

项目创建、模板使用及版本受控

项目创建、模板使用及版本受控

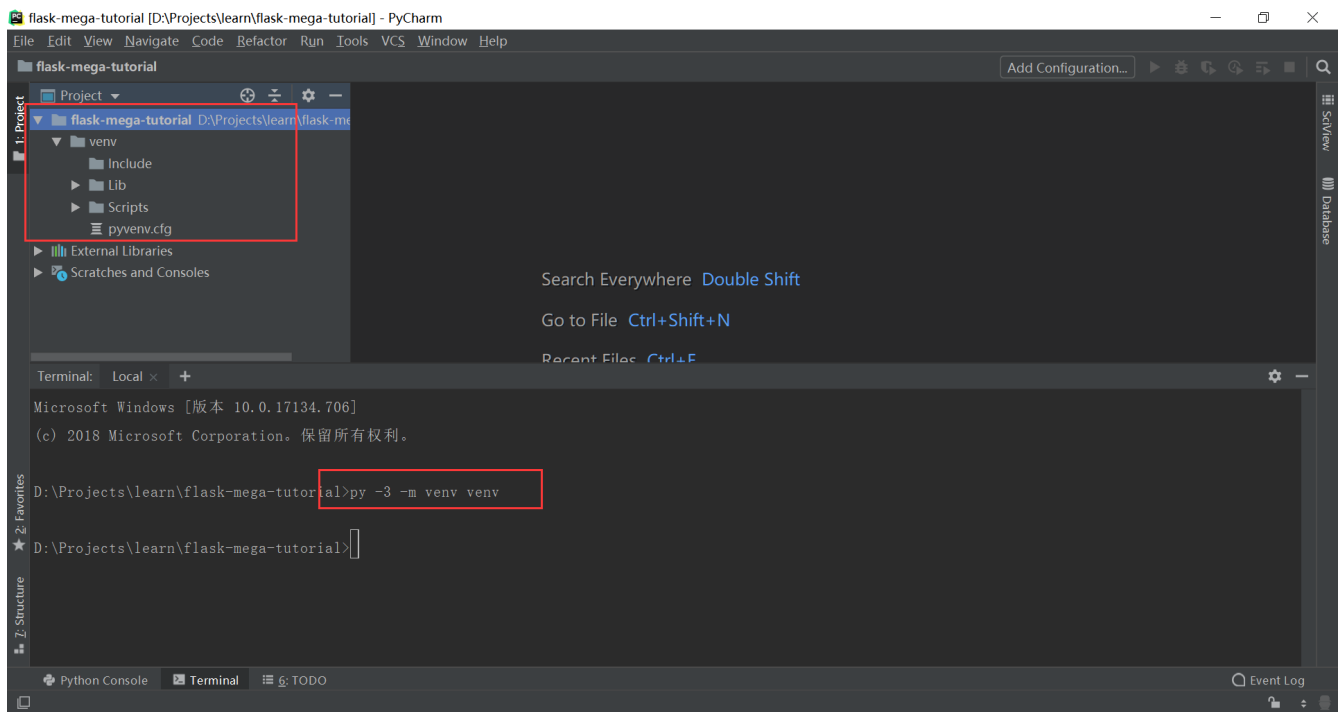
1. 创建项目
2. 设置项目依赖环境
3. 安装环境依赖库
4. 创建项目基础目录
5. 创建应用的工厂函数
6. 创建应用视图代码
7. 注册视图URL
8. 配置pycharm启动设置
9. 启动服务
 - 9.1. pycharm启动服务方式
 - 9.2. cmd命令行启动服务方式
10. 访问服务地址
11. 编写base模板文件
12. 编写hello模板文件
13. 修改视图函数渲染模板文件
14. 重启服务，查看结果
15. 创建远程git仓库，代码版本受控
16. 初始化本地git仓库，配置远程仓库

