

NSD Devops DAY03

1. [案例1：熟悉HTTP工作流程](#)
2. [案例2：爬取网页](#)
3. [案例3：爬取图片](#)
4. [案例4：处理下载错误](#)
5. [案例5：利用多线程实现ssh并发访问](#)

1 案例1：熟悉HTTP工作流程

1.1 问题

1. 为Firefox安装firebug插件
2. 打开Firefox的firebug或Chrome开发者工具
3. 访问http://www.tedu.cn
4. 在开发者工具的“网络”选项卡中查看请求和响应

1.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：为Firefox安装firebug插件

1)打开Firefox浏览器，点击右上角打开菜单按钮/，打开附加组件，如图-1所示：

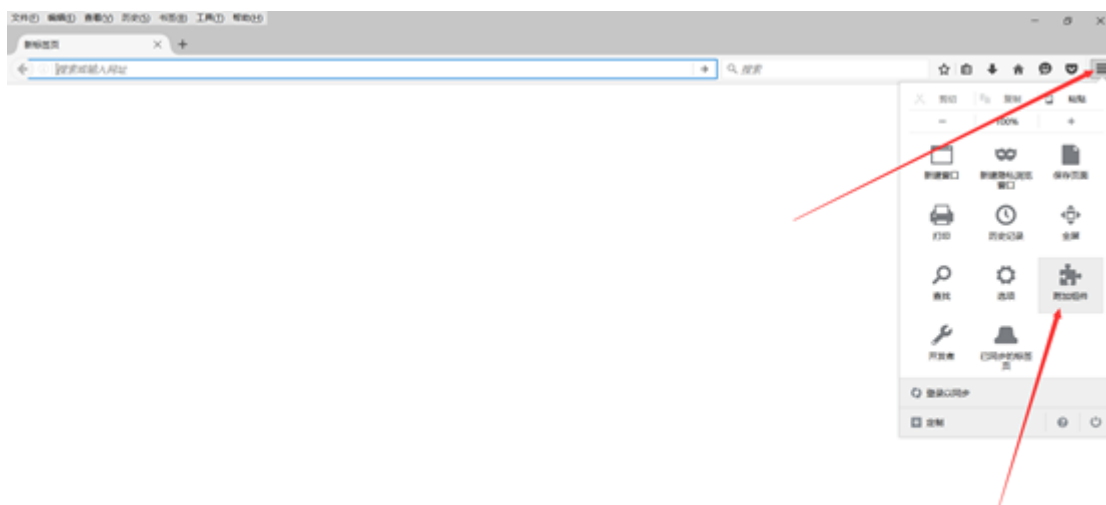


图-1

2)右上角搜索框搜索firebug插件，如图-2所示：

[Top](#)



