爬虫入门

王浩然

清华大学计算机科学与技术系

2024 年 7 月 22 日

1. 爬虫简介
2. 基础知识回顾
3. 实战
4. 参考资料

1 爬虫简介

2 基础知识回顾

3 实战

4 参考资料

爬虫简介

爬虫是一个自动化程序，它会浏览互联网，就像你在浏览网页一样。想象一下，你在网上查找信息时，可能会打开一个网页，点击一些链接，阅读内容，再点击更多的链接，继续浏览其他网页。爬虫做的事情和你很相似，只不过它是自动进行的，而且速度快得多。

具体来说，爬虫会做什么？

- **访问一个网页**：爬虫会像你输入网址并按下回车键一样，访问一个特定的网页。
- **读取网页内容**：它会“阅读”网页上的所有文字、图片、链接等内容，就像你看网页一样。
- **跟随链接**：如果网页上有链接，爬虫可以点击这些链接，访问新的网页，然后重复这个过程。

几种爬虫写法

- **Python**
  - `requests`：用于发送 HTTP 请求
  - `beautifulSoup`：用于解析 HTML 和 XML
  - `scrapy`：一个强大的爬虫框架
- **JavaScript**
  - `axios`：用于发送 HTTP 请求
  - `cheerio`：用于解析 HTML
  - `puppeteer`：用于无头浏览器操作
- **Java**: `Jsoup`, `HttpClient`, `Selenium`
- **Ruby**: `Nokogiri`, `HTTParty`, `Watir`
- **Go**: `colly`, `goquery`, `http`

是不是很慌?

~~爬虫好乱，为什么有这么多种写法啊啊啊啊啊啊！~~

原理是共通的，无非 · · ·

- 获得目标网页源码模板（你要爬的是一类有共性网页，对吧 https://www.bilibili.com/video/
- 解析你想要的部分，比如标题、正文、图片链接等（一般都有固定格式）
- 摘取或访问这些部分，把相应数据存下来

1 爬虫简介

2 基础知识回顾

3 实战

4 参考资料

HTML

HTML 是用来描述网页的一种语言。HTML 指的是超文本标记语言：HyperText Markup Language。HTML 不是一种编程语言，而是一种标记语言。标记语言是一套标记标签 (markup tag)。HTML 使用标记标签来描述网页。HTML 文档包含了 HTML 标签及文本内容。HTML 文档也叫做网页。HTML 是一种特殊的

XML，它使用标签来描述文档的结构和内容。但对各标签进行了规范。HTML 标签通常是成对出现的，比如 和 。第一个标签是开始标签，第二个标签是结束标签。

常见 HTML 标签

- <title>：定义文档的标题
- <body>：定义文档的主体
- <h1> 到 <h6>：定义标题
- <p>：定义段落
-
：定义换行
- <a>：定义链接
- <img>：定义图像
- ：定义无序列表
- ：定义有序列表
- ：定义列表项
- <img>：定义图像
- <a>：定义链接
- <td>：定义表格数据

HTML ID、属性、类

我们不能只靠一堆标签来描述期望得到的网页，我们还需要额外的信息，比如我想要蓝色的标题和红色的标题。属性用来指导浏览器渲染，比如 `<p style="color: red; font-size: 20px;">`。类和 ID 用来区分不同标签，它们的区别在于，ID 是唯一的，而类是可以重复的。

CSS 选择器

用于选取 HTML 元素并对其应用样式。选择器可以基于元素的 **element**、**id**、**class**、**attr** 等来选取特定的元素。了解选择器的使用方法是掌握 CSS 的基础。

基本选择器

```
1 // 元素选择器：选取指定标签的所有元素
2 p
3 // 类选择器：选取指定类的所有元素，类名前加 '.'
4 .classname
5 // ID 选择器：选取指定唯一 ID 的元素，ID 前加 '#'
      （不唯一？）
6 #uniqueId
7 // 通配选择器：选取所有元素。
8 *
```