1. 创建项目

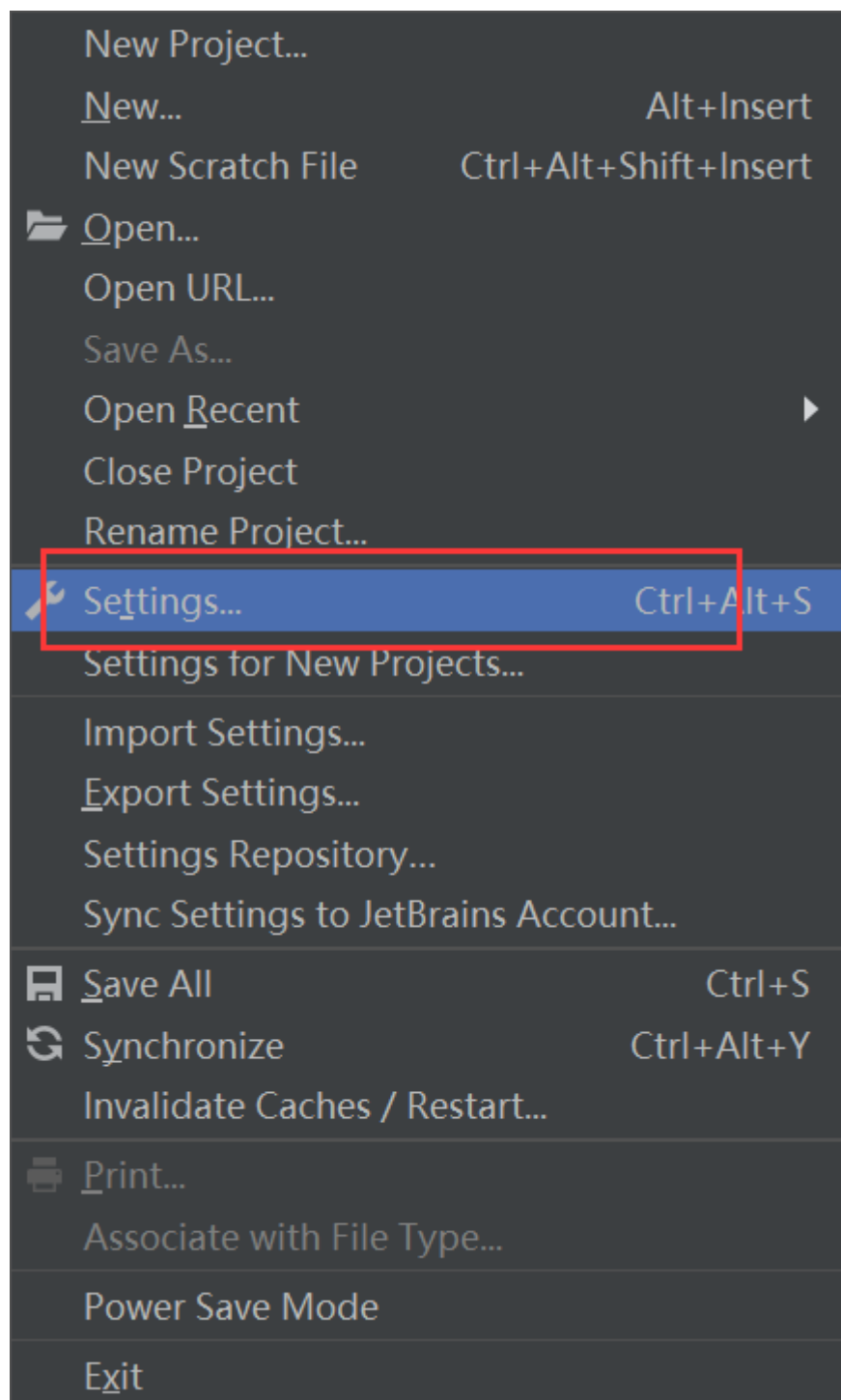
创建项目目录 `flask-mega-tutorial1`

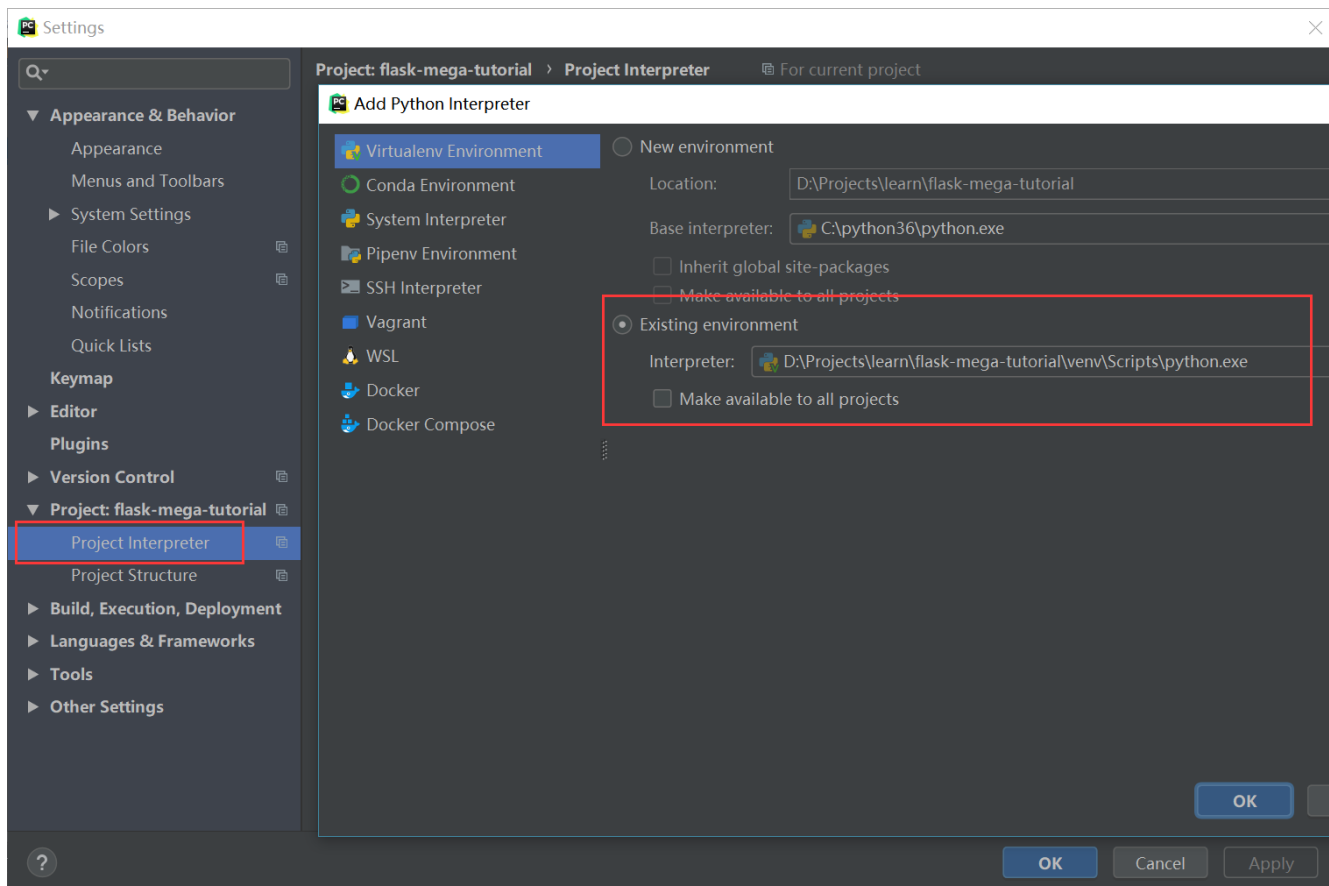
使用pycharm打开项目目录，使用命令行工具创建虚拟环境

注：python3.4+环境均已内部支持虚拟环境创建，执行命令为 `py -3 -m venv venv`；对于python3.3-环境的均需要安装virtualenv三方库进行虚拟环境安装。



