

Week 11 · 데이터 저널리즘

## Social Data Mining 02

---

Joonhwan Lee  
human-computer interaction + design lab.

## 오늘 다룰 내용

---

- Crawling Twitter Data
- OAuth 인증
- Crawling Data using OpenAPI
- Advanced Web Crawling

# 1. Crawling Twitter Data

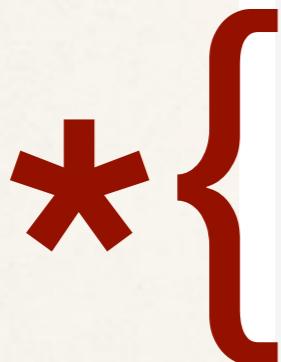
---

# Twitter API 소개

- ◆ 트위터에서 데이터를 수집하기 위해 제공되는 API
  - ◆ REST API  
<https://dev.twitter.com/rest/public>
  - ◆ Streaming API  
<https://dev.twitter.com/streaming/public>  
등이 제공된다.
- ◆ 사용자가 너무 많은 데이터 수집하는 것을 막기 위해 여러 제한 장치를 두고 있음.
- ◆ 개발자는 먼저 트위터 개발자로 등록하여 제작할 어플리케이션에 인증도구로 사용할 consumer\_key 등을 받아야 함.
  - ◆ <https://dev.twitter.com/docs>

# Twitter 개발자 등록

- ❖ <https://apps.twitter.com/app/new>



## Create an application

### Application Details

Name \*

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description \*

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website \*

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.

(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth\_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

### Developer Agreement

Yes, I have read and agree to the [Twitter Developer Agreement](#).

Create your Twitter application

# Twitter 개발자 등록

- ♦ Customer Key and Access Token

**hcid-dev-test-app-for-python**

Test OAuth

Details    Settings    **Keys and Access Tokens**    Permissions

### Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

{ Consumer Key (API Key) [REDACTED]  
Consumer Secret (API Secret) [REDACTED]  
Access Level Read and write ([modify app permissions](#))  
Owner [REDACTED]  
Owner ID [REDACTED]

### Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

{ Access Token [REDACTED]  
Access Token Secret [REDACTED]  
Access Level Read and write  
Owner [REDACTED]  
Owner ID [REDACTED]

# Data Formats for Exchange

- ❖ Twitter, Facebook 등의 API는 JSON 포맷으로 데이터를 전달.
- ❖ JSON (JavaScript Object Notation)
  - ❖ 인터넷에서 자료를 주고받을 때 그 자료를 표현하는 방법. 자료의 종류에 큰 제한은 없으며, 특히 컴퓨터 프로그램의 변수값을 표현하는 데 적합. (<http://ko.wikipedia.org/wiki/JSON>)
  - ❖ key:value 형태로 되어 있으며, 자바스크립트의 구문 형식을 따르고 있으나 프로그래밍 언어나 플랫폼에 독립적이어서 많은 프로그래밍 언어가 사용.

# Data Formats for Exchange

## ◆ JSON(JavaScript Object Notation)

- ◆ {"name2": 50, "name3": "값3", "name1": true} → JSON 객체
- ◆ {  
    "이름": "테스트",  
    "나이": 25,  
    "성별": "여",  
    "주소": "서울특별시 양천구 목동",  
    "특기": ["농구", "도술"],  
    "가족관계": {"#": 2, "아버지": "홍판서", "어머니": "춘섬"},  
    "회사": "경기 안양시 만안구 안양7동"  
}

# Data Formats for Exchange

## ♦ JSON: Facebook Example

```
{
  "id": "100000000000000",
  "name": "Joonhwan Lee",
  "education": [
    {
      "school": {
        "id": "111485558870421",
        "name": "영동고등학교"
      },
      "year": {
        "id": "112936752090738",
        "name": "1989"
      },
      "type": "High School"
    },
    {
      "school": {
        "id": "104038622966911",
        "name": "Seoul National University"
      },
      "year": {
        "id": "137409666290034",
        "name": "1995"
      },
      "type": "College"
    },
    {
      "school": {
        "id": "7133766387",
        "name": "Carnegie Mellon University"
      },
      "degree": {
        "id": "170434169669210",
        "name": "PhD"
      },
      "year": {
        "id": "170434169669211",
        "name": "1998"
      }
    }
  ]
}
```

# Data Formats for Exchange

- ❖ XML: Extensible Markup Language
  - ❖ W3C에서 정의된 마크업 언어.
  - ❖ HTML도 XML의 일종.
  - ❖ 여러 종류의 데이터를 기술하는데 사용 됨.
  - ❖ 파일의 크기가 커진다는 단점이 있음  
→ 순수하게 데이터만 교환하고자 할 때는 JSON을 선호.

# Data Formats for Exchange

## ❖ XML: Food Menu

```
▼<breakfast_menu>
  ▼<food>
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    ▼<description>
      Two of our famous Belgian Waffles with plenty of real maple syrup
    </description>
    <calories>650</calories>
  </food>
  ▼<food>
    <name>Strawberry Belgian Waffles</name>
    <price>$7.95</price>
    ▼<description>
      Light Belgian waffles covered with strawberries and whipped cream
    </description>
    <calories>900</calories>
  </food>
  ▼<food>
    <name>Berry-Berry Belgian Waffles</name>
    <price>$8.95</price>
    ▼<description>
      Light Belgian waffles covered with an assortment of fresh berries and whipped cream
    </description>
    <calories>900</calories>
  </food>
  ▼<food>
    <name>French Toast</name>
    <price>$4.50</price>
    ▼<description>
      Thick slices made from our homemade sourdough bread
    </description>
    <calories>600</calories>
  </food>
  ▼<food>
    <name>Homestyle Breakfast</name>
    <price>$6.95</price>
    ▼<description>
      Two eggs, bacon or sausage, toast, and our ever-popular hash browns
    </description>
    <calories>950</calories>
  </food>
</breakfast_menu>
```

