**HTTP客户端**

万维网与HTTP

HTTP概述

超文本传输协议（HTTP，HyperText Transfer Protocol）是互联网上应用最为广泛的一种网络协议。所有的www文件都必须遵守这个标准。设计HTTP最初的目的是为了提供一种统一发布和接受HTML页面的方法。

HTTP是一个客户端和服务器端请求和应答的标准。客户端是终端用户，服务器端是网站。通过使用web浏览器、网络爬虫等工具，客户端发起一个到服务器上指定端口（默认是80）的HTTP请求。

HTTP工作过程

客户端发起一个请求，建立一个到服务器指定端口（默认是80）的连接

服务器则在那个端口监听客户端发送过来的请求。一旦接受到请求，服务器（向客户端）发回一个状态行，比如200ok，和（响应的）消息

消息的消息体可能是请求的文件、错误消息、或者其他一些信息

HTTP使用TCP协议

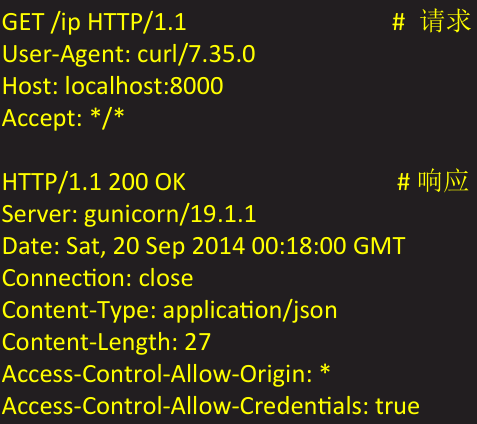
HTTP消息详解

请求和响应

客户端向服务器发送获取文档的请求（request）

一旦发送完请求，客户端就会进行等待，知道从服务器接收到完整的响应（response）为止

当前最流行的HTTP1.4版本的协议中，不允许客户端在尚未收到上一个请求前就在同一套接字上开始发送第二个请求



第一行包含一个方法名喝药请求的文档名；在响应消息中，第一行包含了返回码和秒数信息。无论是在请求和响应消息中，第一行都以回车和换行（CR-LF）结尾

第二部分包含零或多个头信息，每个头信息由一个名称，一个冒号以及一个值组成。HTTP头的名称区分大小写。头信息之后要再跟上一个空行

第三部分是一个可选的消息体。消息体紧跟着头信息后面的空行

HTTP方法

GET和POST这两种方法提供了HTTP基本的“读”和“写”操作

GET请求获取request-URI所标识的资源

POST在request-URI所标识的资源后附加新数据

其余方法可以分为两大类：本质上类似于GET的方法和本质上类似于POST的方法

OPTIONS请求与给定路径匹配的HTTP头的信息

HEAD请求服务器做好一切发送资源的准备，但是只发送头信息

DELETE请求服务器删除request-uri所标识的资源

PUT请求服务器存储一个资源，并用request-uri作为其标识

TRACE请求服务器回送收到的请求信息，主要用于测试或诊断

CONNECT保留将来使用

GET请求

在浏览器的地址栏中输入网址的方式访问网页时，浏览器采用GET方法，现在默认使用的协议版本是1.1



常用的请求报头

METHOD请求资源的方法，这个是必须的

HOST被请求资源的名字，这个是必须的

Accept请求报头域用于指定客户端接收哪些类型的信息

Accept-encoding它是用于指定可接受的内容编码

user-agent客户端信息

connection是否关闭连接

GET响应信息

HTTP/1.1 200 协议、版本和状态码

DATE 日期时间

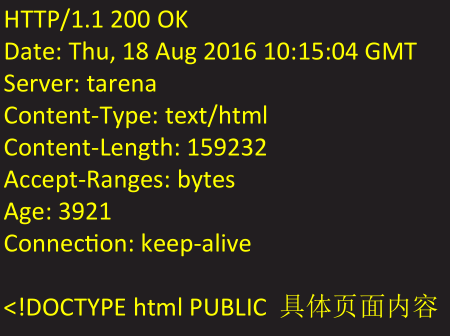
Server 服务器信息

Content-Type 响应内容类型

Content-Length响应数据长度

Last-Modified 资源最后更改时间

Connection 连接方式



POST方法

要求被请求服务器接收附在请求后面的数据

常用于提交表单

一般要在头部声明数据长度

信息头说明参见GET方法

**urllib模块**

urllib基础

urllib简介

在python2版本中，由urllib和urllib2两个库可以用来实现request的发送。而在python3中，已经不存在urllib2这个库了，统一为urllib

urllib中包含了四个模块

urllib.request可以用来发送request和获取request的结果

urllib.error包含了urllib.request产生的异常

urllib.parse用来解析和处理URL

urllib.robotparse用来解析页面的robots.txt文件

爬取网页

先需要导入用到的模块：urllib.request

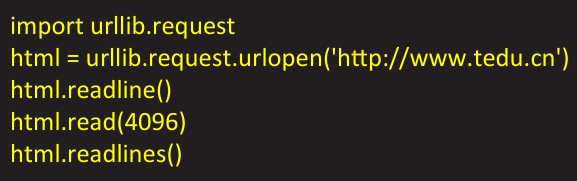
在导入了模块之后，需要使用urllib.request.urlopen打开并爬取一个网页

