실전 웹 크롤링과 자동화 (For 신호용)

1권: 실전 크롤링 입문 - 원티드(Wanted) 정복하기

> 집필: 시니어 개발자 멘토, Gemini
> 최종 목표: 카카오 신입 공채 합격

Chapter 1-1: 웹 크롤링의 이해와 윤리

Prologue: 첫걸음을 떼기 전에

신호용 님, `채용 정보 크롤러` 프로젝트의 첫 페이지에 오신 것을 환영합니다. 앞으로 우리는 코드를 통해 웹 세상의 정보를 수집하는 흥미로운 여정을 떠나게 될 것입니다.

하지만 강력한 기술을 배우기 전, 우리는 반드시 '책임'과 '규칙'에 대해 먼저 알아야 합니다. 훌륭한 개발자는 단순히 코드를 잘 짜는 사람을 넘어, 자신이 만든 코드가 세상에 미치는 영향을 이해하고 올바르게 사용하는 사람이기 때문입니다. 이 첫 번째 챕터는 우리가 '프로 개발자'로서 갖춰야 할 기본적인 소양에 대한 이야기입니다.

---

1. 학습 목표 (Objectives)

- 웹 크롤링의 기본 원리를 설명할 수 있다.
- `robots.txt`의 목적과 역할을 이해하고 직접 확인할 수 있다.
- 책임감 있는 크롤링을 위해 고려해야 할 기술적, 윤리적 사항을 설명할 수 있다.

---

2. 핵심 개념 (Core Concepts)

2.1 웹 크롤링(Web Crawling)이란 무엇인가?

웹 크롤링(Web Crawling)은 아주 간단하게 말해, **'사람 대신 프로그램(로봇)이 웹사이트를 자동으로 돌아다니며 정보를 수집하는 기술'**을 의미합니다. '크롤러(Crawler)' 또는 '스파이더(Spider)'라고 불리는 이 로봇은, 마치 거미가 거미줄을 타고 이동하듯, 웹페이지의 링크를 따라다니며 데이터를 긁어모읍니다.

우리가 매일 사용하는 Google 검색 엔진이 바로 세상에서 가장 거대한 웹 크롤러입니다. Google의 로봇이 전 세계의 웹사이트를 돌아다니며 정보를 수집해두었기 때문에, 우리가 무엇이든 검색하면 순식간에 결과를 보여줄 수 있는 것입니다.

> **[Our Project]**
> 우리의 프로젝트에서는 **'원티드', '점핏'**과 같은 채용 사이트를 돌아다니며 **'채용 공고'**라는 특정 정보만 수집하는 우리만의 작은 로봇을 만들게 됩니다.

2.2 크롤링의 '규칙': 개발자의 책임과 윤리

모든 웹사이트는 방문객을 환영하지만, 모든 방문객이 '착한 손님'은 아닙니다. 어떤 로봇은 너무 자주, 너무 많이 정보를 가져가려 해서 웹사이트 서버에 큰 부담을 주기도 합니다. 그래서 웹사이트 운영자들은 로봇 방문객들을 위한 몇 가지 규칙을 만들어 두었습니다.

가. `robots.txt`: "이곳은 들어오지 마세요"

`robots.txt`는 웹사이트의 루트 디렉토리(`https://sitename.com/robots.txt`)에 위치한 간단한 텍스트 파일로, 웹사이트 운영자가 크롤러에게 보내는 **'방문 규칙 안내서'**입니다.

- `User-agent:`: 어떤 로봇에게 보내는 메시지인지 지정합니다. `*`는 '모든 로봇'을 의미합니다.
- `Disallow:`: 이 경로 밑으로는 수집해 가지 말아 달라는 요청입니다. (예: `/admin`, `/mypage` 등)
- `Allow:`: `Disallow`로 막힌 경로 중, 특정 하위 경로는 수집을 허용할 때 사용합니다.

나. 과도한 요청 금지 (서버 부하 방지)

사람이 1초에 100번씩 웹페이지를 클릭할 수는 없지만, 프로그램은 가능합니다. 만약 우리의 크롤러가 사이트에 너무 짧은 간격으로 수많은 요청을 보내면, 해당 사이트 서버는 과부하로 인해 느려지거나 다운될 수 있습니다. 이는 다른 일반 사용자들에게 큰 피해를 주는 행위입니다.

> **[Developer's Tip]**
> 따라서, 우리는 코드에 **의도적인 지연 시간(`time.sleep`)**을 주어, 마치 사람이 둘러보는 것처럼 적절한 간격을 두고 정보를 요청하는 '착한 크롤러'를 만들어야 합니다.

다. API가 있다면 API를 사용하라

만약 사이트에서 공식적으로 데이터를 제공하는 **API(Application Programming Interface)**가 있다면, 크롤링 대신 API를 사용하는 것이 항상 우선입니다. API는 서버와 클라이언트가 데이터를 주고받기 위한 '공식적인 창구'이므로, 서버에 부담을 주지 않고 안정적으로 원하는 정보를 얻을 수 있는 가장 좋은 방법입니다.

---

3. 기초 실습: `robots.txt` 직접 확인하기

웹 브라우저를 열고, 우리가 잘 아는 사이트들의 `robots.txt` 파일을 직접 확인해 봅시다.

1. **네이버:** `https://www.naver.com/robots.txt`
   - `Disallow: /` 라는 내용을 보실 수 있습니다. 이는 네이버가 대부분의 경로에 대해 로봇의 접근을 허용하지 않겠다는 강력한 의지를 보여줍니다.
2. **구글:** `https://www.google.com/robots.txt`
   - 네이버와는 달리, `Disallow:` 항목이 매우 적습니다. 이는 구글이 대부분의 정보 수집을 허용하고 있음을 의미합니다.

---

4. 프로젝트 적용 (Project Integration)

우리가 앞으로 수집할 채용 사이트들은 어떨까요? 우리의 첫 번째 목표인 **원티드(Wanted)**의 규칙을 확인해 봅시다.

- **원티드:** `https://www.wanted.co.kr/robots.txt`

- 파일 내용을 살펴보면, 다행히도 채용 공고 목록 페이지인 `/wdlist`나 상세 페이지인 `/wd` 경로가 `Disallow` 되어 있지 않습니다. 이는 우리가 채용 공고 정보를 수집하는 것이 규칙에 어긋나지 않음을 의미합니다. **좋은 소식이죠!**

[Mission 1] 첫 번째 과제

신호용 님의 첫 번째 미션입니다.

> 우리가 앞으로 수집할 다른 채용 사이트인 **점핏(Jumpit)**과 **프로그래머스(Programmers) 채용**의 `robots.txt` 파일을 직접 확인해보세요.
>
> 1. 각 사이트의 `robots.txt` URL은 무엇인가요?
> 2. 파일 내용을 보고, 우리가 채용 공고를 수집하는 데 특별히 금지된 규칙이 있는지 분석하여 알려주세요.

이 과제를 통해, 우리는 본격적인 코딩에 앞서 '개발자로서의 책임감'이라는 가장 중요한 첫 단추를 끼우게 될 것입니다.

결과를 기다리겠습니다.