# Using JSON from Python

- ♦ json 모듈을 이용하여 JSON 불러오기

```
import json  
json_data = json.loads(json_string)
```

\* json\_data는 python dictionary

# Twitter 사용자 정보의 수집

- ❖ tweepy 및 OAuth 설정
  - ❖ 앞서 부여받은 개발자 token 을 사용하여 트위터로의 접근을 승인받을 수 있도록 설정 작업을 한 후 tweepy api 오브젝트를 생성한다.

```
import tweepy

# OAuth setup
consumer_key = 'YOUR-CONSUMER-KEY'
consumer_secret = 'YOUR-CONSUMER-SECRET'
access_token = 'YOUR-ACCESS-TOKEN'
access_secret = 'YOUR-ACCESS-SECRET'

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)

api = tweepy.API(auth)
```

# Twitter 정보 수집

```
• api = tweepy.API(auth)
  my_timeline = api.home_timeline()
  for tweet in my_timeline:
    print(tweet.text)
```

>>

RT @skibbie81: 사실 오늘 나도 히스패닉 대학원생이 아파트 현관에 go home이라고 누가 낙서해 놔서 패닉에 떨며 우는 것도 봤다 숨어 있던 제노포빅이 아파트 현관에 그런 낙서를 할 만큼 대담해졌다는 게 끔찍하다. 누가 그런 용기를 줬는...

RT @tora\_ru: 트럼프 효과가 벌써부터 나타난다..

오늘 내 동생이 학교가서 트럼프 지지하던 백인 학생들한테 "북한으로 돌아가라" 라는 말 들었음... 그런 독설을 동생한테뿐만이 아니라 울던 여자아이들한테도 퍼부었다고..

RT @Keyton\_S\_Park: 요즘 칼 세이건 아저씨를 인용해서 정신을 다잡는 -\_-;; 트윗이 많은데 CDMA 기술을 개발한 과학자이자 배우 "헤디 라마르" 여사를 많이 기억해주셨으면 좋겠다. 이 분 생일이 아마 미대선일인가, 그 다음날이었을거...

RT @PRESSIAN\_news: 경찰이 11월 12일 서울광장~청운효자동주민센터 행진을 불허했습니다. 집시법 위반도 아닌데 말이죠. 그래도 우리는 행진합니다. 일명 #청와대\_에워싸기! 포스터 참고하세요. <https://t.co/01kacdAuMX>

# Twitter Streaming APIs

- ♦ <https://dev.twitter.com/docs/streaming-apis>
- ♦ 트위터 메시지를 실시간으로 전송하는 API
  - ♦ REST API: request 한 메시지만 가져올 수 있다.
  - ♦ Public Streaming API
  - ♦ User Streaming API
  - ♦ Site Streaming API

# Twitter Streaming APIs

- ❖ Public Streaming API
  - ❖ 전체 데이터 중 1%를 랜덤으로 실시간 전송
  - ❖ 하루 400만건 정도 수집 가능
  - ❖ Global Trends 분석 등에 사용
- ❖ User Streaming API
  - ❖ 인증된 사용자에게 한 사용자의 모든 정보를 실시간 전송
- ❖ Site Streaming API
  - ❖ 여러 사용자의 user stream 데이터를 실시간 전송

