网络数据采集与爬虫

陈华珊 (中国社科院社会发展战略研究院)

网络数据获取方式

- 模拟鼠标点击
  - 按键精灵
  - `sikulix`
- 调用浏览器进行自动操作
  - IE：Document Object Model (DOM)
  - Chrome, Firefox
    - `rdom`
    - `RSelenium`
    - headless
- HTTP 采集

HTML基础知识

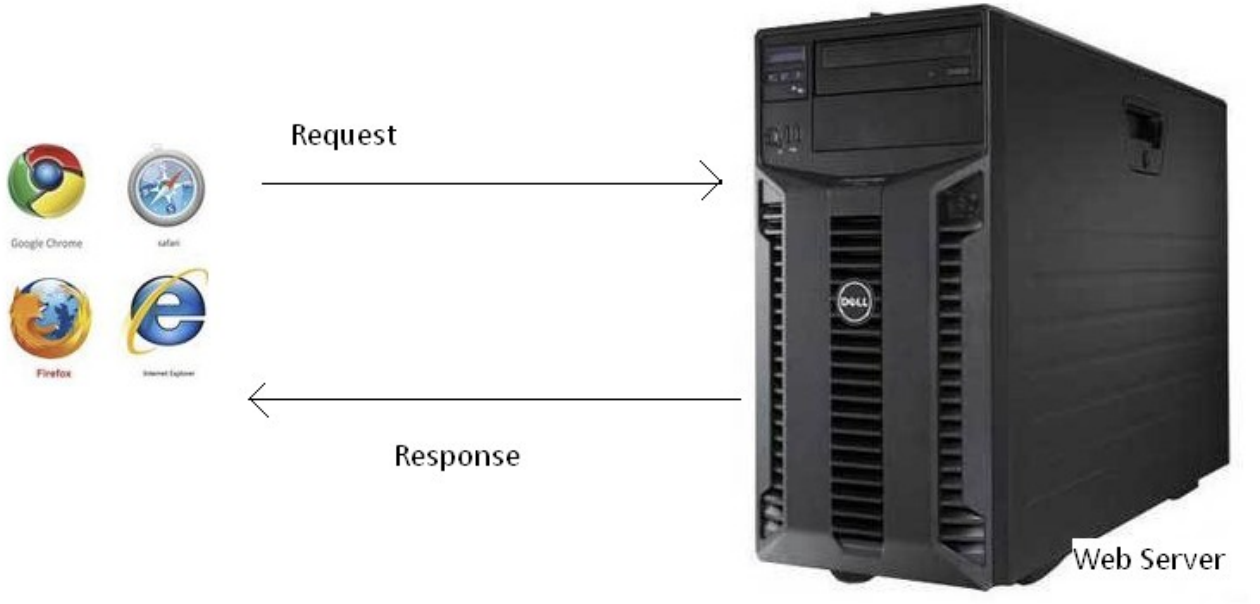
```
1  <html>
2  <meta>title, description, keywords</meta>
3  <body>
4  <div>...</div>
5  <p> 正文 <a href='http://www.google.com'>Google</a> </p>
6  <table>表格</table>
7  </body>
8  </html>
```

