爬虫与 Python

随着大数据时代的到来，网络信息需求显着增长。 许多不同的企业出于各种目的从 Internet 收集第三方数据，包括竞争对手分析、新闻报道摘要、市场趋势跟踪以及通过收集每日股票价值开发预测模型。 因此，网络爬虫变得越来越重要。 根据预定义的规则设置 2808 代理 IP，爬虫自动浏览或收集来自互联网的数据。

网络爬虫分类

根据所采用的技术和组织结构，网络爬虫可以分为通用网络爬虫、专用网络爬虫、增量网络爬虫和深度网络爬虫。

基本网络爬虫工作流程

一般的网络爬虫的基本流程如下：

- 获取原始网址。 第一个 URL 链接到要抓取的网站，作为网络爬虫的入口点；
- 我们在抓取网页的过程中，必须要获取到该页面的 HTML 内容，然后解析出来，才能得到链接到这个页面的每一个页面的 URL。
- 使用这些 URL 创建队列；
- 遍历队列，逐个读取每个 URL，为每个 URL 爬取匹配的网页，重复上述爬取过程；
- 验证是否满足停止条件。 如果没有提供停止条件，爬虫将继续搜索，直到用完新的 URL。

网络爬虫的环境准备

1. 确保环境中已安装浏览器，如 Chrome、IE 等。
2. Python 的安装
3. 下载适当的 IDL。
4. 本文使用 Visual Studio Code。

为 Python 安装必要的包。

Python 包管理器称为 Pip。 它提供了用于查找、获取、安装和删除 Python 包的工具。 安装下载 Python 时，会提供这个工具。 因此，使用“pip install”安装必要的库很简单。

1. 用 pip 安装 beautifulsoup4
2. 安装 pip 请求
3. lxml 点安装

- Beautifulsoup 包使解析 HTML 和 XML 数据变得简单。
- lxml 库加速了 XML 文件的解析。

- 名为 requests 的库模拟 HTTP 请求（例如 GET 和 POST）。 我们将主要使用它来访问网站的源代码。

Python 中的网络爬虫

数据收集曾经是一个费力且有时代价高昂的过程。 没有数据，机器学习计划就不可能实现。 幸运的是，我们现在可以通过互联网获取大量信息。 要生成我们的数据集，我们可以从互联网上复制数据。使用手动下载文件并将其保存到光盘是一种选择。 但如果我们将数据收集自动化，我们就能更快地完成它。 Python 有许多工具可以帮助实现自动化同时需要借助 2808 代理 IP 产品。

如何使用请求库

为了编写一个从 Web 读取数据的 Python 应用程序，我们不可避免地要使用 requests 包。 你必须设置它。

pip 安装请求 beautifulsoup4 lxml

它为您提供了一个界面，使您可以轻松使用网络。

从 URL 读取网页是最直接的用例。

导入请求

河南的 Lat-Lon

网址 = "https://weather.com/weather/today/l/40.75,-73.98"

resp = requests.get(URL)

打印（resp.status_code）

打印（resp.text）

200

<!doctype html><html dir="ltr" lang="en-US"><head>

<meta data-react-helmet="true" charset="utf-8"/><meta data-react-helmet="true"

name="viewport" content="width=device-width, initial-scale=1, viewport-fit=cover"/>

如果数据是 JSON 格式，您可以让请求为您解码它或只是将其作为文本读取。 例如，以下代码将从 GitHub 中提取一些 JSON 格式的数据并将其转换为 Python 字典：

1. 导入请求
2. 网址 = "https://api.github.com/users/jbrownlee"
3. resp = requests.get(URL)
4. 如果 resp.status_code == 200：
5. 数据 = resp.json()
6. 打印（数据）

但是，如果 URL 包含二进制数据，如 ZIP 文件或 JPEG 图像，则必须在 content 属性中获取它，因为它是二进制数据。 例如，可以通过以下方式下载百科 LOGO：

导入请求

URL = "https://en.wikipedia.org/static/images/project-logos/enwiki.png"

wikilogo = requests.get(URL)

如果 wikilogo.status_code == 200：

以 open("enwiki.png", "wb") 作为 fp:

fp.write(wikilogo.content)

既然有了网页，我们应该如何提取数据呢？ 虽然 requests 库不能帮助我们解决这个问题，但我们可以使用另一个库来代替。 根据我们想要指定数据的方式，我们可以采用两种方法。

第一种方法需要将 HTML 视为某种类型的 XML 文档并使用 XPath 语言提取元素。 在这种情况下可以使用 lxml 库首先生成文档对象模型 (DOM)，然后执行 XPath 搜索：

1. 从 lxml 导入 etree
2. # 从 HTML 文本创建 DOM
3. dom = etree.HTML(resp.text)
4. # 搜索温度元素并获取内容
5. elements = dom.xpath("//span[@data-testid='TemperatureValue' and contains(@class,'CurrentConditions')]")

称为 XPath 的字符串描述了如何定位元素。 lxml 对象的 Xpath() 函数允许用户在 DOM 中搜索与 XPath 字符串匹配的项，该字符串可能包含多个匹配项。 使用上面的 XPath 可以将具有标签、属性数据测试匹配“TemperatureValue”和以“CurrentConditions”开头的类的 HTML 元素定位到任何地方。 通过在浏览器的开发者工具中查看 HTML 源码（如下图 Chrome 截图），我们可以查出这些信息。

此示例使用我们从该网站获取的特定元素来确定河南市的温度。 我们可以读取标签内的文本，我们知道 XPath 匹配的第一个元素就是我们需要的。这边为您提供一个免费代理 IP 以供使用。