组合选择器

```
1 // 后代选择器：选取某元素内的所有指定后代元素。
2 div p
3 // 子选择器：选取某元素的所有指定直接子元素。
4 div > p
5 // 相邻兄弟选择器：选取紧接在某元素后的指定兄弟元
      素。
6 h1 + p
7 // 通用兄弟选择器：选取某元素后面的所有指定兄弟元
      素。
8 h1 ~ p
```

伪类选择器

```
1 // 链接伪类：选取链接的不同状态。
2 a:link
3 a:visited
4 a:hover
5 a:active
```

结构伪类

```
// 结构伪类：选取特定的子元素或基于位置的元素。
p:first-child
p:last-child
p:nth-child(2)
p:nth-of-type(odd)
```

其他伪类

```
1 // 其他伪类:
2 input:focus
3 p:empty
```

伪元素选择器

```
1 // ::before 和 ::after: 在元素内容之前或之后插入内
      容。
2 p::before
3 p::after
```

属性选择器

```
1 // 存在属性选择器：选取包含指定属性的元素。
2 [title]
3 // 等于属性选择器：选取属性值等于指定值的元素。
4 [type="text"]
5 // 包含词属性选择器：选取属性值包含指定词的元素。
6 [class~="example"]
7 // 开头匹配属性选择器：选取属性值以指定值开头的元
      素。
8 [class^="prefix"]
9 // 结尾匹配属性选择器：选取属性值以指定值结尾的元
      素。
10 [class$="suffix"]
11 // 包含子串属性选择器：选取属性值包含指定子串的元
      素。
12 [class*="substring"]
```

组合选择器示例

```
1 // 选取所有 class 为 "example" 的 p 元素
2 p.example
3 // 选取所有 class 为 "example" 的 p 元素，并且它们
      是 div 的直接子元素
4 div > p.example
5 // 选取所有 class 为 "example" 的 p 元素，并且紧跟
      在 h1 元素后
6 h1 + p.example
7 // 选取所有 class 为 "example" 的 p 元素，并且是
      div 的后代
8 div p.example
```

CSS 选择器总结

- 基本选择器：元素、类、ID、通配
- 组合选择器：后代、子、相邻兄弟、通用兄弟
- 伪类选择器：链接、结构、其他
- 伪元素选择器：before、after、first-line、first-letter
- 属性选择器：存在、等于、包含词、开头匹配、结尾匹配、包含子串
- 组合选择器：组合多种选择器
- 通过掌握这些选择器，可以更灵活和高效地对 HTML 元素进行样式化，从而实现复杂的页面布局和设计。

XPath 简介

XPath 是一门在 XML 文档中查找信息的语言。XPath 用于在 XML 文档中通过元素和属性进行导航。XPath 是由 W3C 组织制定的标准，可用来在 XML 文档中对元素和属性进行遍历。

基本语法

- //book 选择文档中所有的 <book> 元素，而不考虑它们的位置。
- /bookstore/book 这个表达式选择根节点 <bookstore> 下的所有 <book> 元素。
- /bookstore/book[@category] 选择 <bookstore> 下带有 category 属性的 <book> 元素。

基本语法

- `/bookstore/book[@category='cooking']` 这个表达式选择 <bookstore> 下 category 属性值为 cooking 的 <book> 元素。
- `//*` 选择文档中的所有元素。
- `/bookstore/book/title` 选择 <bookstore> 下每个 <book> 元素的 <title> 子元素。

函数使用

- string-length(/bookstore/book/title) 这个函数返回 <title> 元素中字符串的长度。
- sum(/bookstore/book/price) 这个函数返回 <price> 元素值的总和。
- contains(/bookstore/book/title, 'Cook') 这个函数检查 <title> 元素的值是否包含字符串 Cook。

一个实例

- `/bookstore/book/title` 选择所有 <title> 元素。
- `//title[@lang='en']` 选择所有 lang 属性值为 en 的 <title> 元素。
- `/bookstore/book[price>30.00]` 选择所有 price 大于 30.00 的 <book> 元素。
- `/bookstore/book/title/text()` 选择所有 <title> 元素的文本内容。