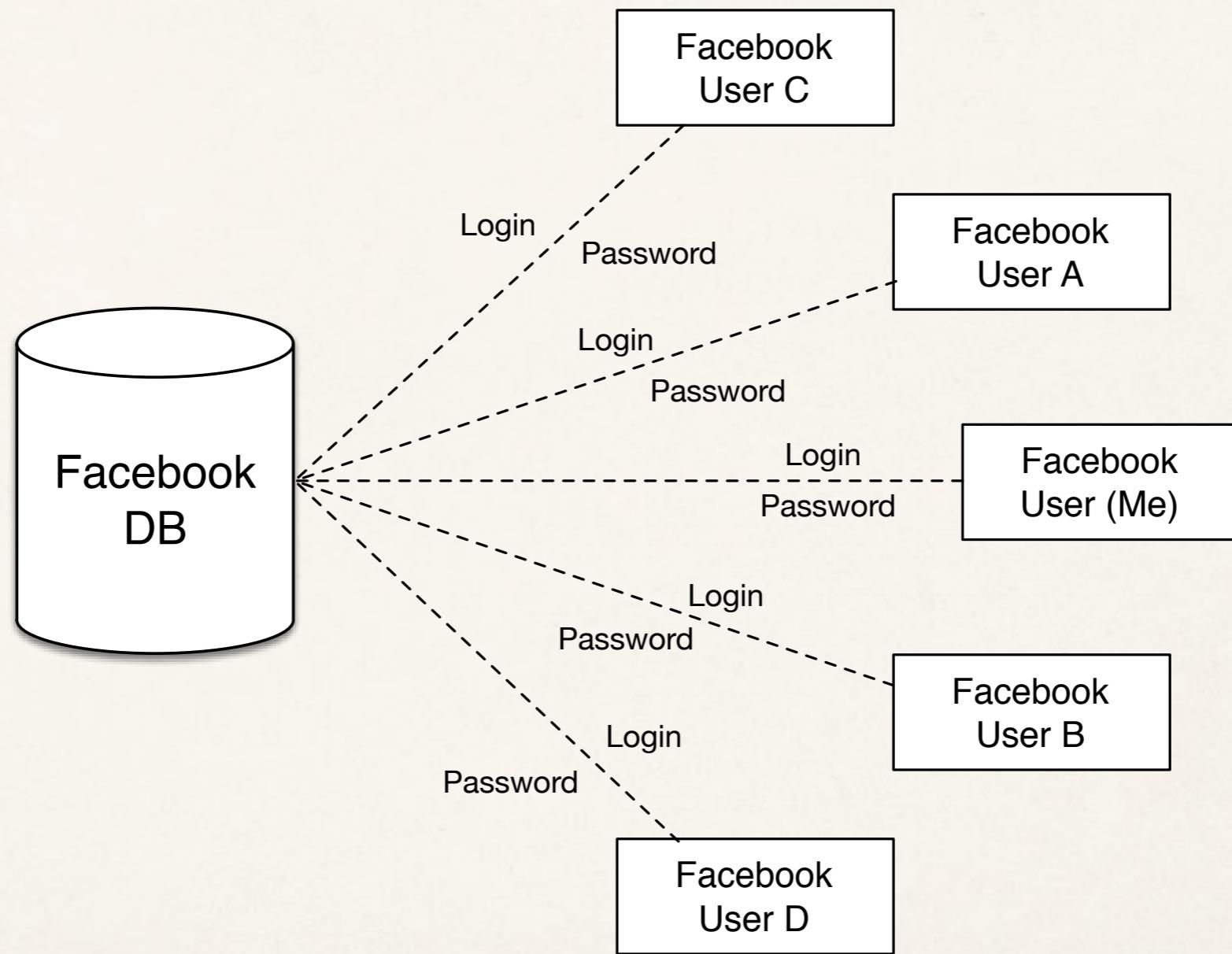
# Streaming Tweet Data

- ❖ Streaming API를 사용하여 트윗을 수집하려면 다음과 같은 순서를 따른다.
  - ❖ StreamListener 클래스를 상속받은 lister 클래스를 만든다.
  - ❖ Stream 오브젝트를 생성한다.
  - ❖ Stream 오브젝트에 Twitter API를 연결한다.

## 2. OAuth 인증

---

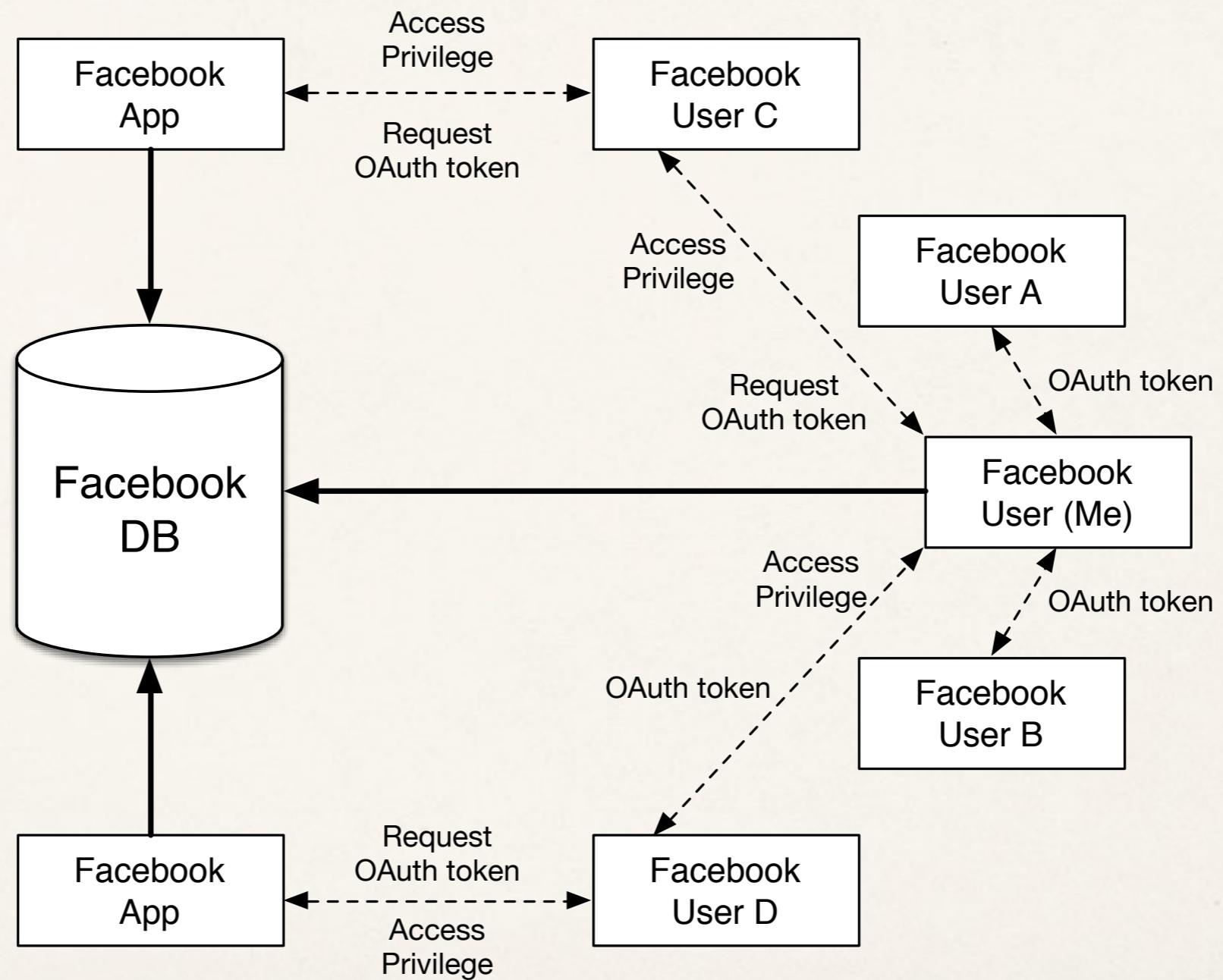
# Facebook Login



## OAuth 인증

- ♦ OAuth는 3rd party를 위한 범용적인 인증 표준. 제3자가 사용자의 ID, 패스워드 대신 Access Token이라는 것을 얻어서 인증이 필요한 데이터에 접근.
- ♦ OAuth는 구글, 야후, 트위터, 페이스북, 다음, 네이버, 네이트 등 주요 인터넷 기업이 사용하는 인증 기술.