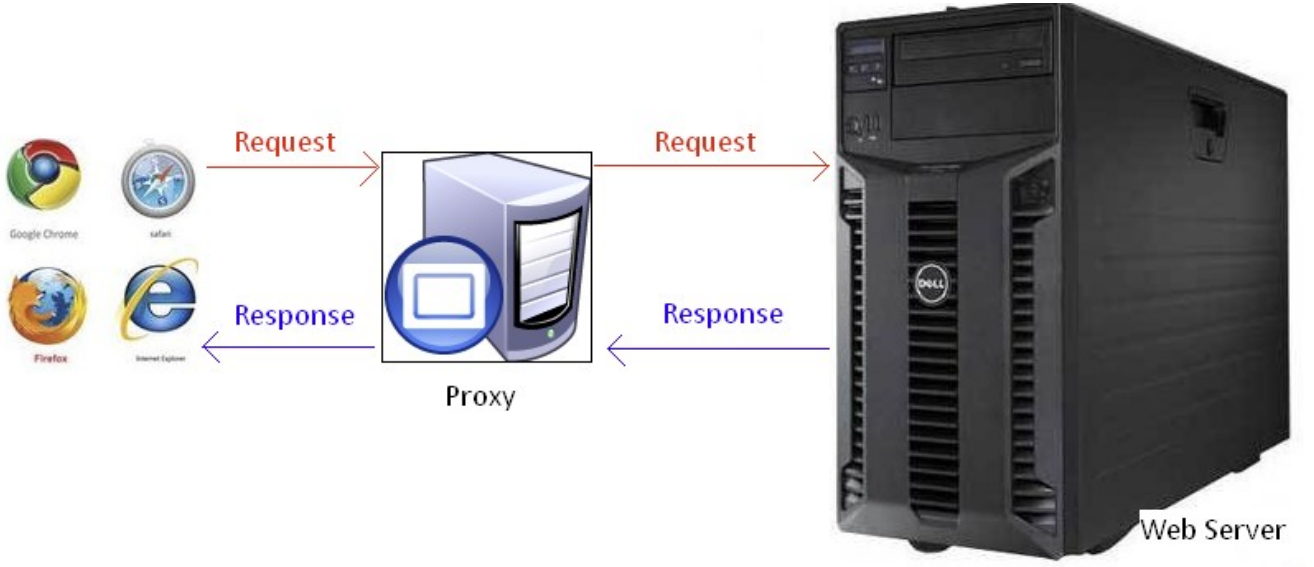
HTTP基础知识

HTTP protocal (超文本传输协议)

- 计算机通信网络中两台计算机之间进行通信所必须共同遵守的规定或规则，超文本传输协议(HTTP)是一种通信协议，它允许将超文本标记语言(HTML)文档从Web服务器传送到客户端的浏览器。
- 目前我们使用的是HTTP/1.1 版本
- HTTP/2 is coming

Web服务器，浏览器，代理服务器




Request
Request
Response
Response
Proxy
Web Server

URL

URL(Uniform Resource Locator) 地址用于描述一个网络上的资源，基本格式如下

```
schema://host[:port#]/path/.../[;url-params][?query-string][#anchor]
```

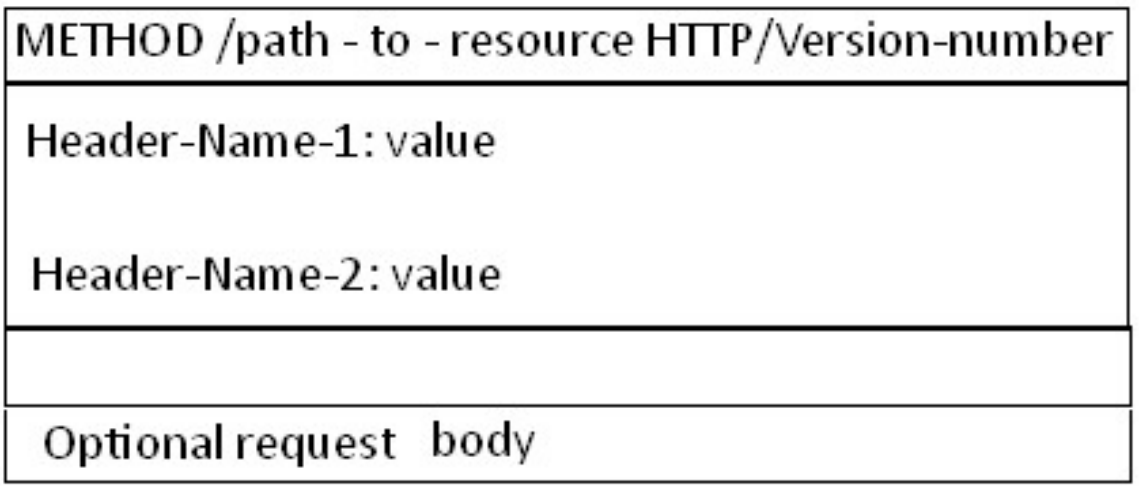
Name	Desc.
scheme	指定低层使用的协议(例如：http, https, ftp)
host	HTTP服务器的IP地址或者域名
port#	HTTP服务器的默认端口是80，这种情况下端口号可以省略。如果使用了别的端口，必须指明，例如 http://www.cnblogs.com:8080/
path	访问资源的路径
url-params	
query-string	发送给http服务器的数据
anchor-	锚