图-2

3)选定搜索结果安装，如图-3所示：

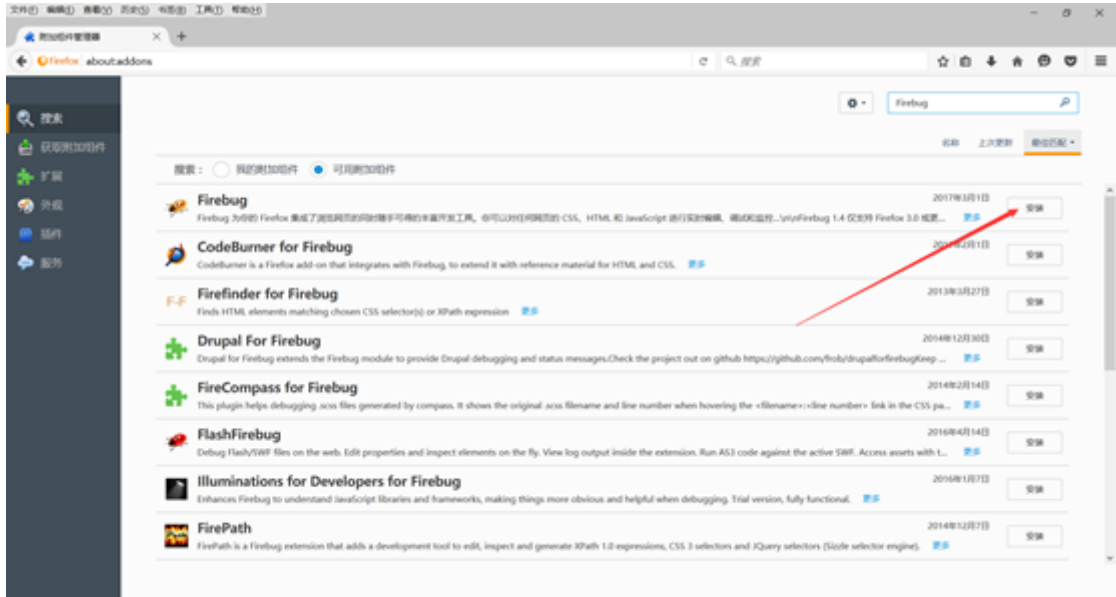


图-3

4)安装成功，如图-4所示：

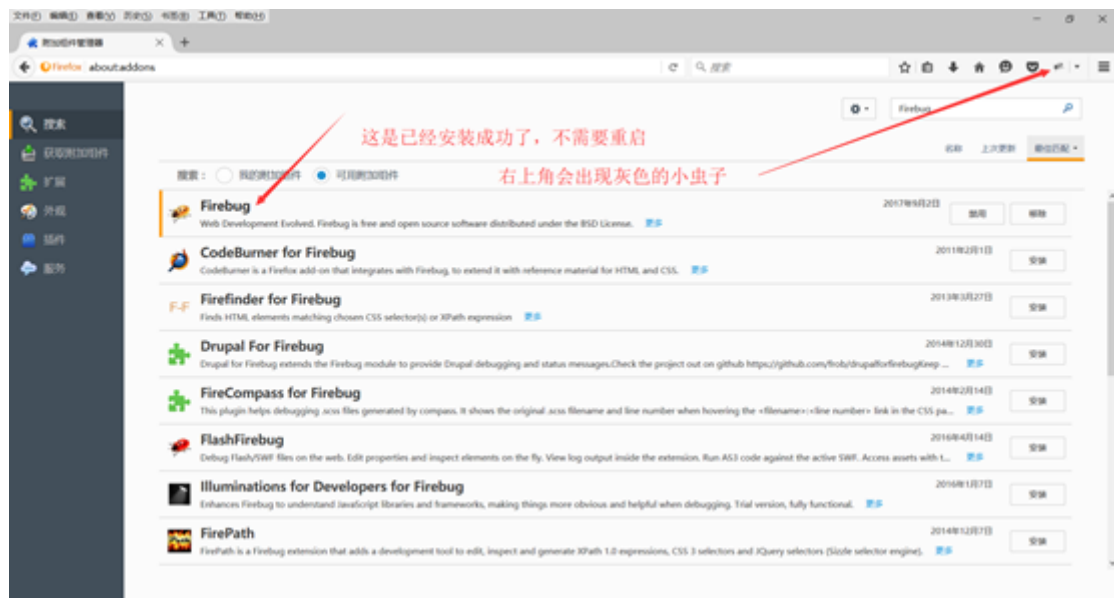


图-4

步骤二：访问<http://www.tedu.cn>

访问<http://www.tedu.cn>，按“F12”打开Firefox的firebug，如图-5所示：

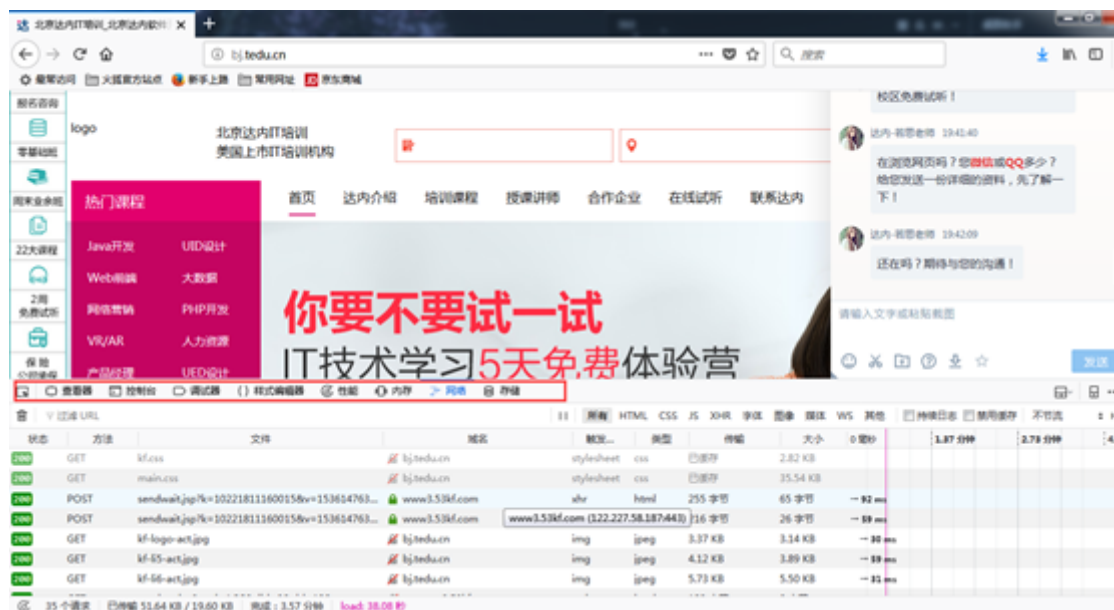


图-5

步骤二：在开发者工具的“网络”选项卡中查看请求和响应

如图-6所示：

[Top](#)



图-6

注意：

常用的请求报头：

METHOD 请求资源的方法，这个是必须的

Host 被请求资源的名子，这个是必须的

Accept 请求报头域用于指定客户端接受哪些类型的信息

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响应消息：

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

2 案例2：爬取网页

2.1 问题

编写一个get_web.py脚本，实现以下功能：

1. 爬取的网页为http://www.tedu.cn
2. 保存的文件名为/tmp/tedu.html

2.2 方案

[Top](#)

导入sys模块，用sys.argv方法获取get_web函数实参，让用户在命令行上提供http://www.tedu.cn和/tmp/tedu.html两个参数，调用get_web函数实现如下功能：

- 1)导入urllib模块，使用urllib模块的urlopen函数打开url（即网址），赋值给html
- 2)以写方式打开/tmp/tedu.html文件
- 3)以循环方式：
读html获取的数据，保存到data
将data写入/tmp/tedu.html
- 4)关闭html

2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：编写脚本

```
01. [ root@localhost day 11] # vim get_web.py
02.  #!/usr/bin/env python3
03.
04.  import sys
05.  from urllib.request import urlopen
06.
07.  def get_web( url, fname ):
08.      html = urlopen( url )    #使用urllib模块的urlopen函数打开url，赋值给html
09.
10.      with open( fname, 'wb' ) as fobj:
11.          while True:
12.              data = html.read( 4096 )
