



链滴

# Flask Web 测试驱动开发最佳实践 之 开篇

作者: [guoweikuang](#)

原文链接: <https://ld246.com/article/1542377915162>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## 前言

---

一直都有听到 TDD 测试驱动开发的开发方式，之前看了一本《Python Web开发测试驱动方法》，本书基于 Django 框架严格遵循测试驱动开发（TDD）的方式去实现一个简单网站。一直都很想尝试这种方式进行一次简单的TDD实践，这篇文章就使用我之前的 仿V2EX 网站项目来一步步实践。

## TDD

---

测试驱动开发（Test-driven development）简称 TDD，是一种软件开发过程中的应用方法，由极限编程中倡导，以其倡导先写测试程序，然后编码实现其功能得名。

测试驱动开发是戴两顶帽子思考的开发方式：先戴上实现功能的帽子，在测试的辅助下，快速实现其能；再戴上重构的帽子，在测试的保护下，通过去除冗余的代码，提高代码质量。测试驱动着整个过程：首先，驱动代码的设计和功能的实现；其后，驱动代码的再设计和重构。**以上概念来源于维基科**

## 需求分析

---

相信各位读者都已经知道 [V2EX](#) 社区，首先一个技术社区需要有以下几个功能：

- 话题：发布话题、修改话题、追加话题等
- 节点：话题的归类
- 用户：标识你在社区的身份

- 评论：评论话题，技术社区最主要的功能
- 用户关系：用户之间能相互关注

以上几个功能对一个技术社区来说是十分重要的，本文是要使用 TDD 方式实现这样一个社区，因此这里介绍一下

## 编写测试

---

首先测试驱动开发的第一步就是：编写测试。上一节已经提到了一些需求，这里就根据这些需求写一个简单的测试，并设法让其通过。首先是用户功能，一个用户需要有注册、验证、登录等步骤后才能真在社区中发布话题等功能。

本文使用 [GitHub](#) 来托管项目代码，这里新建一个项目名称叫 v2ex, 初始结构如下：

```
v2ex
├── LICENSE
└── README.md
```

然后新建一个 tests 文件夹，用于保存测试代码，为什么要建 tests 文件夹，而不是在根路径下建个试文件就好了，这不是良好的项目结构，本文根据项目最佳结构来实践。然后在 tests 文件夹下新建试文件 test\_user.py, 下面是项目的当前结构：

```
v2ex
├── LICENSE
├── README.md
└── tests
    └── test_user.py
```

1 directory, 3 files

项目已经存在了，先别着急编写测试代码，先把开发环境搭建起来，下一节介绍如何搭建一个完美的 python 开发环境。

## 虚拟环境

---

本文基于 python3.7 版本，使用 pipenv 来管理开发环境，可能有些人没有使用过 pipenv。这东西于管理虚拟环境而言太好用了，具体使用可参照[官网](#)或者 [使用pipenv管理你的项目](#) 这篇文章来使用。

本文假设你已经存在了 Python3.7 环境了，然后可以通过 pip 来安装 pipenv

```
pip install pipenv -i https://pypi.tuna.tsinghua.edu.cn/simple(如果配了就不要-i后面的)
```

如果不了解 pipenv，可以通过下面来参考相关命令：

```
$ pipenv -h
.....(省略)
Usage Examples:
  Create a new project using Python 3.6, specifically:
  $ pipenv --python 3.6
```

Install all dependencies for a project (including dev):

```
$ pipenv install --dev
```

Create a lockfile containing pre-releases:

```
$ pipenv lock --pre
```

Show a graph of your installed dependencies:

```
$ pipenv graph
```

Check your installed dependencies for security vulnerabilities:

```
$ pipenv check
```

Install a local setup.py into your virtual environment/Pipfile:

```
$ pipenv install -e .
```

Use a lower-level pip command:

```
$ pipenv run pip freeze
```

安装后 pipenv 后，在 v2ex 根路径下创建虚拟环境，即是通过

```
$ pipenv --python 3.7
```

执行完这条命令后，就默认帮你创建好一个Python虚拟环境了，然后通过

```
$ pipenv shell 进入虚拟环境中
```

仔细观察 v2ex 根路径多了个 Pipfile 文件，该文件其实和以前的 requirements.txt 文件一样保存项的各种依赖库的信息，这个Pipfile 文件结构如下：

```
[[source]]
```

```
url = "https://pypi.tuna.tsinghua.edu.cn/simple" # 修改成国内源
```

```
verify_ssl = true
```

```
name = "pypi"
```

```
[packages]
```

```
[dev-packages]
```

```
[requires]
```

```
python_version = "3.7"
```

so，因为使用的是 Flask 框架，那就先安装一下 Flask 吧，使用以下命令安装，注意不是使用 pip 安装了

```
$ pipenv install flask
```

如果安装过程出现TypeError: 'module' object is not callable 错误  
你需要

```
$ pipenv run pip install pip==18.0
```

```
$ pipenv install
```

最简单的 Flask 环境已经搭建好了，可以使用 git 进行一次提交了，本文会基于 git 来分阶段提交代，不了解 git 的同学要自己反省一下了。

下面看看当前项目的文件结构

```
v2ex
├── LICENSE
├── Pipfile
├── Pipfile.lock
├── README.md
├── tests
│   └── test_user.py
```

然后使用Git提交至远程仓库

在 v2ex 的根路径下

```
$ git add .
$ git commit -m "add tests and pipfile"
$ git push origin master
```

## 总结

---

本文主要介绍了使用 TDD 方式实现一个技术社区，并初步介绍了 TDD 的概念，然后构建了项目的试结构，介绍了社区的相关需求，后续会根据需求来开发，最后搭建了一个Python 开发环境来隔离地环境。本文主要兼顾各层次的读者，所以就介绍了一些最基本的东西，后续主要以 TDD 开发实现主了。

1234