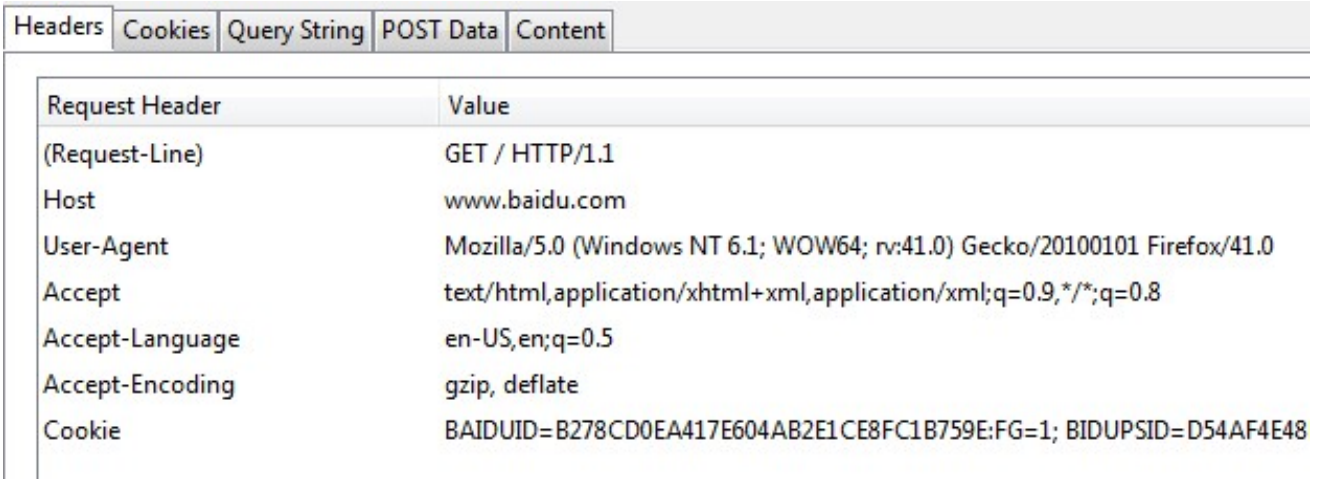
URL 的一个例子

```
http://www.mywebsite.com/sj/test;id=8079?name=sviergn&x=true#stuff

Schema: http

host: www.mywebsite.com

path: /sj/test

URL params: id=8079

Query String: name=sviergn&x=true

Anchor: stuff
```