# OAuth 인증



## 3. Crawling using OpenAPI

---

# OpenAPI의 사용

- ♦ Twitter → 데이터 크롤링을 위한 파이썬 라이브러리를 이용하여 데이터를 수집할 수 있었다.
  - ♦ 이들 라이브러리는 Twitter가 제공하는 OpenAPI에 맞게 개발되었다.
- ♦ 모든 서비스를 위한 라이브러리가 제공되는지는 않기 때문에 특정 서비스의 OpenAPI를 이용하려면 OpenAPI가 제공하는 방식에 맞게 프로그램을 설계하고 데이터를 수집한다.

# OpenAPI의 사용

- ❖ API를 사용하기 위해서는 공개된 OpenAPI라 하더라도 개발자로 등록을 해야 한다.
- ❖ 개발자로 등록을 한 후에는 데이터 수집을 위해 제작한 application을 등록한다.
- ❖ application을 등록하면 보통 app-key라는 것을 주는데, 이것은 일종의 아이디-패스워드이다. 즉, 누가 접속을 해서 데이터를 수집해가는지를 서버에 알려주는 역할을 하며, 또한 서버 입장에서 데이터를 수집하는 앱의 트래픽을 콘트롤하기도 한다. (대개의 경우 call 숫자가 정해져 있다.)

# OpenAPI 제공 사이트

## ♦ data.go.kr (정부 3.0)

The screenshot shows the homepage of the data.go.kr website. At the top, there is a navigation bar with the logo '정부 3.0 DATA .GO .KR' and links for '데이터셋', '활용사례', '참여마당', and '정보공유'. Below the navigation bar is a search bar with a magnifying glass icon. The main content area features three tabs: 'FILE DATA', 'OPEN API', and 'STANDARD DATA'. Under each tab, there are eight icons representing different data categories: Education, National Land Management, Public Administration, Financial Services, Industry, Social Welfare, Consumer Safety, and Cultural Tourism under FILE DATA; Health Care, Disaster Prevention, Transportation, Environment, Science and Technology, Agriculture, International Trade, and Law under OPEN API; and Health Care, Disaster Prevention, Transportation, Environment, Science and Technology, Agriculture, International Trade, and Law under STANDARD DATA. At the bottom, there are three cards: '국가중점개방 데이터' (National Key Open Data) featuring a map and icons; '공공데이터 활용사례' (Public Data Usage Examples) featuring a blue background with a white cross icon and the text '스마일닥터'; and '2016년 10월 이달의 추천 데이터 국가공간정보' (Recommended Data for October 2016 National Spatial Information) featuring a 3D city model and a magnifying glass icon.

# OpenAPI 제공 사이트

- ◆ [data.seoul.go.kr](http://data.seoul.go.kr) (서울시 열린데이터광장)

The screenshot displays the homepage of the Seoul Open Data Plaza. At the top, there's a navigation bar with three main tabs: '오픈데이터' (Open Data), '데이터서비스' (Data Services), and '참여 · 소통' (Participation · Communication). On the left, there's a section titled '활용 갤러리' (Usage Gallery) featuring a collage of icons related to various data categories like government, culture, environment, health, industry, urban management, welfare, transportation, safety, and education. In the center, there's a search bar with the placeholder '열린데이터를 검색하세요' (Search open data) and a magnifying glass icon. Below the search bar is a grid of icons representing different sectors: 일반행정 (General Administration), 문화관광 (Culture and Tourism), 환경 (Environment), 보건 (Health), 산업경제 (Industry and Economy), 도시관리 (Urban Management), 복지 (Welfare), 교통 (Transportation), 안전 (Safety), and 교육 (Education). To the right, there's a sidebar titled '인기검색어' (Popular Search Terms) with a list of items numbered 1 to 6, each with a small icon and a change indicator (new, 7, 1). At the bottom, there are four large cards representing specific datasets: '서울시 지역구별 인구밀도' (Seoul City District Population Density) showing a map of Seoul districts; '서울시 인구' (Seoul City Population) showing a map of Seoul districts; '서울시 출생시 체중별 ...' (Seoul City Birth Weight ...); and '서울시 1인가구' (Seoul City One-person Household). The 'hci+d lab.' logo is visible in the bottom left corner.

# OpenAPI 제공 사이트

- ❖ [data.seoul.go.kr](http://data.seoul.go.kr) (서울시 열린데이터광장)

The screenshot shows the Seoul Open Data Plaza homepage. At the top, there's a navigation bar with the logo '서울 열린데이터 광장 SEOUL OPEN DATA PLAZA', '오픈데이터' (Open Data), '데이터서비스' (Data Services), and '참여 · 소통' (Participation · Communication). Below the navigation is a search bar with the placeholder '열린데이터를 검색하세요' and a magnifying glass icon. To the right of the search bar is a sidebar titled '인기검색어' (Popular Searches) with links for '지하철' (Subway) and other categories like '- 0', 'new', 'new', '7', '1', and 'new'. On the left, there's a '활용 갤러리' (Usage Gallery) section with a '샘플 URL' (Sample URL) entry for 'Monthly Outbreak Statistics' with the URL [http://openAPI.seoul.go.kr:8088/\(인증키\)/xml/MonthlyOutbreakStatsEng/1/5](http://openAPI.seoul.go.kr:8088/(인증키)/xml/MonthlyOutbreakStatsEng/1/5). Below this is a '시각화 서비스' (Visualization Service) section with a '예제' (Example) button. At the bottom, there are four small charts: '서울시 지역구별 인구밀도' (Seoul City District Population Density), '서울시 인구' (Seoul City Population), '서울시 출생시 체중별 ...' (Seoul City Birth Weight ...), and '서울시 1인가구' (Seoul City One-person Household).

