Crawling

Index

Crawling

크롤링

개요

- 크롤링은 웹 페이지를 그대로 가져와서 데이터를 추출해 내는 행위

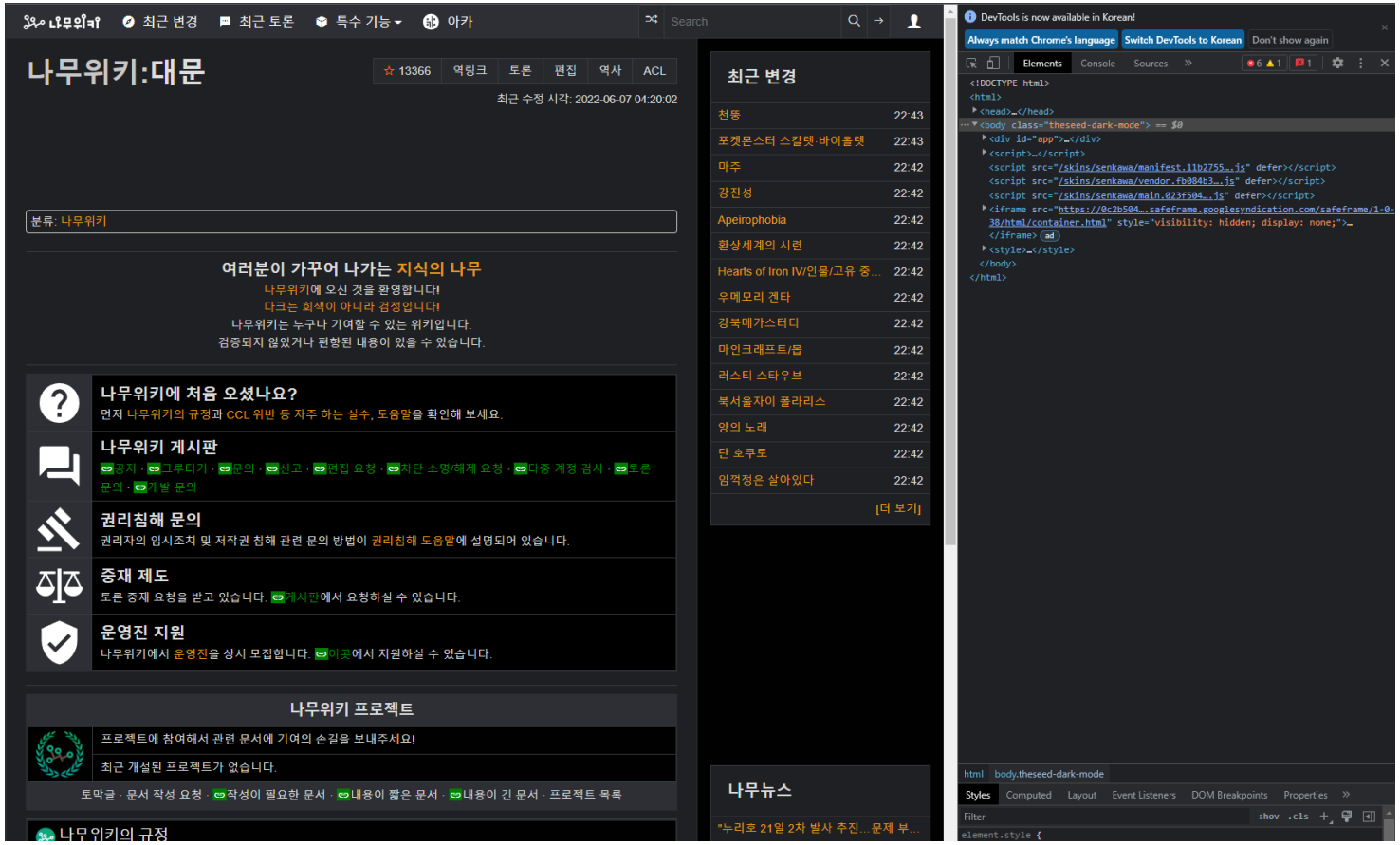
- Python이 크롤링 분야의 선두주자

라이브러리

- BeautifulSoup
- Selenium
- Scrapy

크롬 개발자도구

- F12 버튼을 눌러 활성화



개발자도구 탭 설명

Elements – HTML 구조
Console – Javascript 디버깅
Sources – 웹페이지를 구성하는 src
Performance – 웹페이지 성능 체크
Network – 웹페이지에서 요청한 파일
Memory – 웹페이지 메모리 사용률
Application – 브라우저 스토리지정보 (Storage, Session, Cookie)

