1:23333 root@ssh.intern-ai.org.cn -p 你的 ssh 端口号

接下来在本地的浏览器页面中打开 http://localhost:7860 以使用 Lagent Web Demo。首先输入模型 IP 为 127.0.0.1:23333，在输入完成后按下回车键以确认。并选择插件为 ArxivSearch，以让模型获得在 arxiv 上搜索论文的能力。

2. 用 Lagent 自定义工具

在本节中，我们将基于 Lagent 自定义一个工具。Lagent 中关于工具部分的介绍文档位于 https://lagent.readthedocs.io/zh-cn/latest/tutorials/action.html 。使用 Lagent 自定义工具主要分为以下几步：

继承 BaseAction 类

实现简单工具的 run 方法；或者实现工具包内每个子工具的功能

简单工具的 run 方法可选被 tool\_api 装饰；工具包内每个子工具的功能都需要被 tool\_api 装饰

下面我们将实现一个调用和风天气 API 的工具以完成实时天气查询的功能。

2.1 创建工具文件

首先通过 touch /root/agent/lagent/lagent/actions/weather.py（大小写敏感）新建工具文件，该文件内容如下：

import json

import os

import requests

from typing import Optional, Type

from lagent.actions.base\_action import BaseAction, tool\_api

from lagent.actions.parser import BaseParser, JsonParser

from lagent.schema import ActionReturn, ActionStatusCode

class WeatherQuery(BaseAction):

"""Weather plugin for querying weather information."""

def \_\_init\_\_(self,

key: Optional[str] = None,

description: Optional[dict] = None,

parser: Type[BaseParser] = JsonParser,

enable: bool = True) -> None:

super().\_\_init\_\_(description, parser, enable)

key = os.environ.get('WEATHER\_API\_KEY', key)

if key is None:

raise ValueError(

'Please set Weather API key either in the environment '

'as WEATHER\_API\_KEY or pass it as `key`')

self.key = key

self.location\_query\_url = 'https://geoapi.qweather.com/v2/city/lookup'

self.weather\_query\_url = 'https://devapi.qweather.com/v7/weather/now'

@tool\_api

def run(self, query: str) -> ActionReturn:

"""一个天气查询API。可以根据城市名查询天气信息。

Args:

query (:class:`str`): The city name to query.

"""

tool\_return = ActionReturn(type=self.name)

status\_code, response = self.\_search(query)

if status\_code == -1:

tool\_return.errmsg = response

tool\_return.state = ActionStatusCode.HTTP\_ERROR

elif status\_code == 200:

parsed\_res = self.\_parse\_results(response)

tool\_return.result = [dict(type='text', content=str(parsed\_res))]

tool\_return.state = ActionStatusCode.SUCCESS

else:

tool\_return.errmsg = str(status\_code)

tool\_return.state = ActionStatusCode.API\_ERROR

return tool\_return

def \_parse\_results(self, results: dict) -> str:

"""Parse the weather results from QWeather API.

Args:

results (dict): The weather content from QWeather API

in json format.

Returns:

str: The parsed weather results.

"""

now = results['now']

data = [

f'数据观测时间: {now["obsTime"]}',

f'温度: {now["temp"]}°C',

f'体感温度: {now["feelsLike"]}°C',

f'天气: {now["text"]}',

f'风向: {now["windDir"]}，角度为 {now["wind360"]}°',

f'风力等级: {now["windScale"]}，风速为 {now["windSpeed"]} km/h',

f'相对湿度: {now["humidity"]}',

f'当前小时累计降水量: {now["precip"]} mm',

f'大气压强: {now["pressure"]} 百帕',

f'能见度: {now["vis"]} km',

]

return '\n'.join(data)

def \_search(self, query: str):

# get city\_code

try:

city\_code\_response = requests.get(

self.location\_query\_url,

params={'key': self.key, 'location': query}

)

except Exception as e:

return -1, str(e)

if city\_code\_response.status\_code != 200:

return city\_code\_response.status\_code, city\_code\_response.json()

city\_code\_response = city\_code\_response.json()

if len(city\_code\_response['location']) == 0:

return -1, '未查询到城市'

city\_code = city\_code\_response['location'][0]['id']

# get weather

try:

weather\_response = requests.get(

self.weather\_query\_url,

params={'key': self.key, 'location': city\_code}

)

except Exception as e:

return -1, str(e)

return weather\_response.status\_code, weather\_response.json()

2.2 获取 API KEY

为了获得稳定的天气查询服务，我们首先要获取 API KEY。首先打开 https://dev.qweather.com/docs/api/ 后，点击右上角控制台。（如下图所示）

2.3 体验自定义工具效果

与 1.2 部分类似，我们在两个 terminal 中分别启动 LMDeploy 服务和 Tutorial 已经写好的用于这部分的 Web Demo：

conda activate agent

lmdeploy serve api\_server /root/share/new\_models/Shanghai\_AI\_Laboratory/internlm2-chat-7b \

--server-name 127.0.0.1 \

--model-name internlm2-chat-7b \

--cache-max-entry-count 0.1

export WEATHER\_API\_KEY=在2.2节获取的API KEY

# 比如 export WEATHER\_API\_KEY=1234567890abcdef

conda activate agent

cd /root/agent/Tutorial/agent

streamlit run internlm2\_weather\_web\_demo.py --server.address 127.0.0.1 --server.port 7860

