

/\*elice\*/

# 파이썬 크롤링

API를 이용한 크롤링



김경민 선생님

**API**

# API란?

API(Application Programming Interface)는  
어떤 프로그램과 또 다른 프로그램을 연결해주는 매개체입니다.

컴퓨터를 다루기 위해 마우스와 키보드를 이용하는 것처럼

API는 프로그램 사이를 연결해주는 역할을 합니다.

# API란?

예를 들어 지도 데이터를 이용하여

맛집 찾기 웹 서비스를 제작하려면 어떻게 해야 할까요?

# API란?

보통의 일반인들에게는 지도 데이터를 **갖고 있지 않고**,

이를 **수집**하는 것도 매우 어렵습니다.

그렇다고 **공개된 데이터**를 그대로 사용하는 것도 어렵습니다.

# API란?

Google이 갖고 있는 지도 데이터를 공개하였다고 가정해봅시다.

# API란?

그러나 **원본** 데이터는 너무 방대하기도 하고,  
호환성 등의 문제도 있어 **쉽게 사용할 수 없습니다.**

마치 키보드와 마우스가 없는 컴퓨터를 사용하는 것과 같습니다.

# API란?

그래서 Google은 지도 데이터를 응용하여 사용할 수 있도록

Google Map API라는 **매개체**를 사용자들에게 **제공**합니다.



# API란?

## 인기 검색

1	삼성전자	49,400	▼	-0.90%	6	셀트리온	212,500	▼	-0.70%
2	파미셀	19,300	▼	-12.67%	7	현대차	90,500	▼	-2.06%
3	한진칼	92,300	▲	+8.33%	8	KODEX WTI원유선물...	4,200	▲	+1.69%
4	에스맥	1,500	▲	+7.14%	9	SK하이닉스	81,500	▼	-1.33%
5	셀트리온헬스케어	85,200	▲	+1.43%	10	씨젠	90,200	▲	+0.11%

## 업종 상위 - 코스피

1	음식료품	+1.58%	▲	풀무원
2	운수창고	+1.18%	▲	한진칼
3	의료정밀	+1.19%	▲	한화시스템
4	종이목재	-0.04%	▼	모나리자
5	철강금속	-0.48%	▼	풍산

## 업종 상위 - 코스닥

1	운송	+3.59%	▲	W홀딩컴퍼니
2	섬유·의류	+1.13%	▲	케이엠
3	유통	+0.47%	▲	에이프로젠 H...
4	음식료·담배	+0.19%	▲	아미코젠
5	인터넷	-0.26%	▼	다나와

위 사진은 daum 증권 사이트입니다.

여러 기업들의 주가 정보를 API를 거쳐 받아온 후 표시하고 있습니다.

# API란?

방금 보신 daum 증권 사이트와 같이

API를 이용해 정보를 가져오는 웹 사이트가 꽤 있습니다.

이런 경우 정보가 HTML에 **처음부터 존재하지 않고,**

정보를 **API로부터 불러오고 나서** HTML에 존재하게 됩니다.

# API란?

따라서 daum 증권 사이트에서는

BeautifulSoup를 이용하여 주가 데이터를 크롤링할 수 없습니다.

웹 사이트를 처음 로드할 때 HTML 문서에는

주가 데이터가 존재하지 않기 때문입니다.

# API란?

보통 API를 이용하여 데이터를 불러오는 경우는

데이터가 ‘동적’으로 변화하는 일이 많아

**실시간**으로 값을 불러와야 하는 경우입니다.

기업의 주가도 하나의 예시입니다.

# API란?

이럴 땐 daum 증권 사이트에서  
주가 정보를 요청하는 **API에 접근**하여  
어떤 정보를 전달해주고 있는지 접근하면 됩니다.