2. 设置项目依赖环境

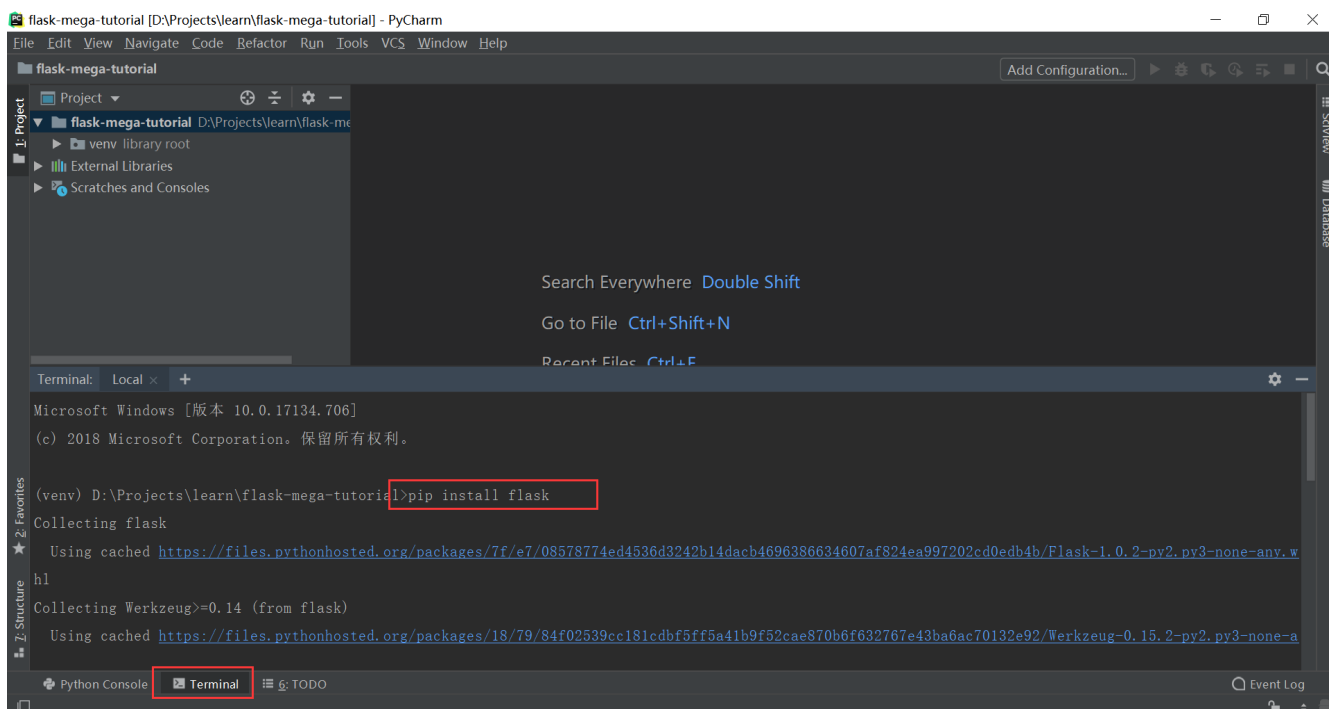




3. 安装环境依赖库

设置完虚拟环境，IDE会自动加载虚拟环境，通过 `pip install flask` 命令安装flask三方库。

若通过windows中的cmd命令行操作，可以通过 `.\venv\Scripts\activate.bat` 命令进入虚拟环境。

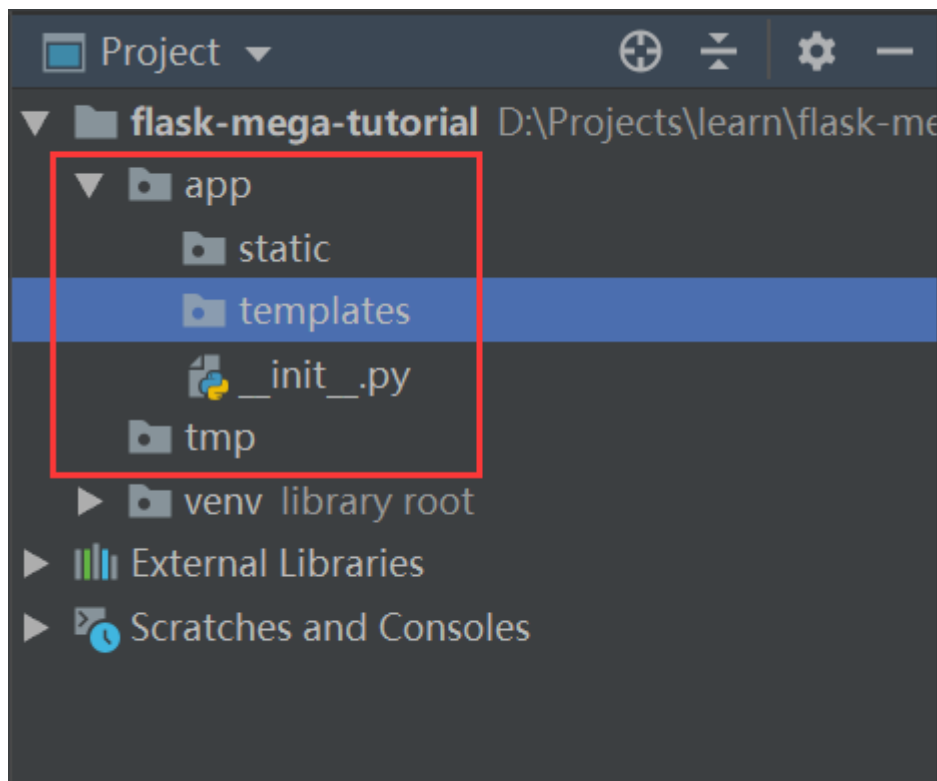


需要安装的库如下：

```
pip install flask
pip install flask-login
pip install flask-openid
pip install flask-mail
pip install flask-sqlalchemy
pip install sqlalchemy-migrate
pip install flask-whooshalchemy
pip install flask-wtf
pip install flask-babel
pip install guess_language
pip install flipflop
pip install coverage
```