# OpenAPI 제공 사이트

## ♦ data.seoul.go.kr (서울시 열린데이터광장)

요청인자			
변수명	타입	변수설명	값설명
KEY	String(필수)	인증키	OpenAPI에서 발급된 인증키
TYPE	String(필수)	요청파일타입	xml : xml, xml파일 : xmIf, 엑셀파일 : xls, json파일 : json
SERVICE	String(필수)	서비스명	MonthlyOutbreakStatsEng
START_INDEX	INTEGER(필수)	요청시작위치	정수 입력 (페이지 시작번호입니다 : 데이터 행 시작번호)
END_INDEX	INTEGER(필수)	요청종료위치	정수 입력 (페이지 끝번호입니다 : 데이터 행 끝번호)

출력값		
No	출력명	출력설명
공통	list_total_count	총 데이터 건수 (정상조회 시 출력됨)
공통	RESULT.CODE	요청결과 코드 (하단 메세지설명 참고)
공통	RESULT.MESSAGE	요청결과 메시지 (하단 메세지설명 참고)
1	YEAR	YEAR
2	MONTH	MONTH
3	ROADNM	Road Name
4	NO_OUTBK_SIT	No. of Outbreak Situations
5	ACCIDENT	No. of Outbreak Situations by ACCIDENT
6	BREAKDOWN	No. of Outbreak Situations by BREAKDOWN
7	MAINTENACE_WK	No. of Outbreak Situations by Maintenance Work
8	FALL	No. of Outbreak Situations by FALL
9	ETC	No. of Outbreak Situations by ETC

# OpenAPI 제공 사이트

- ❖ [data.seoul.go.kr](http://data.seoul.go.kr) (서울시 열린데이터광장)

The screenshot shows a web page from the Seoul Open Data portal. The left sidebar has categories like '제공기관' (Data Source), '제공 분야' (Category), '제공 항목' (Item), and '제공 일자' (Date). The main content area has a title '요청인자' (Requester) and a message: 'This XML file does not appear to have any style information associated with it. The document tree is shown below.' Below this is a large block of XML code.

```
<?xml version="1.0"?>
<MonthlyOutbreakStatsEng>
  <list_total_count>180</list_total_count>
  <RESULT>
    <CODE>INFO-000</CODE>
    <MESSAGE>정상 처리되었습니다</MESSAGE>
  </RESULT>
  <row>
    <YEAR>2013</YEAR>
    <MONTH>12</MONTH>
    <ROADNM>Internal Circulation Road</ROADNM>
    <NO_OUTBK_SIT>244</NO_OUTBK_SIT>
    <ACCIDENT>100</ACCIDENT>
    <BREAKDOWN>79</BREAKDOWN>
    <MAINTENACE_WK>55</MAINTENACE_WK>
    <FALL>1</FALL>
    <ETC>9</ETC>
  </row>
  <row>
    <YEAR>2013</YEAR>
    <MONTH>12</MONTH>
    <ROADNM>Gangbyeon Expressway</ROADNM>
    <NO_OUTBK_SIT>227</NO_OUTBK_SIT>
    <ACCIDENT>115</ACCIDENT>
    <BREAKDOWN>47</BREAKDOWN>
    <MAINTENACE_WK>63</MAINTENACE_WK>
    <FALL>0</FALL>
    <ETC>2</ETC>
  </row>
  <row>
    <YEAR>2013</YEAR>
    <MONTH>12</MONTH>
    <ROADNM>Bukbu Expressway</ROADNM>
    <NO_OUTBK_SIT>27</NO_OUTBK_SIT>
    <ACCIDENT>10</ACCIDENT>
    <BREAKDOWN>10</BREAKDOWN>
    <MAINTENACE_WK>6</MAINTENACE_WK>
    <FALL>0</FALL>
    <ETC>1</ETC>
  </row>
  <row>
    <YEAR>2013</YEAR>
    <MONTH>12</MONTH>
```