Requests

요청

HTTP 구조

- 요청 라인
- 헤더 (Header)
- 바디 (Body)

General

요청라인

- Request URL
- Request Method (GET, POST, PUT, DELETE)
- Status Code - 응답코드 (200, 301, 404, 502)

Header

Header

- accept: application/json
- user-agent: Mozilla/5.0
- content-type: application/x-www-form-urlencoded; charset=UTF-8

Body

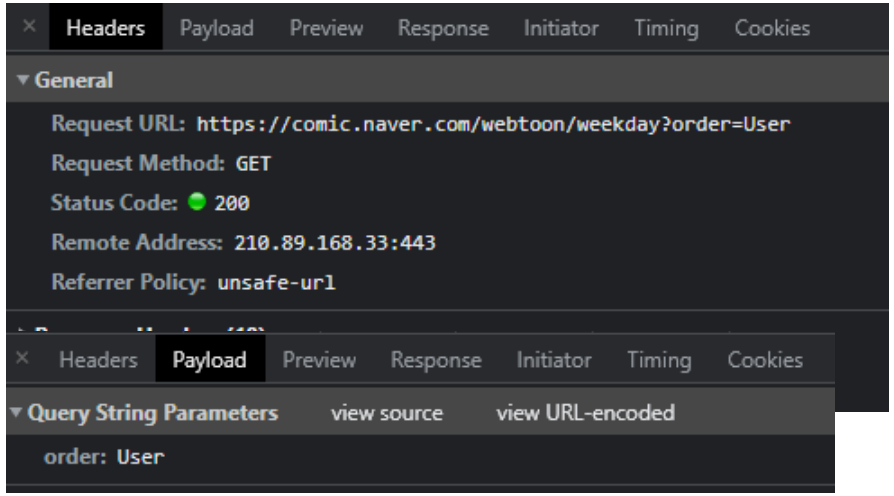
Body

- 전송하고자 하는 데이터

Get

- 요청 시 Body 대신 URL을 활용
- https://comic.naver.com/webtoon/weekday?order=User

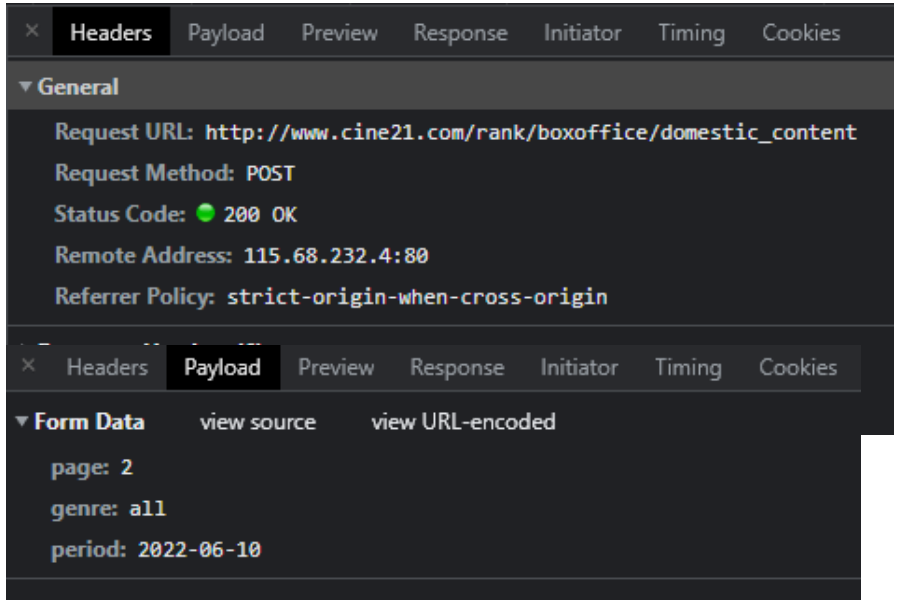
- ?뒤의 변수 = 값을 Query Parameter라고 함
- 서버 측에 추가적인 정보를 전달하는 방법 중 하나