4. 创建项目基础目录

其中应用程序包放置在 `app` 目录，子文件夹 `static` 存放静态文件像图片，JS文件以及样式文件。子文件夹 `templates` 是存放模板文件。



5. 创建应用的工厂函数

修改 `app` 包中 `app/__init__.py` 文件，编写应用的工厂函数，项目启动后会自动加载工厂函数 `create_app`。flask 会自动加载的工厂函数名包括：`create_app`、`make_app`。

```
import os
from flask import Flask
```

```
def create_app():
    """应用工厂函数"""
    # instance_relative_config: 设置配置文件以instance文件夹为相对文件夹
    # 即config.py配置文件放置在instance文件夹下。
    app = Flask(__name__, instance_relative_config=True)

    try:
        # 确保 app.instance_path 存在
        os.makedirs(app.instance_path)
    except OSError:
        pass

    return app
```

6. 创建应用视图代码

新建视图函数文件 app/hello.py

```
from flask.views import View

class HelloWorld(View):
    methods = ['GET']

    def dispatch_request(self):
        return 'Hello world!'
```

7. 注册视图URL

修改应用工厂函数，新增视图注册

```
import os
from flask import Flask

def create_app():
    """应用工厂函数"""
    # instance_relative_config: 设置配置文件以instance文件夹为相对文件夹
    # 即config.py配置文件放置在instance文件夹下。
    app = Flask(__name__, instance_relative_config=True)

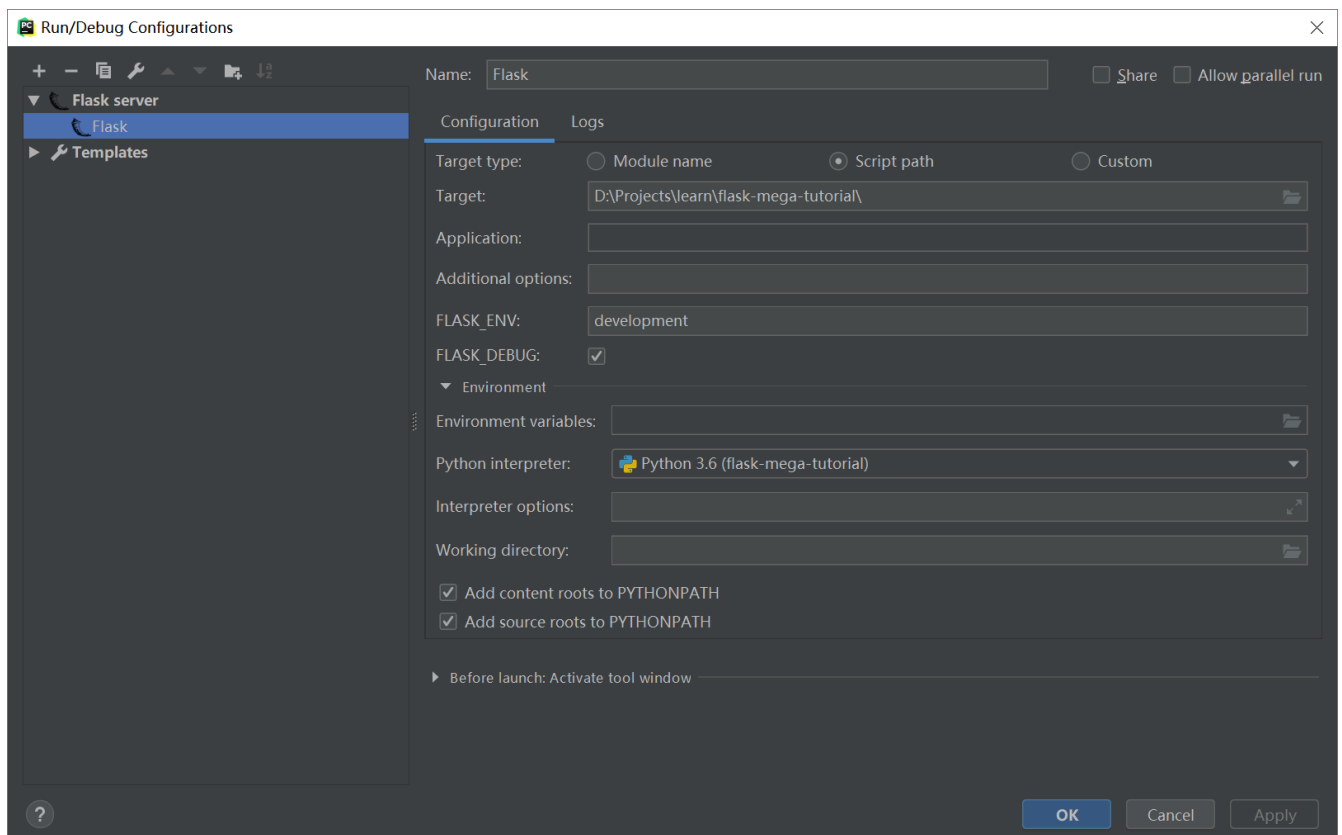
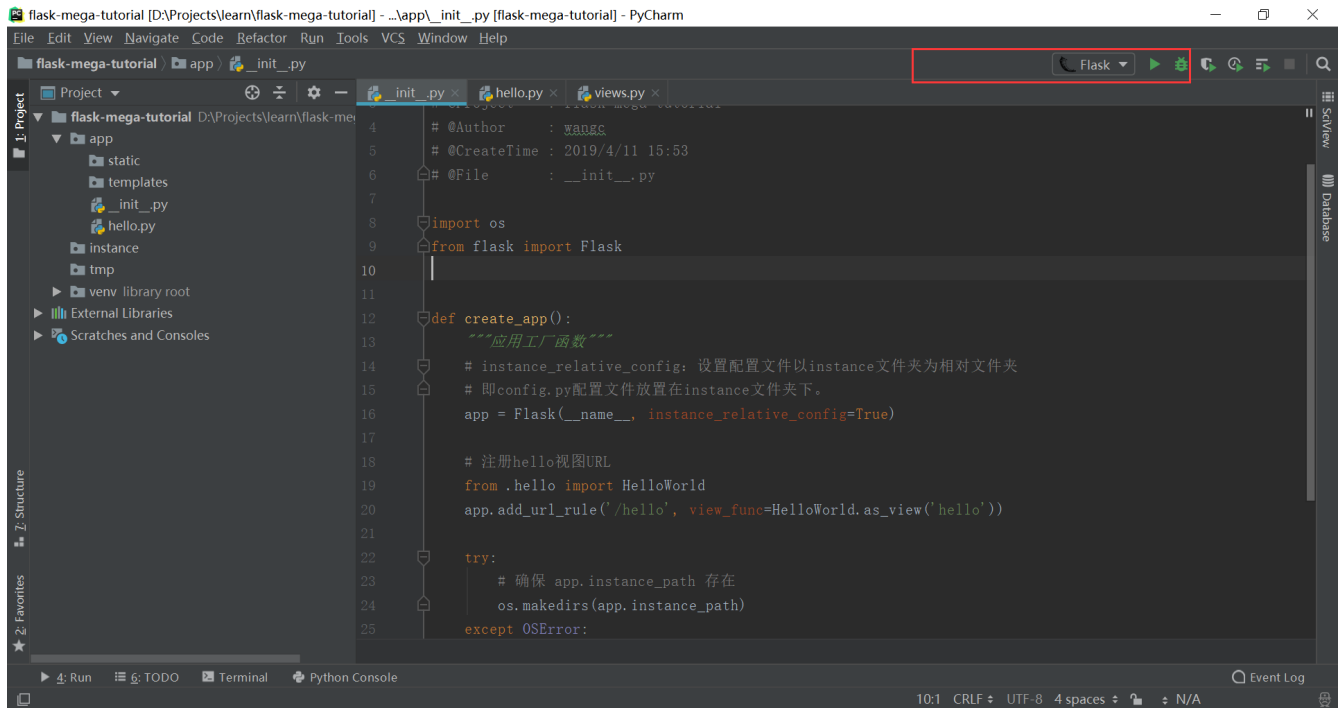
    # 注册hello视图URL
    from .hello import HelloWorld
    app.add_url_rule('/hello', view_func=HelloWorld.as_view('hello'))

    try:
        # 确保 app.instance_path 存在
        os.makedirs(app.instance_path)
```

```
except OSError:
    pass

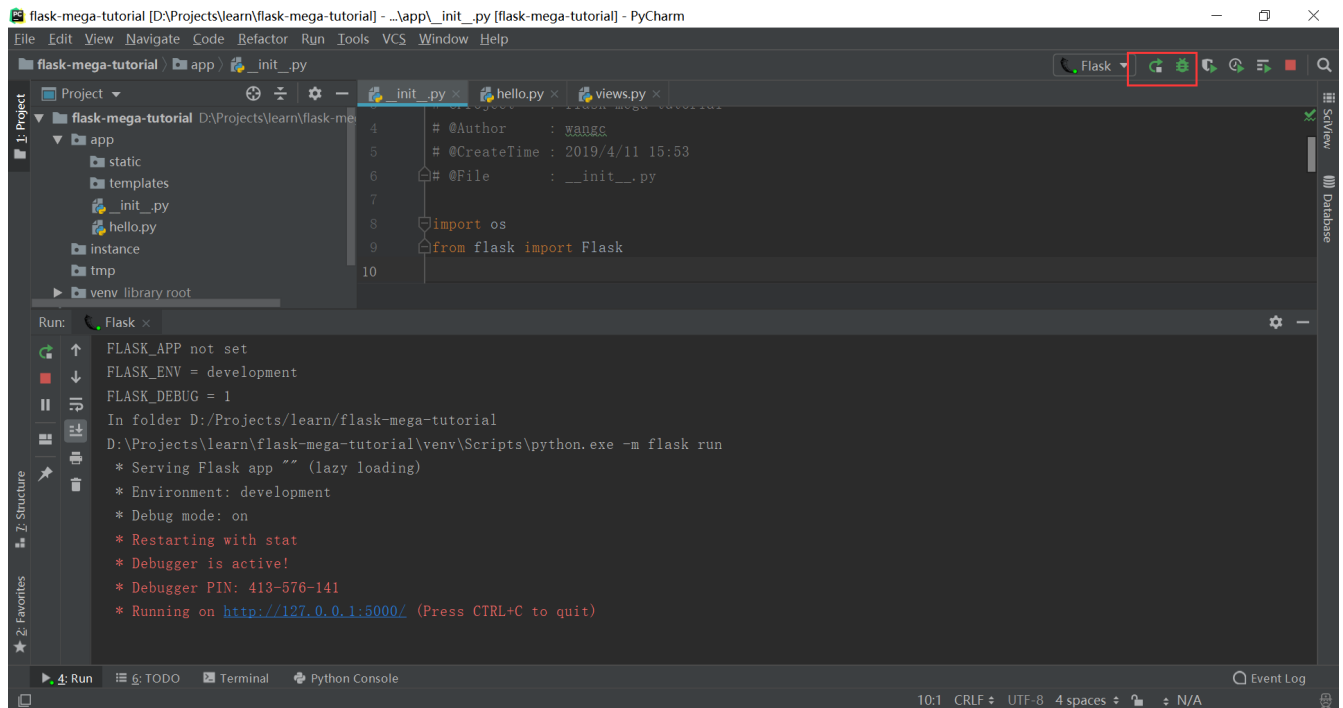
return app
```