## 4. Advanced Web Crawling 1

---

# Dynamic Websites

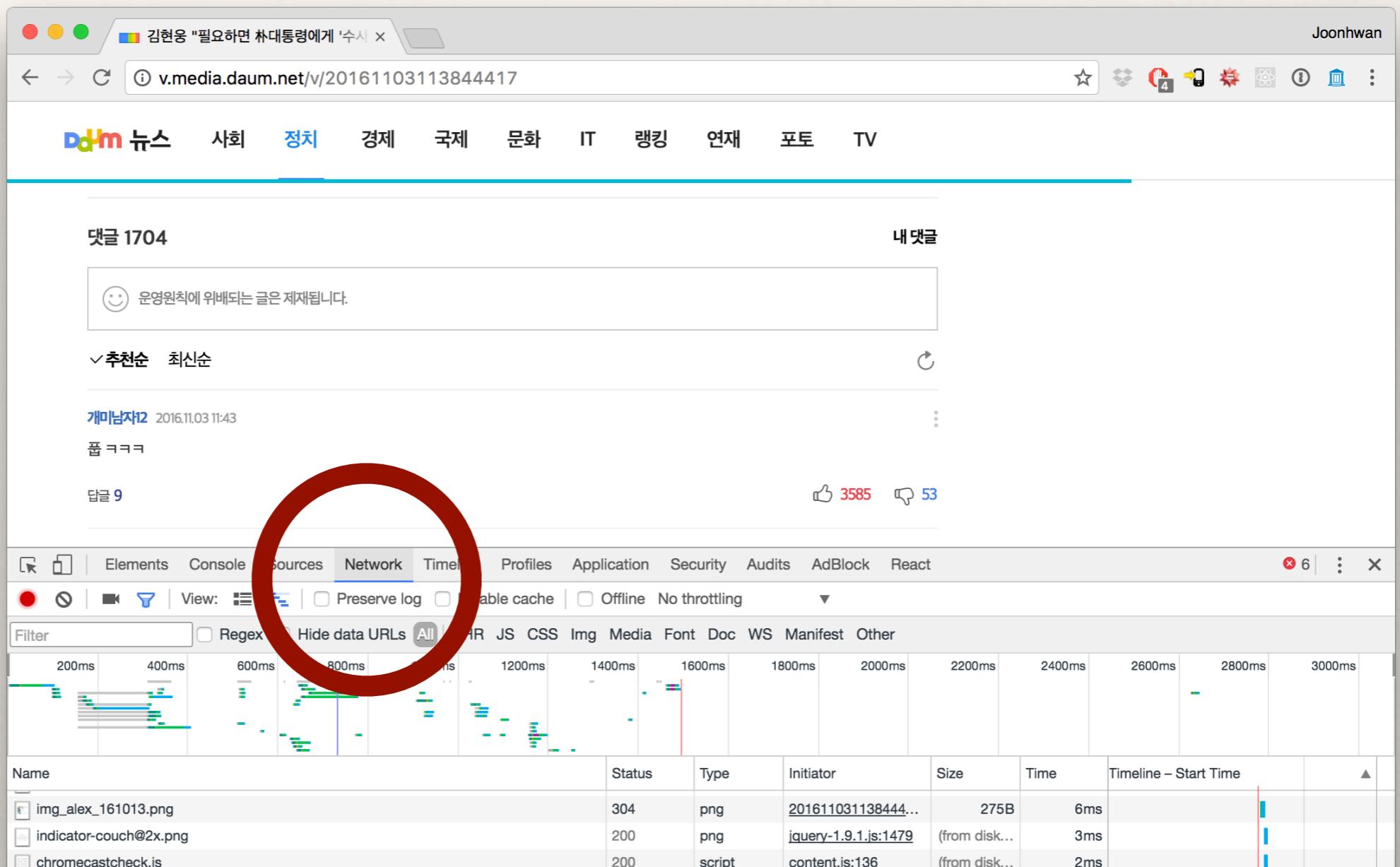
- ❖ 최근에 개발되는 웹사이트
  - ❖ AJAX를 이용하여 데이터를 동적으로 불러온다.  
→ no page reload
  - ❖ 필요한 부분의 데이터만 자바스크립트가 가져와서 해당 부분을 업데이트 (예: facebook, twitter)
  - ❖ 문제점!
    - ❖ 소스에 데이터가 표시되지 않음. (예: media daum 댓글 페이지)

## 데이터 스트림 찾기

- ❖ developer tools → inspector에서 동적으로 유입되는 데이터 소스 찾기
- ❖ 데이터 소스 주소의 패턴 찾기
- ❖ crawl (주로 데이터 소스는 json 포맷)

