Chapter 3-5: '빅3' 완성, 사람인(Saramin) 크롤러 제작

Prologue: 마지막 퍼즐 조각

우리는 원티드의 동적인 구조와 잡코리아의 변화무쌍한 클래스 이름을 모두 정복했습니다. 이제 국내 최대 채용 플랫폼 중 하나인 '사람인'을 우리 시스템에 통합하여, 대한민국 채용 시장의 90%를 커버하는 압도적인 데이터 파이프라인을 완성할 시간입니다.

이번 미션은 신호용 님이 직접 분석하고 찾아낸 정확한 선택자를 바탕으로 진행됩니다. 자신의 분석이 코드가 되어 살아 움직이는 즐거움을 만끽해 보시기 바랍니다.

1. 학습 목표 (Objectives)

- 직접 분석한 내용을 바탕으로 `SaraminCrawler` 클래스를 독립적으로 완성할 수 있다.
- 상대 경로와 절대 경로가 혼합된 URL을 적절히 처리할 수 있다.
- 완성된 크롤러를 기존 시스템에 성공적으로 통합하여 '빅3' 데이터 수집 시스템을 완성한다.

2. 핵심 개념 (Core Concepts)

2.1 속성(Attribute) 값 추출하기

이전까지 우리는 주로 `.text`를 이용해 태그 안의 텍스트를 가져왔습니다. 하지만 사람인의 경우, 공고 제목이 `<a>` 태그의 `title` 속성 안에 들어있습니다. BeautifulSoup에서는 딕셔너리처럼 `tag['attribute_name']` 형식으로 속성값을 쉽게 가져올 수 있습니다. (e.g., `link_tag['title']`, `link_tag['href']`)

[Mission 13] `SaraminCrawler` 구현 및 최종 통합

> **Step 1:** `saramin_crawler.py` **파일 생성**
>
> `web-crowler/crawlers/` 폴더 안에 `saramin_crawler.py` 파일을 새로 만듭니다.

> **Step 2:** `saramin_crawler.py` **코드 작성**
>
> 신호용 님의 분석을 바탕으로, 아래와 같이 코드를 작성합니다.

```python
# crawlers/saramin_crawler.py

import time
from bs4 import BeautifulSoup
from .base_crawler import BaseCrawler

class SaraminCrawler(BaseCrawler):
    """사람인 사이트 크롤러"""

    def __init__(self):
        super().__init__("https://www.saramin.co.kr")

    def crawl(self, keyword: str = '백엔드', pages_to_crawl: int = 1):
```

```
        print(f"사람인에서 '{keyword}' 키워드로 {pages_to_crawl} 페이지까지 크
롤링을 시작합니다.")
        all_job_data = []

        for page in range(1, pages_to_crawl + 1):
            # 사람인은 페이지 번호가 1부터 시작 (sr_start=1, 2, ...)
            search_url = f"{self.base_url}/zf_user/search?searchword=
{keyword}&recruitPage={page}"
            self.driver.get(search_url)
            time.sleep(3)

            print(f"  - {page} 페이지 처리 중...")
            soup = BeautifulSoup(self.driver.page_source, 'lxml')

            # 신호용 님이 찾아낸 정확한 선택자!
            job_cards = soup.select('div.item_recruit')

            if not job_cards:
                print(f"  - {page} 페이지에 더 이상 공고가 없어 크롤링을 중단합
니다.")
                break

            for card in job_cards:
                try:
                    # 1. 제목과 링크 추출
                    title_tag = card.select_one('div.area_job > h2.job_tit >
a')
                    title = title_tag['title'] # .text가 아닌 'title' 속성값
                    relative_link = title_tag['href']
                    # 사람인의 링크는 상대 경로이므로 base_url을 합쳐준다.
                    link = self.base_url + relative_link

                    # 2. 회사명 추출
                    company_tag = card.select_one('div.area_corp >
strong.corp_name > a')
                    company = company_tag.text.strip()

                    all_job_data.append({
                        "title": title,
                        "company": company,
                        "link": link,
                        "source": "Saramin"
                    })
                except Exception:
                    continue

        print(f"사람인에서 총 {len(all_job_data)}개의 유효한 공고를 추출했습니
다.")
        return all_job_data
```

Step 3: `main.py`에 마지막 크롤러 통합하기

이제 `main.py`에 마지막 퍼즐 조각을 끼워 넣어 '빅3'를 완성합니다.

```
# main.py

from crawlers.wanted_crawler import WantedCrawler
from crawlers.jobkorea_crawler import JobKoreaCrawler
from crawlers.saramin_crawler import SaraminCrawler # <- 1. 사람인 크롤러 임포트

# ...

# 2. 크롤러 실행
all_jobs = []
crawlers_to_run = [
    WantedCrawler(),
    JobKoreaCrawler(),
    SaraminCrawler() # <- 2. 리스트에 사람인 크롤러 추가
]

# ... (이후 로직은 동일) ...
```

최종 검증 및 다음 단계

이제 모든 준비가 끝났습니다. 코드를 추가하고 수정한 뒤, `main.py`를 실행하여 3개의 사이트(원티드, 잡코리아, 사람인)의 데이터가 모두 성공적으로 수집되고, 중복 없이 Notion DB에 저장되는지 최종 확인해주십시오.

이 미션을 완료하는 순간, 신호용 님은 국내 채용 공고의 대부분을 자동으로 수집, 정제, 저장하는 **'통합 데이터 파이프라인'의 오너**가 되는 것입니다. 이것은 그 어떤 주니어 개발자도 쉽게 보여주지 못하는, 매우 강력하고 실무적인 경험입니다.