13.              if not data:
14.                  break
15.              fobj.write( data )
16.
17.      html.close()
18.
19.  if __name__ == '__main__':
20.      get_web( sys.argv[ 1 ], sys.argv[ 2 ] )    #让用户在命令行上提供网址和下载数据保存
```

步骤二：测试脚本执行

01. [root@localhost day 11] # python3 get_web.py http://www.tedu.cn /tmp/tedu.html
02. [root@localhost day 11] # cat /tmp/tedu.html
03. [Top](#)
04. 执行cat命令可以看到/tmp/tedu.html文件中爬取到的内容

3 案例3：爬取图片

3.1 问题

1. 将http://www.tedu.cn所有的图片下载到本地
2. 本地的目录为/tmp/images
3. 图片名与网站上图片名保持一致

3.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：编写脚本

1)爬取网页内容放入指定fname (即/tmp/tedu.html) 文件中
创建get_web.py文件，编写代码如下：

```
01. [ root@localhost day 11] # vim get_web.py
02.  #!/usr/bin/env python3
03.
04.  import sys
05.
06.  from urllib.request import urlopen    #导入urllib
07.
08.  def get_web( url, fname):    #url为爬取目标网址 (www.tedu.cn)
09.                               #fname为爬取内容存储文件名
10.      html = urlopen( url)    #使用urllib模块的urlopen函数打开url，赋值给html
11.
12.      with open( fname, 'wb') as fobj:    #以写方式打开文件
13.          while True:
14.              data = html.read( 4096)    #读html获取的数据，保存到data
15.              if not data:
16.                  break
17.              fobj.write( data)    #将data写入文件中
18.
19.      html.close()
```