一个实例

- /bookstore/book/title 选择所有 <title> 元素。
- //title[@lang='en'] 选择所有 lang 属性值为 en 的 <title> 元素。
- /bookstore/book[price>30.00] 选择所有 price 大于 30.00 的 <book> 元素。
- /bookstore/book/title/text() 选择所有 <title> 元素的文本内容。

```
<bookstore>
    <book category="cooking">
        <title lang="en">Everyday Italian</title>
        <author>Giada De Laurentiis</author>
        <year>2005</year>
        <price>30.00</price>
    </book>
    <book category="children">
        <title lang="en">Harry Potter</title>
```

综合实战

```
1 //div[contains(@class, 'example-class')]/a/@href
2
3 //table/tr[1]/td[string-length(text()) > 0]
4
5 //div[starts-with(@id, 'item-')]/span/text()
```

爬虫还能分类

1. 服务端渲染：页面结果由服务端渲染后返回，有效信息直接在 HTML 里面
2. 客户端渲染：页面主要内容由后端渲染得到，真实数据通过 ajax 接口获取

对于客户端渲染

两种方法:

- 逆向分析网页，手动分析 Network 中的 ajax 请求，发 POST 请求拿到 json 数据
- 使用 Selenium 库模拟动态网页动作，直接从浏览器中收集数据

robot.txt

robots.txt 是一个存储在网站根目录下的文本文件，它告诉网络爬虫哪些页面可以抓取，哪些页面不能抓取。

- User-agent: *：表示该规则适用于所有爬虫
- Disallow: /private/：表示禁止爬虫访问 private 目录
- Allow: /public/：表示允许爬虫访问 public 目录
- Sitemap: http://www.example.com/sitemap.xml：表示网站地图的地址