8. 配置pycharm启动设置



9. 启动服务

9.1. pycharm启动服务方式



9.2. cmd命令行启动服务方式

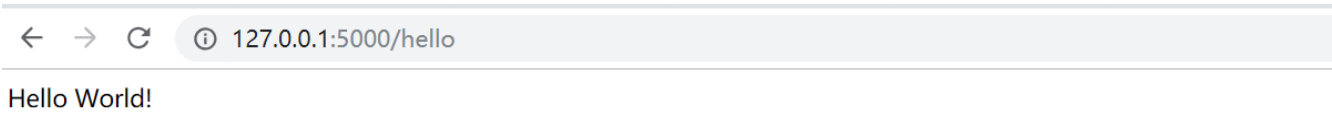
进入虚拟环境

```
.\venv\Scripts\activate.bat
```

通过 `flask run` 命令启动服务

```
(venv) $ flask run
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

10. 访问服务地址



11. 编写base模板文件

创建并编辑模板文件 `app/templates/base.html`

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  {% if title %}
    <title>{{ title }} - 博客</title>
  {% else %}
    <title>博客</title>
  {% endif %}
</head>
<body>
  <div>博客: <a href="{{ url_for('hello') }}">首页</a></div>
  <hr>
  {% block content %}{% endblock %}
</body>
</html>
```