# API란?

The screenshot shows a web browser with the URL `finance.daum.net`. The page displays financial news and market data. The developer tools are open, showing the Network tab. A request to `ranks?limit=10` is selected, and its response is visible. The response is a JSON array of 10 objects, each representing a stock's rank and change.

```
{
  "data": [
    {
      "rank": 1,
      "rankChange": 1,
      "symbolCode": "A005690",
      "code": "KR7005690003",
      "name": "표미셀",
      "tradePrice": 19000,
      "..."
    },
    {
      "rank": 2,
      "rankChange": -1,
      "symbolCode": "A005930",
      "code": "KR7005930003",
      "name": "삼성전자",
      "tradePrice": 49350,
      "..."
    },
    {
      "rank": 3,
      "rankChange": 4,
      "symbolCode": "A097780",
      "code": "KR7097780001",
      "name": "에스맥",
      "tradePrice": 1585,
      "..."
    },
    {
      "rank": 4,
      "rankChange": 2,
      "symbolCode": "A096530",
      "code": "KR7096530001",
      "name": "씨젠",
      "tradePrice": 89500,
      "..."
    },
    {
      "rank": 5,
      "rankChange": -1,
      "symbolCode": "A000660",
      "code": "KR7000660001",
      "name": "SK하이닉스",
      "tradePrice": 81400,
      "..."
    },
    {
      "rank": 6,
      "rankChange": 0,
      "symbolCode": "A064550",
      "code": "KR7064550007",
      "name": "바이오니아",
      "tradePrice": 11650,
      "..."
    },
    {
      "rank": 7,
      "rankChange": 1,
      "symbolCode": "A261220",
      "code": "KR7261220008",
      "name": "KODEX WTI원유선물(H)",
      "tradePrice": 4150,
      "..."
    },
    {
      "rank": 8,
      "rankChange": 2,
      "symbolCode": "A068270",
      "code": "KR7068270008",
      "name": "셀트리온",
      "tradePrice": 211500,
      "..."
    },
    {
      "rank": 9,
      "rankChange": 0,
      "symbolCode": "A005380",
      "code": "KR7005380001",
      "name": "현대차",
      "tradePrice": 90100,
      "..."
    },
    {
      "rank": 10,
      "rankChange": -7,
      "symbolCode": "A245620",
      "code": "KR7245620000",
      "name": "EDGC",
      "tradePrice": 15750,
      "..."
    }
  ]
}
```

크롬 개발자 도구의 **Network** 탭에서  
웹사이트가 데이터를 요청하는 API를 볼 수 있습니다.

# API란?

```
url = "http://finance.daum.net/api/search/ranks?limit=10"  
req = requests.get(url) # JSON 데이터
```

API의 URL에 GET 요청을 보내면 **JSON 데이터**를 얻을 수 있습니다.

JSON은 **key**와 **value**를 저장하는, 딕셔너리 꼴의 데이터 형식입니다.

# API란?

몇몇 웹 사이트들은 크롤러 등을 통한 기계적인 접근을 막고 있습니다.

이를 우회하기 위해 requests.get 메소드에

**"headers"** 매개변수를 지정해주셔야 합니다.



# API란?

‘헤더’란 HTTP 상에서 클라이언트와 서버가  
요청 또는 응답을 보낼 때 전송하는 **부가적인 정보**를 의미합니다.

실습에서 headers에 사용할 옵션을 제공하고 있습니다.

# API란?

```
custom_header = {  
    'referer' : ...  
    'user-agent' : ... }
```

**referer**와 **user-agent** 옵션을 지정하고 있는데,

referer는 **이전 웹 페이지의 주소**를 의미하고

user-agent는 이용자의 여러 가지 **사양**을 의미합니다.

/\* elice \*/

[실습1]

# 필요한 정보를 담고 있는 API에 접근



/\* elice \*/

[실습2]