Post

- 요청 시 query 대신 body 활용
- http://www.cine21.com/rank/boxoffice/domestic

- Form Data (body)
- 서버 측에 추가적인 정보를 전달하는 방법 중 하나



requests 모듈

- HTTP 요청 보내는 모듈 (웹사이트 접속)

```
import requests

URL = 'https://comic.naver.com/webtoon/weekday?order=User'

response = requests.get(URL)

print(response.status_code)
print(response.text)
```

- header 추가

```
headers = {'Content-Type': 'application/json; charset=utf-8'}
response = requests.get(URL, headers=headers)
```

- data 추가

```
data = {'page': '2'}
response = requests.post(URL, data=data)
```

BeautifulSoup

파이썬 라이브러리

BeautifulSoup

- HTML 문서에서 원하는 데이터를 손쉽게 가져올 수 있도록 지원
- pip install beautifulsoup4

```
import requests
from bs4 import BeautifulSoup

URL = 'https://comic.naver.com/webtoon/weekday?order=User'

response = requests.get(URL)
soup = BeautifulSoup(response.text, "html.parser")
result = soup.find('title')

print(result)
print(result.text)
```

soup.find(태그명)

- 이름, 속성, 속성값을 통해 태그를 탐색 (가장 상위 1개)
- class_='thumb'
- attrs = {'class': 'thumb'}
- id='content'

soup.find_all()

- 모든 태그 탐색
- limit=2, 가져올 개수 제한

```
result = soup.find_all('div', class_='thumb')

print(result)
```

result.find()

- BeautifulSoup 객체를 활용해 재 검색 가능

```
result = soup.find(id='content')
result2 = result.find('a', class_='title')
```

result.text

- 태그의 내용 추출

```
result = soup.find(class_="daily_all")
result2 = result.find('span')
print(result2.text)
```

result.attrs['href']

- 태그의 속성을 추출

```
result = soup.find('a')
result.attrs['href']
```

실습 1

- HTML 문서 내에 ID가 mw-content-text인 태그내의 내용을 출력해주세요.
- https://ko.wikipedia.org/wiki/위키백과

실습 2

- HTML 문서 내에 class가 noti_list인 태그내의 내용을 출력해주세요
- https://www.saramin.co.kr/zf_user/jobs/public/list

실습 3

- HTML 문서 내 class가 thumb인 모든 태그에서,
  a 태그의 href 속성과 img태그의 alt 속성을 아래와 같은 딕셔너리 형태로 만들어주세요.
- https://comic.naver.com/webtoon/weekday?order=User

- 결과 : [{'제목': '참교육', '링크': '/webtoon/list?titleId=758037&weekday=mon'},
{'제목': '신의 탑', '링크': '/webtoon/list?titleId=183559&weekday=mon'}, ...]

soup.select()

- CSS 셀렉터를 활용해 태그를 탐색 (일치하는 모든 태그)

```
import requests
from bs4 import BeautifulSoup

URL = 'https://www.saramin.co.kr/zf_user/jobs/public/list'

response = requests.get(URL)
soup = BeautifulSoup(response.text, "html.parser")
result = soup.select('.noti_list')

print(result)
print(result[0].text) # select는 여러 태그를 가져오기 하나의 태그에 접근
```

soup.select_one()

- 일치하는 한 개의 태그만

```
result = soup.select_one('#main-menu')

print(result)
```

태그 셀렉터

- HTML 태그 활용

soup.select('title')