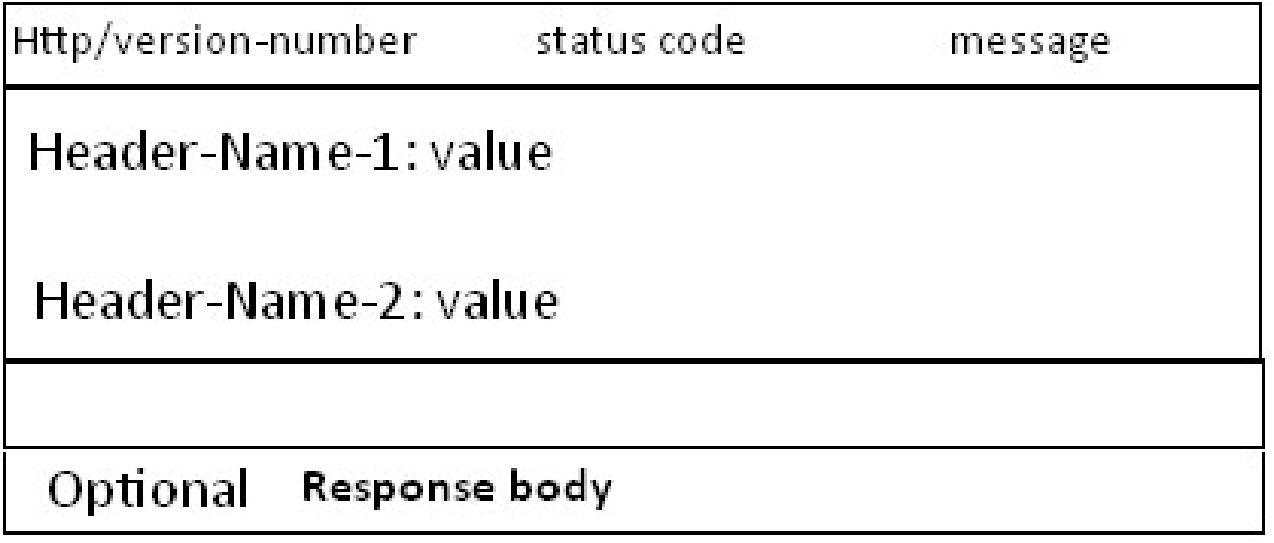
HTTP消息的结构

- Request
  - 请求行
  - http header
  - body




Headers
Cookies
Query String
POST Data
Content
Request Header
Value
(Request-Line)
GET / HTTP/1.1
Host
www.baidu.com
User-Agent
Mozilla/5.0 (Windows NT 6.1; WOW64; rv:41.0) Gecko/20100101 Firefox/41.0
Accept
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language
en-US,en;q=0.5
Accept-Encoding
gzip, deflate
Cookie
BAIDUID=B278CD0EA417E604AB2E1CE8FC1B759E:FG=1; BIDUPSID=D54AF4E48

- Response
  - 请求行
  - request header
  - body



Method

与服务器交互的方法，最基本的有4种，分别是GET,POST,PUT,DELETE;

- GET vs. POST
  - `GET` 提交的数据会放在URL之后，以? 分割URL和传输数据，参数之间以 `&` 相连，如

    ```
    EditPosts.aspx?name=test1&id=123456
    ```

  - `POST` 方法是把提交的数据放在HTTP包的Body中;
  - `GET` 提交的数据大小有限制（因为浏览器对URL的长度有限制）; `POST` 方法提交的数据没有限制;

