

Django 课前准备

Lambda X

2022-07-22

课程内容

- 学会使用 Django 处理基本的请求与响应
- 进行 Django + Nginx + uWSGI 部署

课前准备

在本地计算机安装环境

只需在虚拟环境中

```
pip install Django pymysql
```

然后执行

```
python -m django --version
```

检查 Django 是否安装成功

如果使用 PyCharm，直接新建 Django 项目就可以进行安装并自动新建 Django 项目，同时会配置好如 Django Shell，模板高亮等特性，（虽然可能还是命令行更方便

安装 Postman

Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.

[Postman](#) 可以用来发送请求，从而调用和调试接口。

可以从它的官网下载并进行安装，然后阅读 [Getting Started](#) 来学习 Postman 的使用。

对于基本的需求来说，可以直接在 Workspace 中点击 "+" 来新建并发送请求，也可以新建一个 Collection 并对特定系列 API 的调用进行配置。

在服务器配置 Python 环境

安装 miniconda

在 Python 课前准备中，我们介绍了在 Windows 和 MacOS 中如何安装 miniconda。现在我们需要在一台 Linux 服务器上安装 miniconda。这里以我们分配的服务器镜像为例说明。

虽然 sudo 权限对于安装 conda 不是必要的，但考虑到启动 Nginx 需要 sudo 权限，这里的操作也用一个具有 sudo 权限的用户完成（但无需任何 sudo）。在分发的服务器镜像中，可以使用账户 `nana` 完成。

1. 从 Tuna 源下载 miniconda 安装包

```
curl -fSSL
https://mirrors.tuna.tsinghua.edu.cn/anaconda/miniconda/Miniconda3-latest-
Linux-x86_64.sh > conda.sh
```

2. 给安装包添加权限并运行

```
chmod +x ./conda.sh && ./conda.sh
```

3. 反复按 `ENTER` 来阅读许可协议。如果您接受许可协议请输入 `yes`

```
Please answer 'yes' or 'no':
>>> yes
```

4. 询问安装位置，可以直接 `ENTER`

5. 询问是否希望 installer 来进行初始化，输入 `yes`

```
installation finished.
Do you wish the installer to initialize Miniconda3
by running conda init? [yes|no]
[no] >>> yes
```

6. 待操作完成后重新进入终端，此时 conda 应该已经默认启动

```
(base)
nana@SAST2022-Training [19:34:43] [~]
-> % which python
/home/nana/miniconda3/bin/python
```

然后为 conda 和 pip 换源。

配置环境

新建虚拟环境使 `python==3.10` 并安装以下包

```
pip install django pymysql
conda install -c conda-forge uwsgi
```

Nginx

了解反向代理

<https://www.zhihu.com/question/24723688>

启动 Nginx

使用

```
sudo service nginx status
```

查看 Nginx 运行状态

使用

```
sudo service nginx start
```

启动 Nginx

Nginx 启动后，可以测试访问是否正常

```
-> % curl localhost
<!DOCTYPE html>
<html>
<head>
<title>welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

可以正常显示 Nginx 默认界面。

此时，访问服务器的 80 端口（如果你使用了科协提供的容器，那么需要查看 80 端口被映射到了外网的哪个端口，然后可以从外网使用 IP 和端口进行访问），看到相同的页面。

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

课后练习

这次课后练习将实现一个 Leaderboard 的后端，并与相应的前端进行对接。练习将在课后发布。

在后续的前端课程中，您还会学习 React，并完 Leaderboard 前端的开发工作；在本次练习中，我们会给出一个前端的参考实现，可按照接口约定直接对接。