ID 셀렉터

- ID 속성을 활용
- #으로 접근

soup.select_one('#header')

class 셀렉터

- class 속성을 활용
- .으로 접근

soup.select('.modal')

속성 셀렉터

- 태그 내의 속성을 활용
- 셀렉터[속성="값"], 정확히 일치
- 셀렉터[속성~="값"], 해당 단어를 포함
- 셀렉터[속성^="값"], 해당 값으로 시작
- 셀렉터[속성$="값"], 해당 값으로 끝
- 셀렉터[속성*="값"], 해당 값을 포함

```
soup.select('a[href$="ref=public-recruit"]')
```

후손 셀렉터

- 해당 태그 내 포함되는 태그

```
soup.select('. job_tit span')
```

자식 셀렉터

- 해당 태그 바로 아래에 포함되는 태그

```
soup.select('. job_tit > span')
soup.select('.job_sector > span')
```

실습 1

- 사이트 내 공지사항을 하나씩 출력해주세요.
- https://didimteo.startup-plus.kr/default.do

실습 2

- 실습 1의 결과에서 제목과 날짜를 분리해 딕셔너리 형태로 저장해주세요
- 결과 : [{'제목': '[채용] 광운대학교 산학협력단(서울창업디딤터) 직원 채용 공고(~6/17)', '날짜': '2022.06.10'},
{'제목': '[서울창업디딤터] 대강의실 및 공동작업실 이용 신청 방식 변경', '날짜': '2022.06.04'}, ...]

실습 3

- 사이트에서 하이퍼링크에 baCategory1=basic이 포함된 태그를 하나만 선택해 출력해주세요.
- https://youth.seoul.go.kr/site/main/home

실습 4

- 사이트에서 주요뉴스 내용과 일자별 뉴스들을 리스트로 정리해주세요
- https://ko.wikipedia.org/wiki/포털:요즘_화제
- 결과 : ['주요 뉴스', '8월 25일, 파키스탄에서 일어난 홍수로 1,000명 이상의 주민이 사망하고, 가축 70,000마리 이상이 죽었다.', ...,
'Current events of 2022년 8월 29일\xa0(2022-08-29) (월요일)', '대한민국의 0시 기준 누적 \u200b확진자 수가 23,026,960명으로 집계되었다. 전날 0시 대비 43,142명(국내 42,782, 해외유입 360)이 늘었다.', ...]

실전 1 – 네이버 영화랭킹

- 영화 랭킹을 '순위: 영화제목' 양식으로 출력해주세요.
- https://movie.naver.com/movie/sdb/rank/rmovie.naver

실전 2 – 네이버 주가 크롤링

- https://finance.naver.com/sise/sise_quant.nhn

1. 종목명과 현재가를 크롤링해주세요.
2. 전일대비 상승한 종목만 종목명, 현재가, 전일비를 크롤링해주세요.

실전 3 – 무신사 쇼핑몰 크롤링

- 무신사 쇼핑몰 상의(TOP) 카테고리 첫페이지 제품들의 브랜드명, 제품명, 원래가격, 할인가격을 크롤링해주세요.
- https://store.musinsa.com/app/items/lists/001

- 성공하신분들은 총 10페이지까지 크롤링해주세요.