状态码（Status）

Response 消息中的第一行叫做状态行，由HTTP协议版本号，状态码，状态消息三部分组成。

HTTP/1.1中定义了5类状态码，状态码由三位数字组成，第一个数字定义了响应的类别

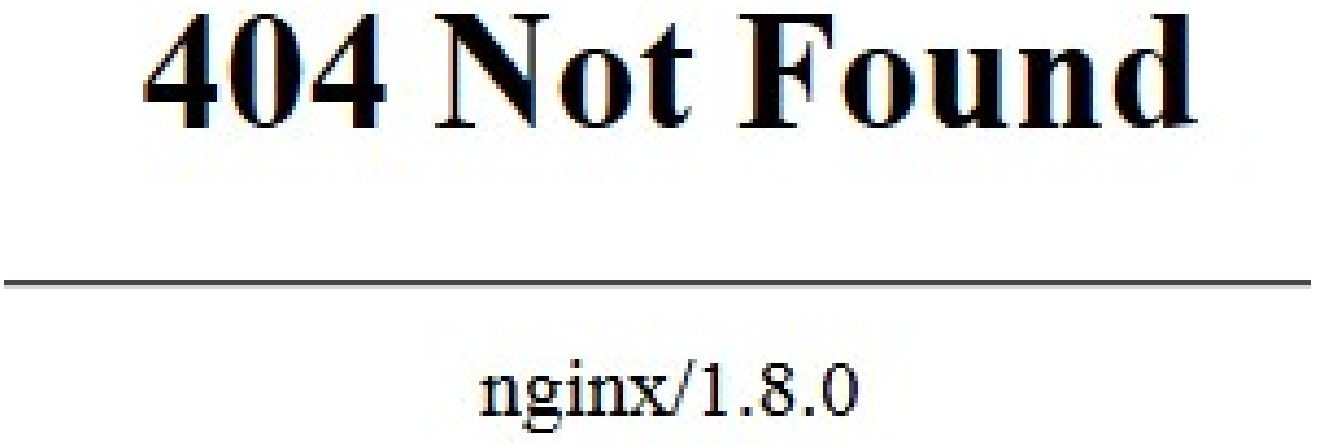
+ 1XX 提示信息 - 表示请求已被成功接收，继续处理

+ 2XX 成功 - 表示请求已被成功接收，理解，接受

+ 3XX 重定向 - 要完成请求必须进行更进一步的处理

+ 4XX 客户端错误 - 请求有语法错误或请求无法实现

+ 5XX 服务器端错误 - 服务器未能实现合法的请求


404 Not Found
nginx/1.8.0

很抱歉，您要访问的页面不存在！

温馨提示：

1. 请检查您访问的网址是否正确
2. 如果您不能确认访问的网址，请浏览百度更多页面查看更多网址。
3. 回到顶部重新发起搜索
4. 如有任何意见或建议，请及时反馈给我们。

连接Web服务

- `WDI`, `wbstats`: World Bank Data
- `Rfacebook`
- `Rflickr`
- `Rlinkedin`
- `tumblR`
- `twitteR`
- Google Trends: `GTrendsR`
- Google Analytics: `RGoogleAnalytics`, `ganalytics`, `GAR`, `RGA`

一个完整的数据采集流程

- 下载
- 网页解析
- 检索与存储
  - `mysql,sqlite`
  - `nosql: mango`
  - `hadoop, spark`
- 动态脚本交互

下载

R包

处理 http 请求（GET, POST, PUT, HEAD, DELETE, etc.）及登录校验。

- `download.file()`, `downloader::download()` for SSL
- `source_Dropbox()` ,`source_XlsxData()`
- `RCurl`
- `httr`
- `request`
- `rvest` + `magrittr`
- `scrapeit.core`

登录验证

- 简单的用户名密码参数提交可用:
  - api key: `http://api.foo.org/?key=yourkey;`
  - user/pass: `http://username:password@api.foo.org`
- OAuth 1.0(linkedin, twitter, vimeo)，OAuth 2.0 (facebook, GitHub, google)

— `httr`, `ROAuth`, `googleAuthR`

URL 编码解码

- `urltools`
- `httr::parse_url()`
- `utils::URLdecode()`,`URLencode()`
- 域名提取 `tldextract`

参数校验

- Referer:
- User-Agent:
- Request-type: GET, POST, ...
- 其它动态生成的参数
- Cookie 伪装
- 代理调度

结构化的网页解析

- HTML/XHTML, XML/XML2, JSON, YAML
- XML/HTML: `XPath` 查询

  `XML`, `xml2`, `selectr`

- JSON: `jsonlite`, `rsjon`, `RJSONIO`
- YAML: `yaml`

示例

```
library(xml2)

html = read_html('http://to.xml')
html

x = xml_find_all(html, xpath = '//*[(@id = "something")]')
xml_attrs(x)
```