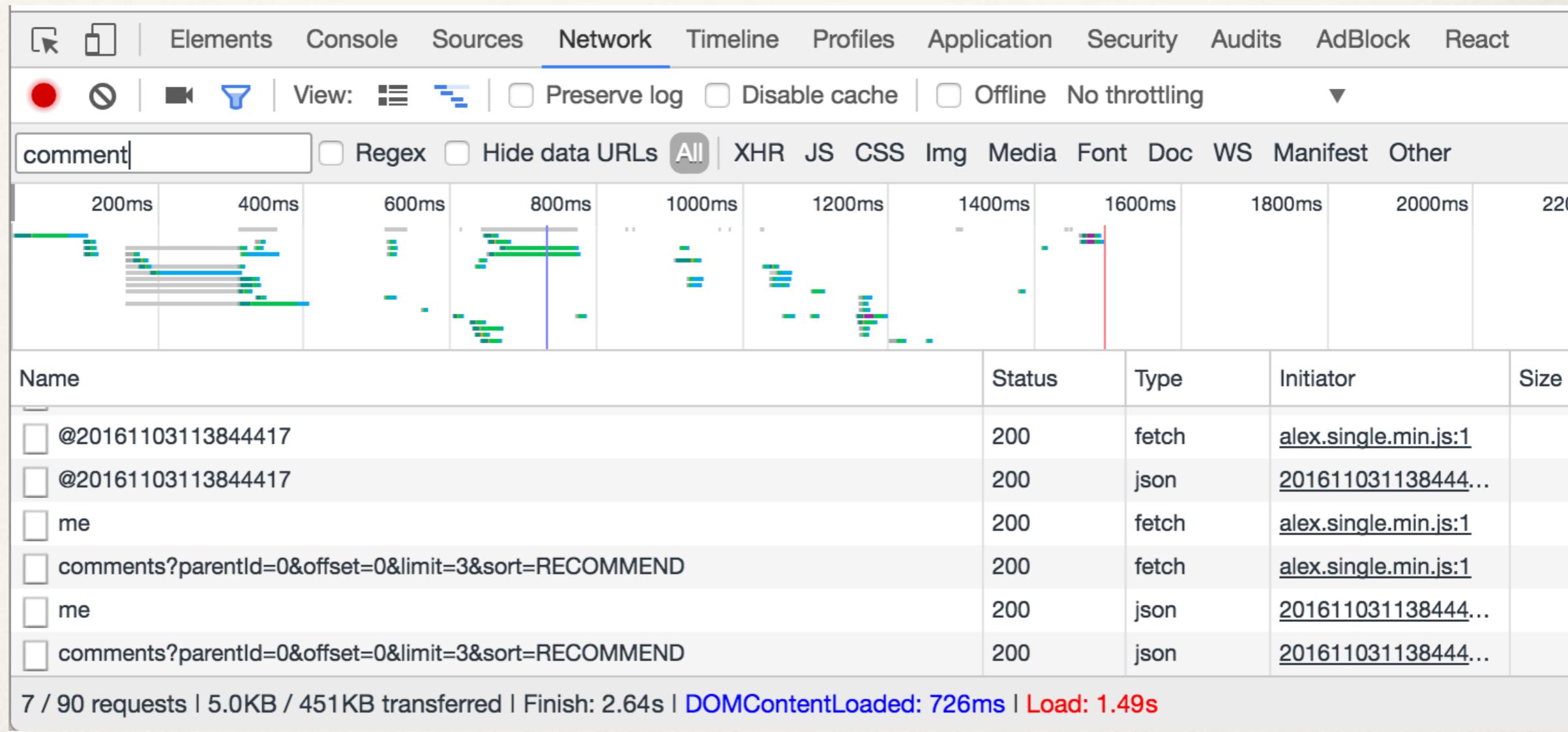
# Developer Tools

- ◆ Chrome (Safari 도 유사): View → Developer → Developer Tools
- ◆ Network 탭 클릭



# Developer Tools

- ♦ Recoding 시작
- ♦ Filter → comment 검색 후 각 스크립트 검사



# Developer Tools

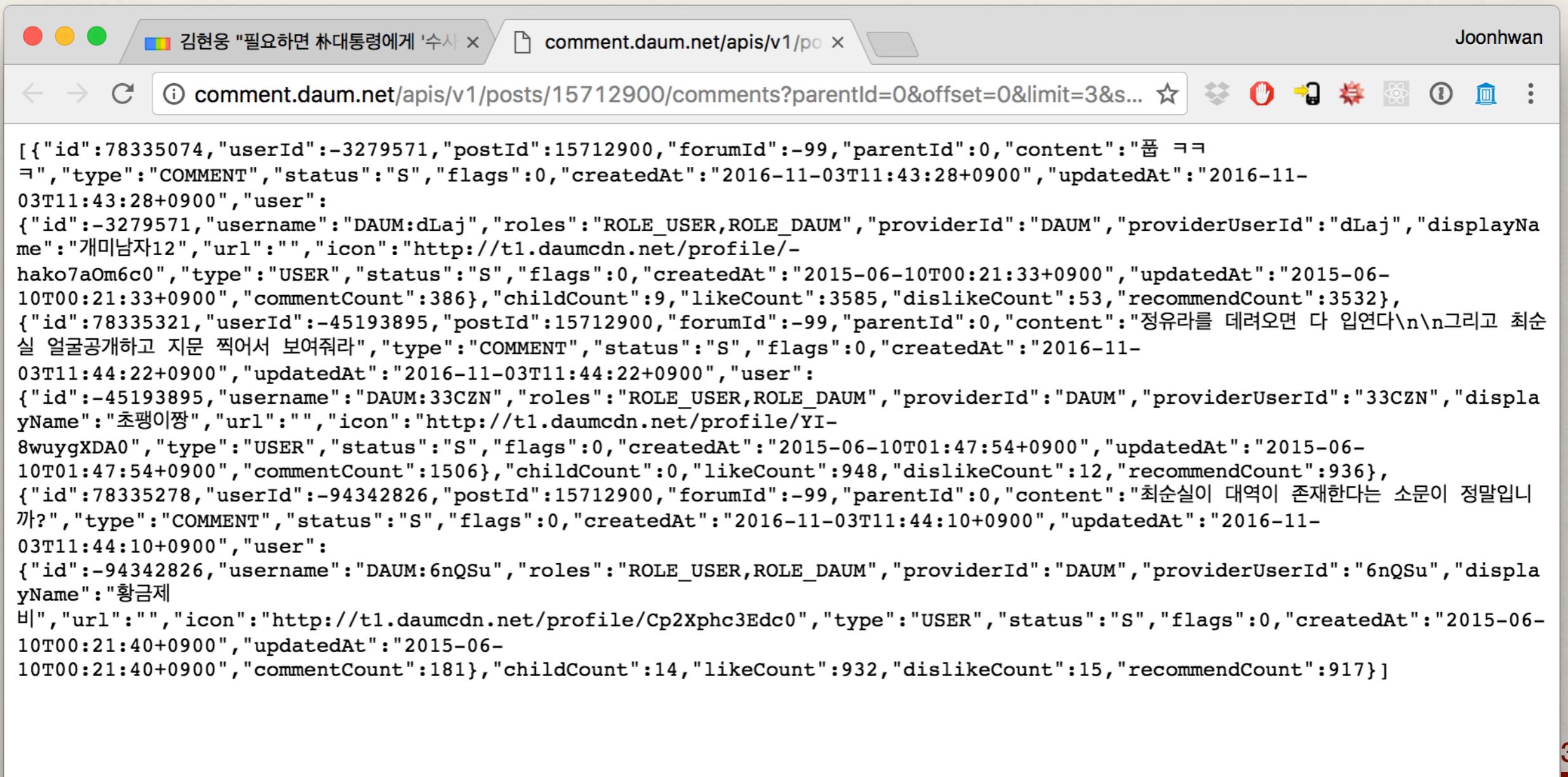
- ♦ 원하는 데이터가 담긴 스크립트를 확인하면 주소 확인  
(새 탭으로 열기+postId 수정)