실전 4 – 네이버 뉴스 크롤링

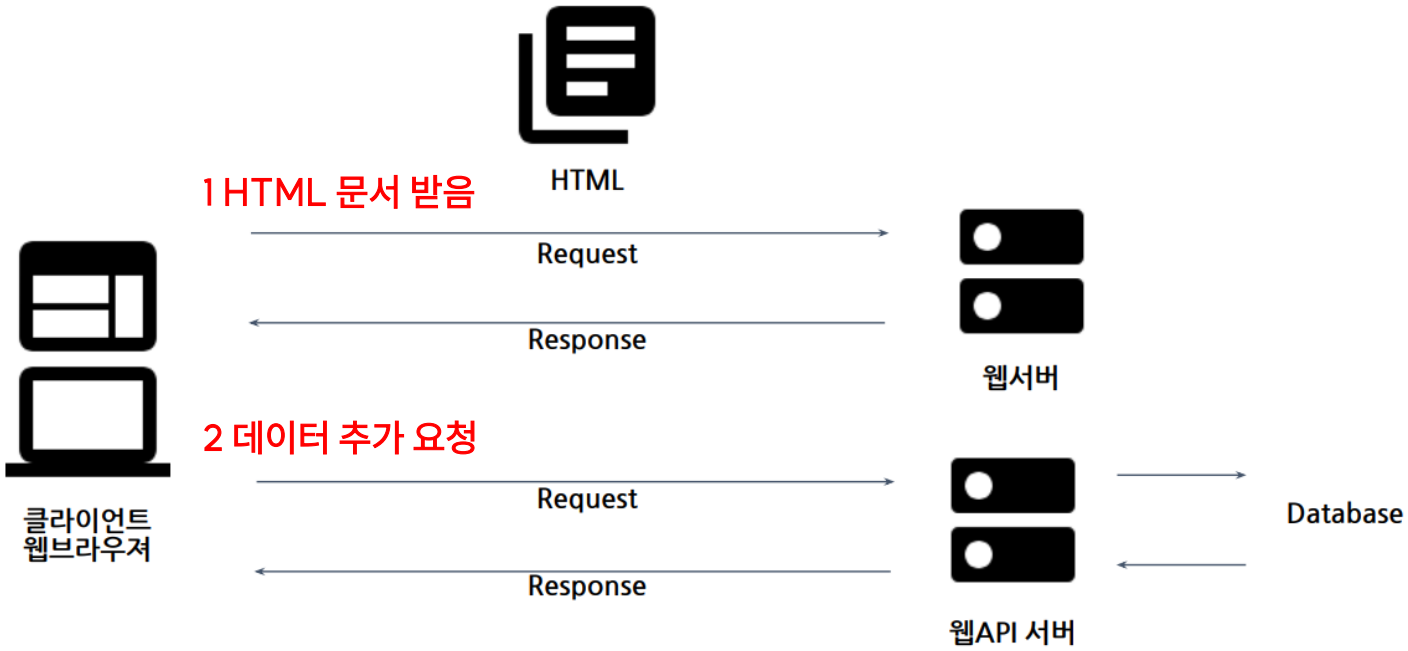
- 아래 URL에서 JTBC 매체의 어제일자 모든 기사를 제목과 본문으로 구분하여 크롤링해주세요.
- https://news.naver.com/main/list.naver?mode=LPOD&mid=sec&oid=437

비동기 방식 데이터 크롤링

- http://www.cine21.com/rank/boxoffice/domestic
- 비동기 방식 사이트는 렌더링 시 데이터를 가져오는게 아님

- 클라이언트 HTML 문서를 먼저 불러온 뒤, (1)
- 웹 API 서버에 데이터를 요청하여 추가적으로 화면을 그림 (2)