如何选择网页元素?

```
<div id="NodeValueDiv3" class="TimeHide">
  <ul>
    <li><a id="NodeValueDiv3ReferType1Link" title="反映本文研究工
作的背景和依据">参考文献</a>
        <span id="NodeValueDiv3ReferType1Level1"></span></li>
    <li><a id="NodeValueDiv3ReferType2Link" title="本文参考文献的
参考文献。进一步反映本文研究工作的
背景和依据">二级参考文献</a>
        <span id="NodeValueDiv3ReferType2Level2"></span></li>
    <li><a id="NodeValueDiv3ReferType4Link" title="引用本文的文
献。本文研究工作的继续、应用、发展或评价">
引证文献</a><span id="NodeValueDiv3ReferType4Level1"></span></li>
    <li><a id="NodeValueDiv3ReferType16Link" title="本文引证文献的
引证文献。更进一步反映本文研究工作
的继续、发展或评价">二级引证文献</a>
        <span id="NodeValueDiv3ReferType16Level2"></span></li>
  </ul>
```

```
18      </div>
```

- by id (“NodeValueDiv3”)
- by tag name
- by css class (“TimeHide”)
- by attribute (“title”)

CSS selector vs. XPath

CSS selector 比 XPath 更易读

- `#`: 选择 `id`，例如 `#xyz`
- `mytagname`: 选择 `tag`，例如 `a`
- “`.`” : 选择 class，例如 `.xyz`
- `[target]`: 选择属性，例如 `[alt]`

http://www.w3schools.com/cssref/css_selectors.asp

```
1  library(selectr)
2  css_to_xpath('#hpKKK')
3  css_to_xpath('a')
4  css_to_xpath('.xxx')
5  css_to_xpath('[src]')
```

http://selectorgadget.com/

rvest 解析网页

- `html_nodes`: 基于 css 或 xpath 选择节点
- `html_attrs`: 获取全部属性
- `html_attr`: 获取指定节点的某个属性
- `html_name`: 获取节点的 tag
- `html_text`: 获取节点的内容
- `html_children`: 获取子节点

非结构化的网页解析

- 网页正文提取 `boilerpipeR`
- 文本查询和匹配

检索与存储

- 一般关系型数据库: `mysql`, `sqlite`,`sql server`, `oracle`
- NoSQL: `Mongo DB`, `Cassandra`, `HBase`
- Hadoop, `spark`
- Key-Value: `Redis`, `memcached`

动态脚本交互

- `RSelenium`
- `V8`
- `SpiderMonkey`

R 调用 HTTP 解析

包的加载及调用

```
1  library(httr)
2
3  r <- GET("http://www.baidu.com")
4  ## Error in curl:    :curl_fetch_memory(url, handle = handle)    :
5  Couldn't resolve host name
```

查看 response 对象，有用的信息包括：真实的URL（URL跳转）、HTTP状态码、内容类型、大小、etc.

```
r
## Error in eval(expr, envir, enclos): object 'r' not found
```

继续挖掘

```
status_code(r)
## Error in status_code(r): object 'r' not found
headers(r)
## Error in headers(r): object 'r' not found
str(content(r))
## Error in inherits(x, "response"): object 'r' not found
```

Body

- 文本内容

```
content(r, 'text')
content(r, "text", encoding = "ISO-8859-1")
```

- 非文本内容

```
content(r, "raw")
```

```
2 writeBin(r, 'aaa.jpg')
```

- 数据

JSON automatically parsed into named list

```
1 content(r, "parsed")
```

Cookie

```
r$cookies
## Error in eval(expr, envir, enclos): object 'r' not found
```

传输 cookie

```
r <- GET("http://httpbin.org/cookies/set", query = list(a = 1))
cookies(r)
```

传递参数

```
r <- GET("http://httpbin.org/get",
  query = list(key1 = "value1", key2 = "value2")
)
content(r)$args

r <- POST("http://httpbin.org/post", body = list(a = 1, b = 2, c
= 3))

# 上传文件
POST(url, body = upload_file("mypath.txt"))
```

抓包与解析

- 浏览器调试
  - 查看网页源代码
  - Firefox: `firebug`
  - Chrome: debug模式 F12
- 网络嗅探、HTTP 抓包工具
  - sniffer
  - `httpfox`
  - `fiddler`

爬虫实践