概述

本文基于**Python3**分享urllib模块的源码分享，所以不要拿这python2来问我为什么找不到对应的源码。

在python3中urllib由以下几个模块构成：

- parse
- request
- response
- robotparser
- error

下面对这个几个模块进行一一分享。

parse模块

parse模块定义了统一的接口并实现了URL解析和引用功能。

简单的理解：parse模块可以把url进行拆分或组合，下面我们看下示例：

```
#-*- coding:utf-8 -*-


__author__ = "苦叶子"

from urllib.parse import urlparse


if __name__ == "__main__":
    print("urllib url切割实例")

    url = "http://username:password@www.baidu.com:80/q=开源优测"

    result = urlparse(url)

    print("看下切割后的整体结果: ")
    print(result)

    print("协议: ", result.scheme)
    print("连接字符串: ", result.netloc)
    print("端口号: ", result.port)
    print("uri资源: ", result.path)
    print("用户名: ", result.username)
    print("密码: ", result.password)
```

通过上述实例，我们将学会如何将url中各个属性进行切割出来。

对于parse模块其他的功能，本文就不一一演示了，请参见官网学习。

requset模块

这个模块可以说是urllib最核心的模块了，其定义了系列函数、类用于实现http/https相关协议功能。

下面我们看一个最问简单的应用实例，后续结合实际的API进行深入实例演示：

```
#-*- coding:utf-8 -*-


__author__ = "苦叶子"

import urllib.request

if __name__ == "__main__":
    print("读取www.python.org首页的html源码")

    response = urllib.request.urlopen("http://www.python.org")

    print("打印下结果")

    print(response.read())
```

通过运行上述代码，将会在console看到一堆的html源码的输出显示。

request模块有着非常强大的功能，后续专门开辟一篇文章来分享。

response模块

response模块比较简单，其定义了http response基本出来方法，作为基类存在，大家有兴趣的可以研究下其源码，了解去编码风格及实现，有利于深入掌握如何处理http的返回值。

这里不做实例演示，因为其提供的方法、功能主要在request模块中进行了应用。

robotparser模块

robotparser模块提供了一个单独的类：robotfileparser，用于处理robot.txt文件。

至于这个文件是干嘛用的你可以访问：http://www.robotstxt.org/norobots-rfc.txt 进行了解、学习。

当你需要研究爬虫时，这个robots.txt是必须深入研究的东西。

error模块

error模块定义了url、http相关的错误基类，总共不到100行代码，很简洁，这里就不做说明了。

小结

本文简要的对urllib的组成进行了说明，后续结合实例进行演示分享，大家先通过本文了解下就好

扫一扫关注微信公众号：