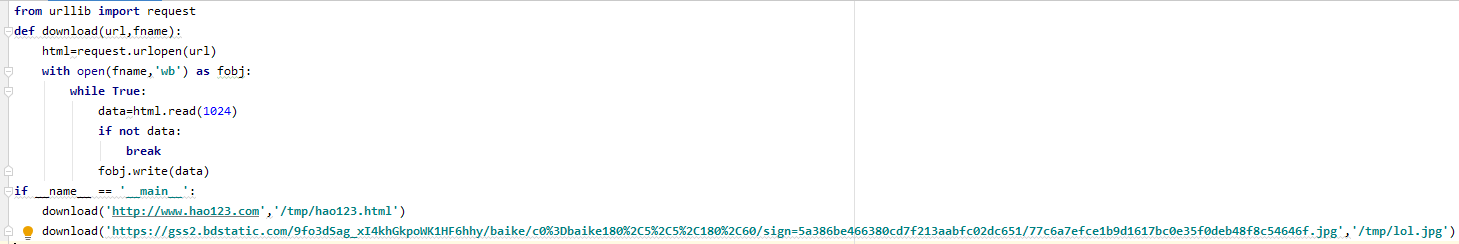
读取内容常见的有三种方式：

read()读取文件的全部内容，与readlines()不同的是，read()会把读取到的内容赋给一个字符串变量

readlines()读取文件的全部内容，readlines()会把读取到的内容赋值给一个列表的变量

readline()读取文件的一行内容





下载网络资源

urllib不仅可以下载网页，其它网络资源均可以下载

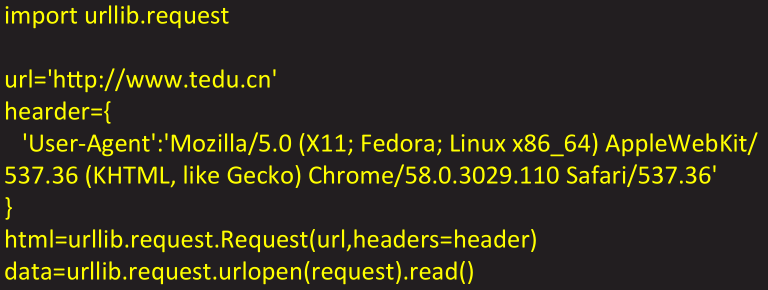
有些文件比较大，需要像读取文件一样，每次读取一部分数据



模拟客户端

有些网页为了防止别人恶意采集其信息所以进行了一些反爬虫的设置，而我们优先进行爬取

可以设置一些Headers信息，模拟成浏览器趣访问这些网站





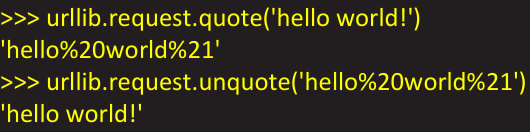
urllib进阶

数据编码

一般来说，URL标准中只会允许一部分ASCII字符，比如数字、字母、部分符号等

而其他的一些字符，比如汉字等，>是不符合URL标准的，此时需要编码。

如果进行编码，可以使用urllib.request.quiote()进行

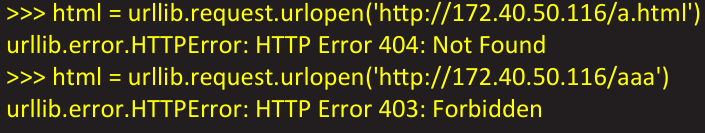




HTTP异常处理

如果访问的页面不存在或者拒绝访问，程序将抛出异常

异常捕获需要导入urllib.error模块





**paramiko模块**

aramiko

安装paramiko模块

本地安装

yum -y install gcc gcc-c++ python-devel

tar -xf paramiko-1.15.4.tar.gz

python setup.py install

网络安装

pip install paramiko

基本使用介绍

SSHClient

创建用于连接ssh服务器的实例



paramiko.AutpAddPolicy

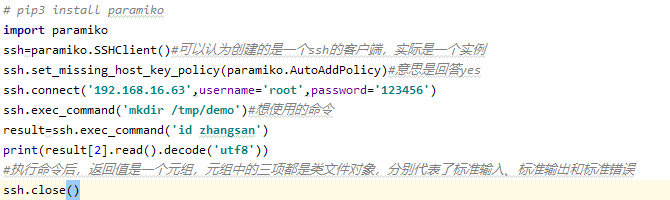
设置自动添加主机秘钥

ssh.connect

连接ssh服务器

ssh.exec\_comand

在ssh服务器上执行指定命令



paramiko实例

编写用于实现ssh访问的脚本

创建SSHClient实例

设置添加主机秘钥策略

连接ssh服务器

执行指定命令

在shell命令行中接受用于连接远程服务器的密码以及在远程主机上执行的命令