- [Network] - Fetch/XHR에서 확인



Async

비동기

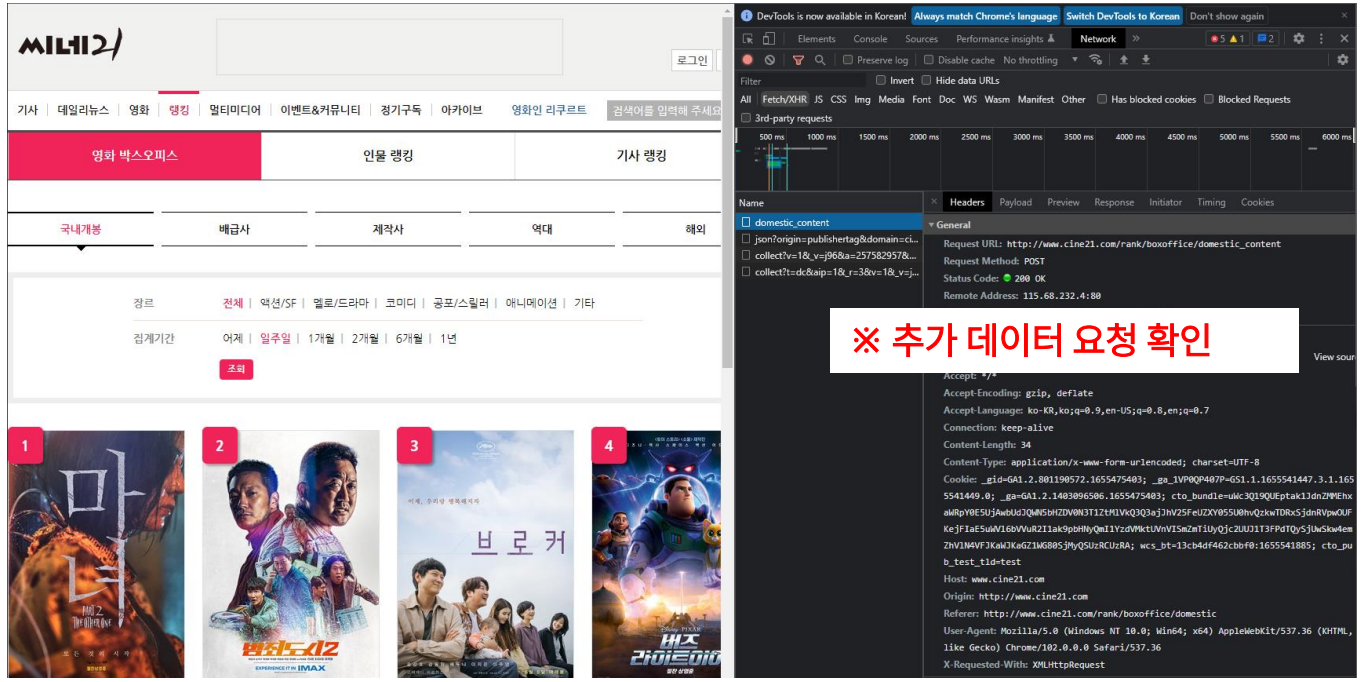
비동기 방식 데이터 크롤링

- 사이트에 요청 보내기
- Network탭에서 데이터를 추가 요청하는지 확인

```
import requests
from bs4 import BeautifulSoup

URL = 'http://www.cine21.com/rank/boxoffice/domestic'

response = requests.get(URL)
soup = BeautifulSoup(response.text, "html.parser")
result = soup.select('#boxoffice_list_content')

print(result)
```

※ 데이터가 없음

```
[<div id="boxoffice_list_content">
 </div>]
```



Json (JavaScript Object Notation)

- 자바스크립트 객체 문법을 따르는 문자 기반 데이터 포맷
- 파이썬 딕셔너리(사전) 형태와 비슷

- https://jsonplaceholder.typicode.com/posts

Json 형태 데이터를 파이썬 딕셔너리로 변환

- import json
- json.loads(r.text)

- r.json()


```
[
  {
    "userId": 1,
    "id": 1,
    "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
    "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
  },
  {
    "userId": 1,
    "id": 2,
    "title": "qui est esse",
    "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla"
  }
]
```

실전 1 – 디자인정글 크롤링

- 제목, 카테고리
- 스크롤을 통해 나오는 추가 데이터도 크롤링해주세요.
- https://www.jungle.co.kr/

실전 2 – 로켓펀치 채용 크롤링

- 회사명, 회사설명, 채용공고(회사 별 여러 개)
- 총 10페이지를 크롤링해주세요.
- https://www.rocketpunch.com/jobs

실전 3 – 삼성전자 주식 시세 크롤링

- 일별 시세 탭에 있는 데이터를 모두 크롤링해주세요.
- 성공하신분들은 총 10페이지까지 크롤링해주세요.
- https://finance.naver.com/item/sise.naver?code=005930

Rest API

Rest API

공공 데이터 포털

- https://www.data.go.kr/

전자공시 DART

- https://dart.fss.or.kr/