12. 编写hello模板文件

创建并编辑模板文件 `app/templates/hello/hello.html`，继承base.html

```
{% extends 'base.html' %}

{% block title %}Hello world!{% endblock %}

{% block content %}
  <h1>Hi, {{ user.nickname }}!</h1>
  {% for post in posts %}
    <div><p>{{ post.author.nickname }} says: <b>{{ post.body }}</b></p></div>
  {% endfor %}
{% endblock %}
```

13. 修改视图函数渲染模板文件

```
from flask.views import View
from flask import render_template

class HelloWorld(View):
    methods = ['GET']

    def dispatch_request(self):
        user = {'nickname': 'Super.Wong'}
        posts = [
            {
                'author': {'nickname': '李白'},
                'body': '举头望明月，低头思故乡'
            },
            {
                'author': {'nickname': '李清照'},
                'body': '知否，知否，应是绿肥红瘦'
            }
        ]
        return render_template('hello/hello.html', title='测试', user=user, posts=posts)
```

14. 重启服务，查看结果

← → ↻ ⓘ 127.0.0.1:5000/hello



博客: [首页](#)

Hi, Super.Wong!

李白 says: 举头望明月，低头思故乡

李清照 says: 知否，知否，应是绿肥红瘦

15. 创建远程git仓库，代码版本受控

 **super.wong / flask-mega-tutorial** Python MIT 

👁 Unwatch 1 ☆ Star 0 🍴 Fork 0

[📄 代码](#) [🔔 Issues 0](#) [🔗 Pull Requests 0](#) [📎 附件 0](#) [📖 Wiki 0](#) [📊 统计](#) [🔧 DevOps](#) [🛠 服务](#) [👤 管理](#)

较为大型的博客联系

编辑

👁 1 次提交

🌿 1 个分支

🏷 0 个标签

📦 0 个发行版

👤 1 位贡献者

master ▾

+ Pull Request

+ Issue









文件 ▾

Web IDE

🔗 挂件

克隆/下载 ▾

 super.wong 最后提交于 刚刚 Initial commit

 .gitignore	 super.wong	Initial commit	刚刚
 LICENSE	 super.wong	Initial commit	刚刚
 README.en.md	 super.wong	Initial commit	刚刚
 README.md	 super.wong	Initial commit	刚刚

16. 初始化本地git仓库，配置远程仓库

通过 `git init` 命令初始化本地仓库

```
D:\Projects\learn\flask-mega-tutorial
(venv) $ git init
Initialized empty Git repository in D:/Projects/learn/flask-mega-tutorial/.git/
```

通过 `git remote add` 命令添加远程仓库

```
D:\Projects\learn\flask-mega-tutorial (master -> origin)
(venv) $ git remote add origin https://gitee.com/superwong/flask-mega-tutorial.git
```

拉取远程仓库代码

```
D:\Projects\learn\flask-mega-tutorial (master -> origin)
(venv) $ git pull origin master
From https://gitee.com/superwong/flask-mega-tutorial
* branch                master       -> FETCH_HEAD
```

修改.gitignore文件，增加IDE工具配置目录的版本控制忽略

```
# IDE
.vscode/
.idea/
```

通过命令 `git add .` 提交本地所有未提交代码

```
D:\Projects\learn\flask-mega-tutorial (master -> origin)
(venv) $ git status -u
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   .gitignore

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        app/__init__.py
        app/hello.py
        app/templates/hello.html

no changes added to commit (use "git add" and/or "git commit -a")

D:\Projects\learn\flask-mega-tutorial (master -> origin)
(venv) $ git add .
```

通过 `git commit` 命令提交本地仓库

```
D:\Projects\learn\flask-mega-tutorial (master -> origin)
(venv) $ git commit -m '首次提交'
[master 8f1d3c4] '首次提交'
 4 files changed, 57 insertions(+)
 create mode 100644 app/__init__.py
 create mode 100644 app/hello.py
 create mode 100644 app/templates/hello.html
```

通过 `git push` 命令推送到远程仓库

```
D:\Projects\learn\flask-mega-tutorial (master -> origin)
(venv) $ git push --set-upstream origin master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 1.43 KiB | 734.00 KiB/s, done.
Total 8 (delta 1), reused 0 (delta 0)
remote: Powered By Gitee.com
To https://gitee.com/superwong/flask-mega-tutorial.git
    d832605..8f1d3c4  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```