# 네이버 실시간 검색어 크롤링



# 프로젝트 - 음식점 리뷰 크롤링하기

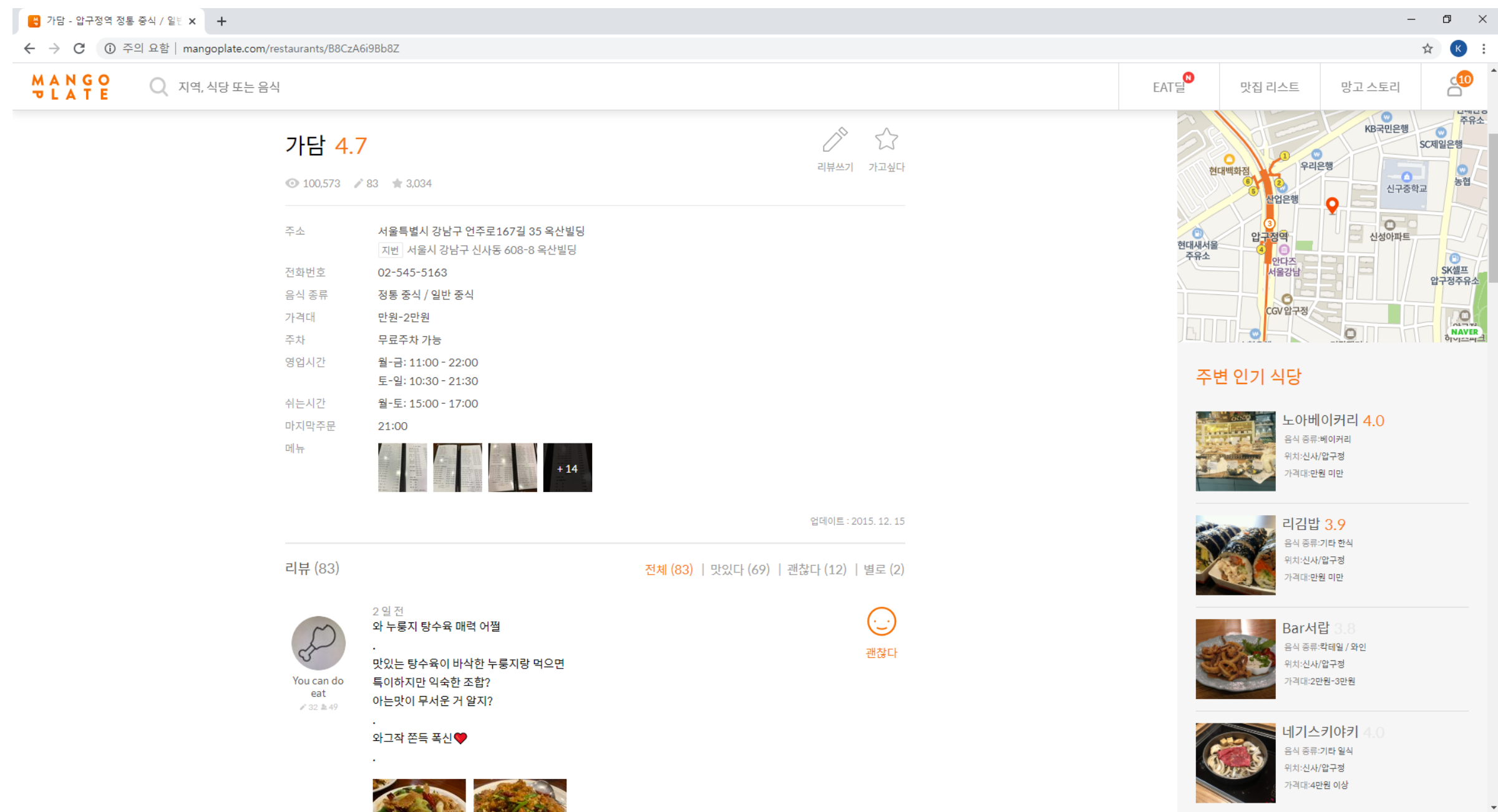
# 음식점 리뷰 크롤링

음식점을 소개하고 추천하는 서비스를 제공하는 웹 사이트인

‘망고플레이트’의 데이터로 음식점 리뷰를

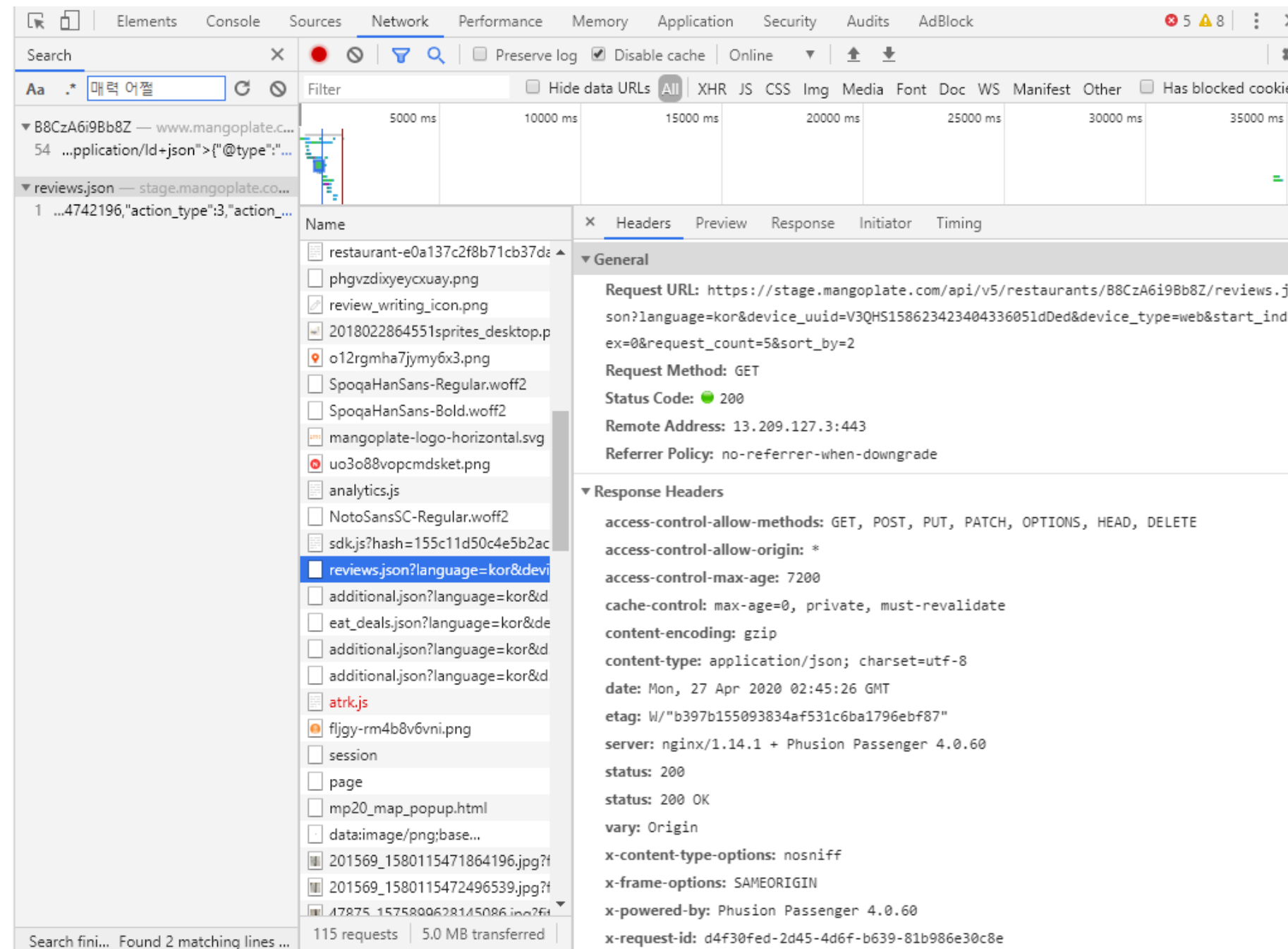
크롤링하는 실습을 만들어보겠습니다.

# 음식점 리뷰 크롤링



음식점 페이지에 들어가면, 하단에 리뷰가 있습니다.

# 음식점 리뷰 크롤링



개발자 도구를 통해 리뷰를 불러오는 곳을 조사할 수 있습니다.



# 음식점 리뷰 크롤링

리뷰를 불러오는 API의 URL에 접근하여  
어떤 음식점의 URL이 주어졌을 때 **해당 음식점의 리뷰**를  
모두 불러오는 코드를 작성해보세요.

# 음식점 리뷰 크롤링

그리고 특정 키워드를 검색하였을 때

나타나는 음식점들의 href으로 접근하여

여러 음식점들의 리뷰를 불러오는 코드도 작성해보세요.

/\* elice \*/

[실습3]

# 음식점 리뷰 크롤링



`/* elice */`

[실습4]

# 음식점 href 링크 크롤링



/\* elice \*/

[실습5]

# 검색 결과 음식점 리뷰 크롤링





`/*elice*/`

[contact@elice.io](mailto:contact@elice.io)