2)利用正则匹配，将爬取到的fname文件内容中所有图片网址放入result列表
创建get_url.py文件，编写代码如下：

```
01. [ root@localhost day 11] # vim get_url.py
02.  #!/usr/bin/env python3
```

[Top](#)

```

03.
04. import sys
05. import re
06.
07. def get_url( patt, fname):      #patt可匹配图片网址正则表达式，
08.                                #fname为1) 中爬取到内容的文件
09.     cpatt = re.compile( patt)   #将正则表达式字符串形式编译为cpatt实例
10.     result = []                #存放匹配正则表达式的图片网址
11.     with open( fname) as fobj:  #打开爬取到网站 (www.tedu.cn) 内容的文件
12.         for line in fobj:       #遍历fname文件
13.             m = cpatt.search( line)    #使用cpatt实例查找匹配规则的网址
14.             if m:
15.                 result.append( m.group())    #将匹配到的图片网址最加到result列表
16.     return result               #函数最终返回result列表
17.
18. if __name__ == '__main__':
19.     url = r'http://[. \w/- ]+\. (jpg| png| jpeg| gif) ' #符合图片网址规则的正则表达式
20.     print( get_url( url, sys.argv[ 1] ))

```

3)遍历图片列表result，将图片网址对应内容爬取下来存入指定文件
创建download.py文件，编写代码如下：

```

01. [ root@localhost day 11] # vim download.py
02. #! /usr/bin/env python3
03.
04. import os
05. from get_url import get_url    #导入get_url函数
06. from get_web import get_web    #导入get_web函数
07.
08. #调用get_web函数爬取/http://www.tedu.cn网站内容，存入/tmp/tedu.html文件中
09. get_web( 'http://www.tedu.cn/', '/tmp/tedu.html')
10. #符合图片网址正则表达式
11. img_url = r'http://[. \w/- ]+\. (jpg| png| jpeg| gif) '
12. #调用get_url函数，从/tmp/tedu.html文件中获取符合匹配规则的图片网址，
13. #存入result列表中，将列表结果赋值给urls变量
14. urls = get_url( img_url, '/tmp/tedu.html')
15. #爬取到的图片存储目录
16. img_dir = '/tmp/images'
17. #判断目录是否存在，如果不存在则创建该目录
18. if not os.path.exists( img_dir):

```

[Top](#)

```

19.     os.mkdir( img_dir)
20.     #遍历图片网址列表，每次循环遍历出一个图片网址
21.     for url in urls:
22.         # url.split('/')[-1] : 将网址切片，取最后一个字符命名图片
23.         #将图片存储目录与图片名拼接，举例：fname=/tmp/images/XXX.png
24.         fname = os.path.join( img_dir, url.split('/')[-1])
25.         #调用get_web函数，爬取图片网址内容，存入fname文件中
26.         get_web( url, fname)

```

步骤二：测试脚本执行

```

01. [ root@localhost day 11] # python3 download.py
02. [ root@localhost day 11] # nautilus /tmp/images
03.
04.  执行以上命令即可看到爬取的图片，且图片命名与网站上图片命名一致

```

4 案例4：处理下载错误

4.1 问题

1. 启动一个web服务
2. 在web服务器的文档目录下创建目录ban，权限设置为700
3. 编写python程序访问不存在的路径和ban目录，处理404和403错误
4. 404错误打印“无此页面”，403错误打印“无权访问”

4.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：启动一个web服务

```
01. [ root@localhost ~] # systemctl restart httpd
```

步骤二：在web服务器的文档目录下创建目录ban，权限设置为700

```
01. [ root@localhost ~] # mkdir - m 700 /var/www/html/ban
```