`https://www.cloudflare.com/robots.txt`

1 爬虫简介

2 基础知识回顾

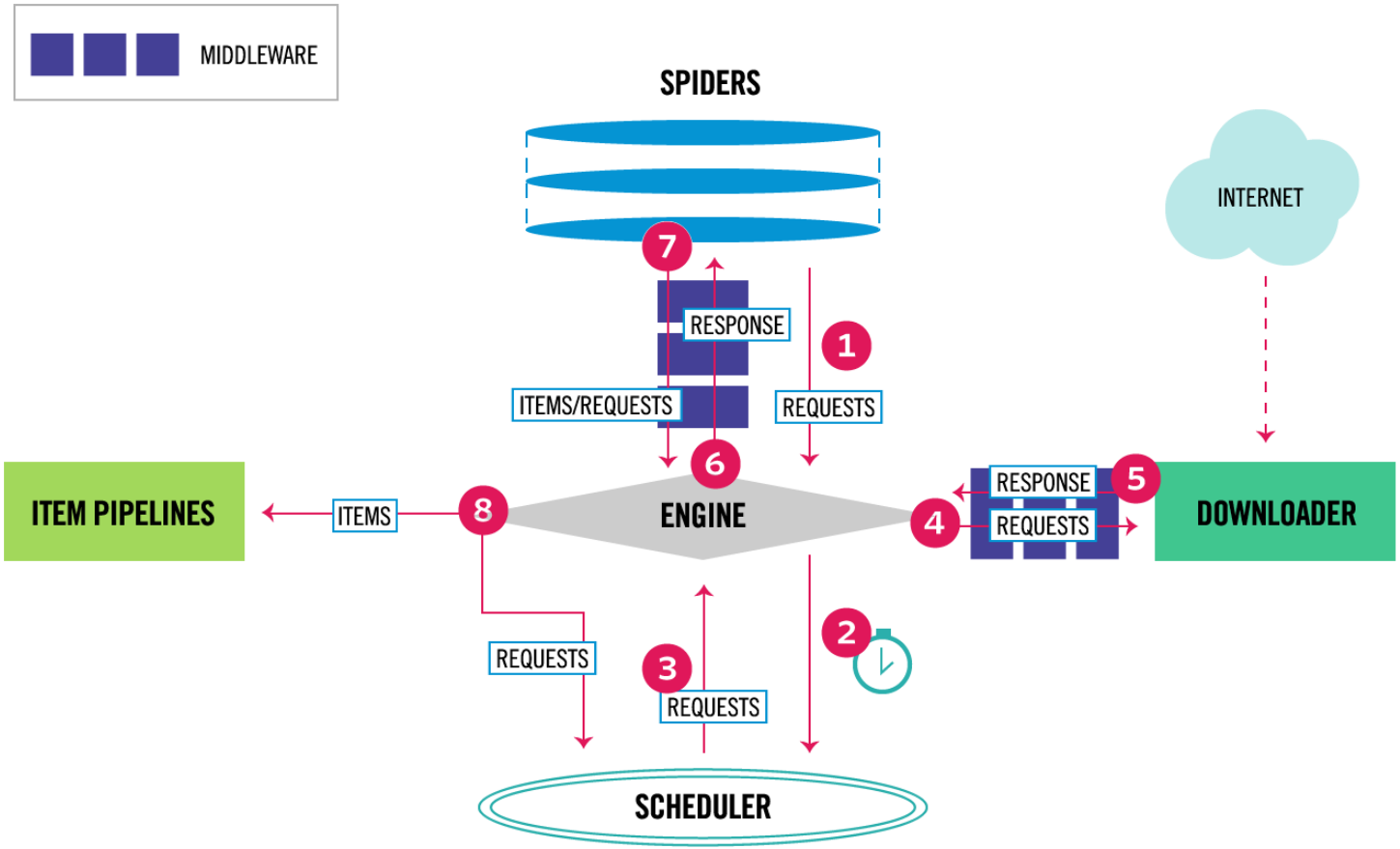
3 实战

4 参考资料

我们将学什么

- **1.** requests + beautifulsoup4
- **2.** axios + cheerio
- **3.** selenium
- **4.** scrapy
- 至于其他语言，都是一样的啦

Scrapy 框架图


MIDDLEWARE
SPIDERS
INTERNET
7
RESPONSE
1
ITEMS/REQUESTS
REQUESTS
6
ITEM PIPELINES
ITEMS
8
ENGINE
RESPONSE
5
4
REQUESTS
DOWNLOADER
REQUESTS
3
REQUESTS
2
SCHEDULER

Scrapy (/ skre pa /) is an application framework for crawling web sites and extracting structured data which can be used for a wide range of useful applications, like data mining, information processing or historical archival.
Even though Scrapy was originally designed for web scraping, it can also be used to extract data using APIs (such as Amazon Associates Web Services) or as a general purpose web crawler.
我们在之前使用的 BeautifulSoup4，lxml 等都只是构建爬虫过程中使用的工具，而 Scrapy 则是构建爬虫的框架。Scrapy 之于爬虫就如同 Django 之于后端，React 之于前端，其为构建爬虫预先准备了大量丰富的方法。

框架的组件包括

- Scrapy Engine: 调节其余各组件，控制数据流;
- Schedule: 优先队列，调度 requests;
- Downloader: 下载器，即向网络发送请求;
- Spider: 自定义类，主类，解析 responses，构建 items 等;
- Item Pipeline: 在 Spider 生成 Item 后，在该组件中执行后续工作，例如验证，加入数据库等;
- Downloader middlewares: 中间件;
- Spider middlewares: 中间件

