

人工智能的 Python Web 高级开发课程研究

王俭

(广东省汕尾职业技术学院)

Python Web 高级开发主要以 Django 为代表的核心框架开发。编程语言仅仅是个工具,要想通过技术实现自己的价值,终究离不开具体的应用场景,而应用场景繁多,我们的时间和精力都是有限的,不可能什么都能学好,必须做出选择。Python Web 高级开发主要的应用领域有 5 大类,自动化测试,运维,爬虫,数据分析以及游戏开发。这几个领域的入门难度从自动化到机器学习也是从易到难,尤其是机器学习,对 Python 的要求并不高,反而是数学及算法要精通。此外,真正上手操作的时候需要的计算机技术的实操能力较强。

一、课程目标

Python Web 高级开发是高职院校大数据及人工智能技术应用专业学生的一门重要的必修专业基础课程。学生通过学习本课程,能够在已有的计算机基础知识上,对 web 高级开发有一个系统的、全面的了解,为其职业规划发展、社会实践、深入学习专业奠定基础。

二、学科核心素养

高等职业教育专科 Python Web 高级开发课程学科核心素养包括:Web 高级开发创新与发展、计算科学思维、动手和社会实践能力。

对于那些刚接触编程的人,可能想知道到底什么是 Web 开发。尽管不可能精确地指出一个确切的术语,但是 Web 开发可以粗略地定义为构建、创建和维护一个网站。通常,Web 开发涉及一个前端,与客户端交互的所有内容,以及一个后端,包含业务逻辑并与数据库交互。

三、开发创新与发展

将学习如何建立与 MySQL 数据库连接的网站项目,使用 django-admin 创建此项目,并安装和创建 MySQL 数据库,然后将项目与数据库连接。这个步骤就至少包括 6 步:

先决条件:在 Ubuntu 16.04 服务器上安装 Django。安装 MySQL。安装并配置了 Django 开发环境之后,就可以继续创建项目了。

第一步:创建 Django 项目

需要使用 django-admin 命令创建项目,这是基础。我们需要做的第一件事是进入到主目录,用以下命令:

```
$ cd ~
```

接下来,列出当前目录的内容:

```
$ ls
```

如果从本系列的开头处开始,你将注意到有一个目录(译者注:在本文之前,作者有另外的项目,这里读者可以略去):

```
# Output
```

```
django-apps
```

这包含我们生成的项目,用于验证是否正确安装了所有内容。

第二步:设置项目

既然我们已经生成了项目的框架,现在就有了一个 settings.py 文件。

在 settings.py 中,可以配置当前的时区,例如,我们将使用 America/New_York(美国/纽约)时间。

进入到 settings.py 文件所在的目录:

```
(env) sammy@ubuntu:~$ cd ~/my_blog_app/blog/blog/
```

然后,使用 nano 或你选择的其他编辑器,打开并编辑 settings.py 文件:

```
(env) sammy@ubuntu:~$ nano settings.py
```

找到打算设置的 TIME_ZONE,请到文件的底部。

我们将修改 TIME_ZONE 行,以便将其设置为当前时区。在 settings.py 中,我们将使用纽约时区(读者也可以设置为北京时间,即 Asia/Shanghai):

让我们保持文件打开,因为需要为静态文件添加一个路径。从 Django 的 Web 项目获得的文件称为静态文件,它们会呈现在 web

页面,包括 JavaScript、CSS 和图像。

转到 settings.py 文件的末尾,添加 STATIC_ROOT,如下所示:

```
...
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/2.0/howto/static-files/
STATIC_URL = '/static/'
STATIC_ROOT = os.path.join(BASE_DIR, 'static')
```

现在我们已经添加了时区和静态文件添的路径,接下来应该将 IP 添加到允许的主机列表中,移到 settings.py 文件中的 ALLOWED_HOSTS 行,在 settings.py 文件的顶部。

```
...
```

SECURITY WARNING: don't run with debug turned on in production!

```
DEBUG = True
```

```
ALLOWED_HOSTS = ['your server IP address']
```

```
# Application definition
```

```
...
```

在方括号内的单引号之间添加服务器的 IP 地址。对所做的更改感到满意后,保存文件。

至此,已经成功编辑了 settings.py 文件,配置了正确的时区,添加了静态文件的路径,并设置了项目的 IP 地址。此时,我们可以继续设置数据库连接。

第三步:配置 MySQL 数据库

为了在我们的项目中使用 MySQL,需要一个与 Django 兼容的 Python3 数据库的连接库。因此,要安装 mysqlclient,它是 MySQLdb 的客户端。

根据 mysqlclient 文档,MySQLdb 是流行的 MySQL 数据库服务器的线程兼容接口,这个服务器提供了 Python 数据库 API,主要的不同之处在于 mysqlclient 具有 Python 3 支持的附加好处。我们需要做的第一件事是安装 python3-dev,你可以通过运行以下命令安装 python3-dev:

```
(env) sammy@ubuntu:~$ sudo apt-get install python3-dev
```

一旦安装了 python3-dev,就可以安装必要的 Python 和 MySQL 开发库:

```
(env) sammy@ubuntu:~$ sudo apt-get install python3-dev libmysqlclient-dev
```

第四步:创建数据库

既然 Django 项目已经设置好,并且 mysqlclient(MySQL 客户端)和 mysql-server(MySQL 服务器)已经安装,我们需要配置 Django 后端以实现 MySQL 兼容性。

第五步:将 MySQL 数据库连接添加到项目中

第六步:测试 MySQL 与项目的连接

我们需要验证 Django 中的配置能否正确地检测到 MySQL 服务器,只要运行这个项目就可以。如果运行失败,则表示没有正确地连接,否则,连接有效。

了解程序设计的基本概念,掌握程序设计的基本方法,熟练掌握 Web 开发语言的基本语法规则。

一、了解 PythonWeb 语言的文件和基本库,掌握各类文件的读写操作。

二、了解计算思维的概念、自顶向下的设计方法及自底向上的执行过程。

三、掌握多种集成开发环境,如 Python 3.6.5、PyCharm 的使用,熟练掌握阅读和分析程序的方法和技巧,熟练掌握程序设计过程和程序编写、调试的方法。

四、针对实际应用问题,了解并初步掌握实用程序的开发方法与技术;能对实际问题进行分析和抽象,并逐步分解,建立项目。