步骤三：如果访问的页面不存在或拒绝访问，程序将抛出异常

执行案例2中get_web.py文件，访问不存在页面，抛出404异常如下：

[Top](#)


```

01. [ root@localhost day 11] # python3 get_web.py http://127.0.0.1/abc/ /tmp/abc.html
02. Traceback ( most recent call last ):
03. ...
04. ...
05.     raise HTTPError( req.full_url, code, msg, hdrs, fp)
06. urllib.error.HTTPError: HTTP Error 404: Not Found

```

执行案例2中get_web.py文件，访问存在页面ban目录，抛出403权限异常如下：

```

01. [ root@localhost day 11] # python3 get_web.py http://127.0.0.1/ban/ /tmp/abc.html
02. Traceback ( most recent call last ):
03. ...
04. ...
05.     raise HTTPError( req.full_url, code, msg, hdrs, fp)
06. urllib.error.HTTPError: HTTP Error 403: Forbidden

```

步骤三：编写python程序捕获异常

创建get_web3.py文件，实现访问不存在的路径和ban目录时，捕获404和403错误，同时404错误打印“无此页面”，403错误打印“无权访问”，代码如下：

```

01. import sys
02. from urllib.request import urlopen
03. from urllib.error import HTTPError #导入urllib.error模块，用HTTPError捕获异常信息
04.
05. def get_web( url, fname ):
06.     try:
07.         html = urlopen( url ) #打开网址时即可知道是否有异常，所以将本语句放入try语句
08.     except HTTPError as e: #捕获返回HTTPError类的实例e
09.         print( e )
10.         if e.code == 403: #捕获异常状态码如果等于403
11.             print( '权限不足' ) #输出'权限不足'
12.         elif e.code == 404: #捕获异常状态码如果等于404
13.             print( '没有那个地址' ) #输出'没有那个地址'
14.         return #return后面代码均不执行
15.
16.     with open( fname, 'wb' ) as fobj:
17.         while True:
18.             data = html.read( 4096)

```

[Top](#)

```

19.         if not data:
20.             break
21.         fobj.write( data)
22.
23.     html.close()
24.
25. if __name__ == '__main__':
26.     get_web( sys.s.argv[ 1], sys.s.argv[ 2])

```

测试脚本执行：

访问不存在页面：

```

01. [ root@localhost day 11] # python3 get_web.py http://127.0.0.1/abc/ /tmp/abc.html
02. HTTP Error 404 : Not Found
03. 没有那个地址

```

访问ban目录：

```

01. [ root@localhost day 11] # python3 get_web.py http://127.0.0.1/ban/ /tmp/abc.html
02. HTTP Error 403 : Forbidden
03. 权限不足

```

5 案例5：利用多线程实现ssh并发访问

5.1 问题

编写一个remote_comm.py脚本，实现以下功能：

1. 在文件中取出所有远程主机IP地址
2. 在shell命令行中接受远程服务器IP地址文件、远程服务器密码以及在远程主机上执行的命令
3. 通过多线程实现在所有的远程服务器上并发执行命令

5.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：安装paramiko

paramiko 遵循SSH2协议，支持以加密和认证的方式，进行远程服务器的连接，可以实现远程文件的上传，下载或通过ssh远程执行命令。

[Top](#)

```

01. [ root@localhost ~] # pip3 install paramiko

```

- 02.
03. ...
04. ...
05. Successfully installed bcrypt- 3.1.4 paramiko- 2.4.1 pyasn1- 0.4.4 py nacl- 1.2.1
06. You are using pip version 9.0.1, however version 18.0 is available.
07. You should consider upgrading via the 'pip install -- upgrade pip' command.

测试是否安装成功

01. >>> import paramiko
02. >>>

步骤二：编写脚本

01. [root@localhost day 11] # vim remote_comm.py
02. #!/usr/bin/env python3
- 03.
04. import sys
05. import getpass
06. import paramiko
07. import threading
08. import os
- 09.
10. #创建函数实现远程连接主机、服务器密码以及在远程主机上执行的命令的功能
11. def remote_comm(host, pwd, command):
12. #创建用于连接ssh服务器的实例
13. ssh = paramiko.SSHClient()
14. #设置自动添加主机密钥
15. ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
16. #连接ssh服务器，添加连接的主机、用户名、密码填好
17. ssh.connect(hostname=host, username='root', password=pwd)
18. #在ssh服务器上执行指定命令，返回3项类文件对象，分别是，输入、输出、错误
19. stdin, stdout, stderr = ssh.exec_command(command)
20. #读取输出
21. out = stdout.read()
22. #读取错误
23. error = stderr.read()
24. #如果有输出
25. if out:

[Top](#)

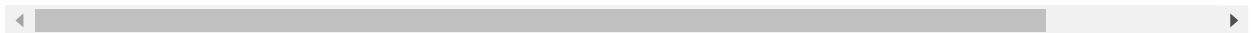
```
26. #打印主机输出内容
27.     print( '[ %s] OUT: \n%s' %( host, out.decode( 'utf8' ) ) )
28. #如果有错误
29.     if error:
30. #打印主机错误信息
31.         print( '[ %s] ERROR: \n%s' %( host, error.decode( 'utf8' ) ) )
32. #程序结束
33.     ssh.close()
34.
35. if __name__ == '__main__':
36. #设定sys.argv长度，确保remote_comm函数中参数数量
37.     if len( sys.argv ) != 3:
38.         print( 'Usage: %s ipaddr_file "command"' % sys.argv[ 0 ] )
39.         exit( 1 )
40. #判断命令行上输入如果不是文件，确保输入的是文件
41.     if not os.path.isfile( sys.argv[ 1 ] ):
42.         print( 'No such file:', sys.argv[ 1 ] )
43.         exit( 2 )
44. #fname为存储远程主机ip的文件，用sys.argv方法，可以在执行脚本时再输入文件名，更
45.     fname = sys.argv[ 1 ]
46. #command为在远程主机上执行的命令，用sys.argv方法，可以在执行脚本时再输入相应
47.     command = sys.argv[ 2 ]
48. #通过getpass输入远程服务器密码，pwd为remote_comm函数第二个参数
49.     pwd = getpass.getpass()
50. #打开存有远程主机ip的文件
51.     with open( fname ) as fobj:
52. #将遍历文件将ip以列表形式存入ips，line.strip()可以去掉每行ip后\n
53.         ips = [ line.strip() for line in fobj ]
54. #循环遍历列表，获取ip地址，ip为remote_comm函数第一个参数
55.         for ip in ips:
56. #将读取到的ip地址作为remote_comm函数实际参数传递给函数，ips中有几个ip地址循环。
57. #创建多线程
58.             t = threading.Thread( target=remote_comm, args=( ip, pwd, command ) )
59. #启用多线程
60.             t.start()
```

步骤三：测试脚本执行

[Top](#)

01. #参数给少了效果如下：

02. [root@localhost day 11] # python3 remote_comm.py server_addr.txt
03. Usage: remote_comm.py ipaddr_file " command"
04. #参数给多了效果如下 :
05. [root@localhost day 11] # python3 remote_comm.py server_addr.txt id zhangsan
06. Usage: remote_comm.py ipaddr_file " command"
07. #正常显示如下 :
08. [root@localhost day 11] # python3 remote_comm.py server_addr.txt " id zhangsan"
09. Password:
10. [192.168.4.2] OUT:
11. uid=1001(zhangsan) gid=1001(zhangsan) 组=1001(zhangsan)
12. [192.168.4.3] OUT:
13. uid=1001(zhangsan) gid=1001(zhangsan) 组=1001(zhangsan)
14. [root@localhost day 11] # python3 remote_comm.py server_addr.txt " echo redhat | pass
15. Password:
16. [192.168.4.3] OUT:
17. 更改用户root的密码 :
18. passwd : 所有的身份验证令牌已经成功更新。
19. [192.168.4.2] OUT:
20. 更改用户root的密码 :
21. passwd : 所有的身份验证令牌已经成功更新。
22. #此时密码已经变成redhat
23. [root@localhost day 11] # python3 remote_comm.py server_addr.txt " id zhangsan"
24. Password:
25. [192.168.4.2] OUT:
26. uid=1001(zhangsan) gid=1001(zhangsan) 组=1001(zhangsan)
27. [192.168.4.3] OUT:
28. uid=1001(zhangsan) gid=1001(zhangsan) 组=1001(zhangsan)

[Top](#)