1 爬虫简介

2 基础知识回顾

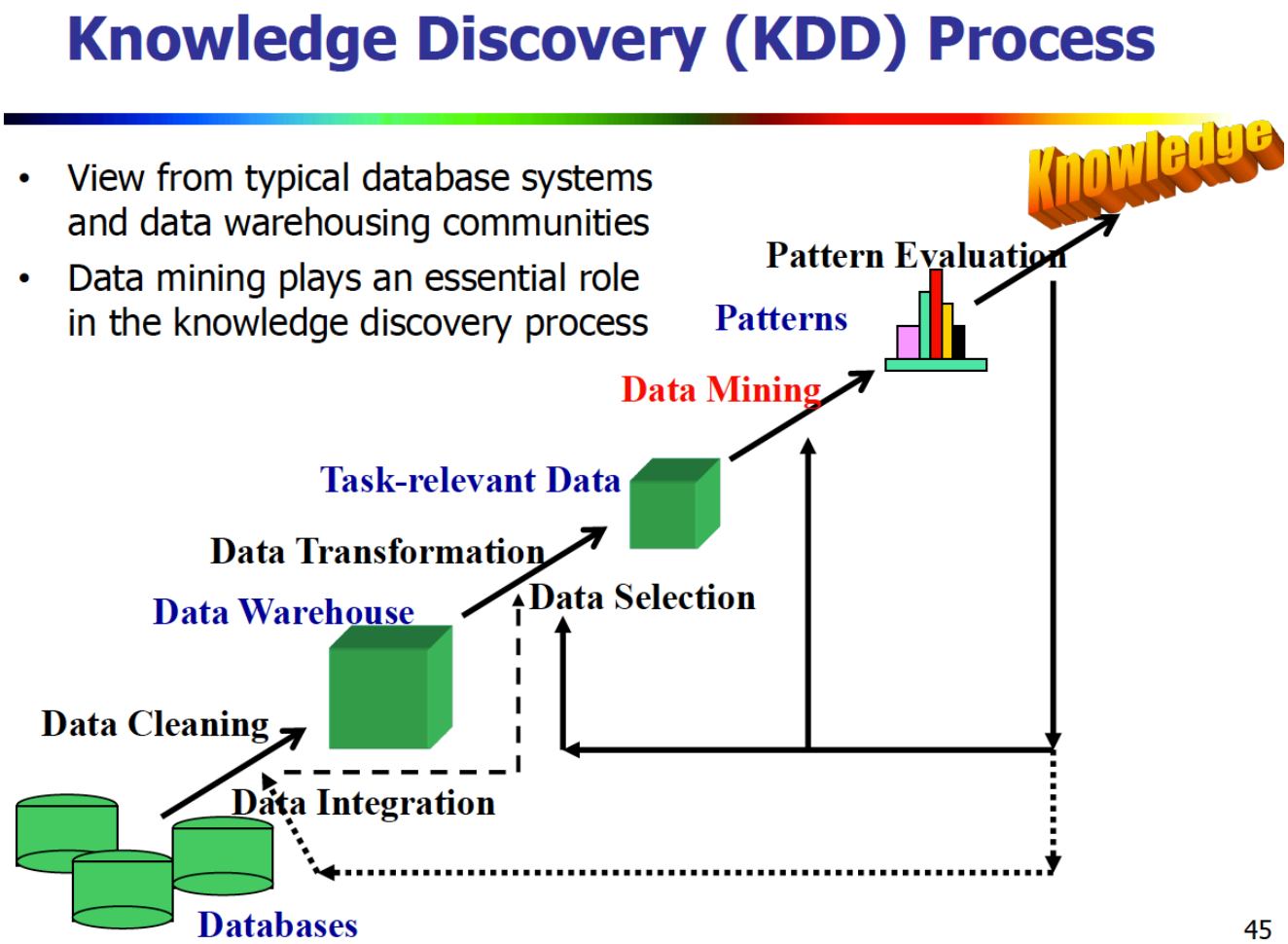
3 实战

4 参考资料

高级爬虫技巧

- 分布式爬虫
- IP 轮换和代理池
- CAPTCHA
- 数据挖掘（爬虫的本质是什么?）

人类获取知识的过程（感谢李涓子老师）



- **requests**: https://requests.readthedocs.io/en/latest/
- **beautifulsoup4**: https://beautifulsoup.readthedocs.io/zh-cn/v4.4.0/
- **selenium**: https://www.selenium.dev/
- **scarpy**: https://scrapy.org/
- **axios**: https://github.com/axios/axios
- **cheerio**: https://github.com/cheeriojs/cheerio
- **puppeteer**: https://pptr.dev/
- **EasySpider**: https://github.com/NaiboWang/EasySpider
- **sast-2023-crawler**: https://github.com/sast-summer-training-2023/sast2023-crawler (Kaiming, Liu)

作业

简单版：爬取新华网 100 条推荐视频

困难版：爬取新浪微博当前热榜签前十条

完成后可发送至 **ubecwang@gmail.com**


酒井科协暑培 5.0 课程匿名反馈
长按图片扫码

Thanks!