The screenshot shows the Network tab in the Chrome Developer Tools. A specific request for a comment with postId 15712900 is highlighted and its response body is circled in red. The response body contains the following JSON data:

```
[{"id": 78335074, "userId": -3279571, "postId": 15712900, "forumId": -99, "parentPostId": null, "childCount": 9, "content": "\ud55c \ud55c\ud55c", "createdAt": "2016-11-03T11:43:28+0900", "dislikeCount": 53, "flags": 0, "forumId": -99, "id": 78335074, "likeCount": 3585}
```

# Developer Tools

- [http://comment.daum.net/apis/v1/posts/15712900/comments?  
parentId=0&offset=0&limit=3&sort=RECOMMEND](http://comment.daum.net/apis/v1/posts/15712900/comments?parentId=0&offset=0&limit=3&sort=RECOMMEND)



The screenshot shows a web browser window with the following details:

- Address bar: comment.daum.net/apis/v1/posts/15712900/comments?parentId=0&offset=0&limit=3&s... (with a placeholder for the sort parameter).
- Content area: A JSON array of comments. The first few comments are displayed as follows:

```
[{"id":78335074,"userId":-3279571,"postId":15712900,"forumId":-99,"parentId":0,"content":"쫌 ㅋㅋㅋ","type":"COMMENT","status":"S","flags":0,"createdAt":"2016-11-03T11:43:28+0900","updatedAt":"2016-11-03T11:43:28+0900","user":{"id":-3279571,"username":"DAUM:dLaj","roles":"ROLE_USER,ROLE_DAUM","providerId":"DAUM","providerUserId":"dLaj","displayName":"개미남자12","url":"","icon":"http://t1.daumcdn.net/profile/-hako7aOm6c0","type":"USER","status":"S","flags":0,"createdAt":"2015-06-10T00:21:33+0900","updatedAt":"2015-06-10T00:21:33+0900","commentCount":386,"childCount":9,"likeCount":3585,"dislikeCount":53,"recommendCount":3532}, {"id":78335321,"userId":-45193895,"postId":15712900,"forumId":-99,"parentId":0,"content":"정유라를 데려오면 다 입연다\n\n그리고 최순실 얼굴공개하고 지문 찍어서 보여줘라","type":"COMMENT","status":"S","flags":0,"createdAt":"2016-11-03T11:44:22+0900","updatedAt":"2016-11-03T11:44:22+0900","user":{"id":-45193895,"username":"DAUM:33CZN","roles":"ROLE_USER,ROLE_DAUM","providerId":"DAUM","providerUserId":"33CZN","displayName":"초팽이짱","url":"","icon":"http://t1.daumcdn.net/profile/YI-8wuygXDA0","type":"USER","status":"S","flags":0,"createdAt":"2015-06-10T01:47:54+0900","updatedAt":"2015-06-10T01:47:54+0900","commentCount":1506,"childCount":0,"likeCount":948,"dislikeCount":12,"recommendCount":936}, {"id":78335278,"userId":-94342826,"postId":15712900,"forumId":-99,"parentId":0,"content":"최순실이 대역이 존재한다는 소문이 정말입니까?","type":"COMMENT","status":"S","flags":0,"createdAt":"2016-11-03T11:44:10+0900","updatedAt":"2016-11-03T11:44:10+0900","user":{"id":-94342826,"username":"DAUM:6nQSu","roles":"ROLE_USER,ROLE_DAUM","providerId":"DAUM","providerUserId":"6nQSu","displayName":"황금제비","url":"","icon":"http://t1.daumcdn.net/profile/Cp2Xphc3Edc0","type":"USER","status":"S","flags":0,"createdAt":"2015-06-10T00:21:40+0900","updatedAt":"2015-06-10T00:21:40+0900","commentCount":181,"childCount":14,"likeCount":932,"dislikeCount":15,"recommendCount":917} ]
```

## **5. Advanced Web Crawling 2**

---

# Crawling Using Selenium & Webdriver

- ❖ Selenium & Webdriver
  - ❖ Selenium: Python libraries for automating web browsers
    - ❖ pip install selenium
  - ❖ 사람이 브라우징 하는 것과 동일한 액션을 제공:  
load url, click link
  - ❖ Selenium을 사용하기 위해서는 각 브라우저를 drive 할 수 있는 driver를 설치해야 함.
    - ❖ Firefox driver: <https://github.com/mozilla/geckodriver/releases>
    - ❖ Chrome driver: <https://sites.google.com/a/chromium.org/chromedriver/downloads>

# Using Selenium

- ♦ Sample Code

```
from selenium import webdriver

url = "..."
driver = webdriver.Firefox()
driver.get(url)

element = driver.find_element_by_xpath("//
div[@class='alex_more']")
element.click()

html = driver.page_source
soup = BeautifulSoup(html, "html.parser")

## process soup
```

# xpath의 사용법

- ❖ xpath 의 사용법

- ❖ xpath(path)

- ❖ `driver.find_element_by_xpath('//h1')`
      - ❖ <h1>~</h1> 태그 안의 내용
    - ❖ `...xpath('//div')`
      - ❖ <div>~</div> 태그 안의 내용
    - ❖ `...xpath('//div[@class="footer"]')`
      - ❖ <div class="footer">~</div> 태그 안의 내용
    - ❖ `...xpath('//div[@id="nav"]')`
      - ❖ <div id="nav">~</div> 태그 안의 내용
    - ❖ `...xpath('//div[@class="header"]//a[@id="twitter_anywhere"]')`
      - ❖ <div class="header"><a href="" id="twitter\_anywhere">~</a></div> 태그 안의 내용
    - ❖ `...xpath('//ul[@class="paging"]//li[not(@class="btn btn_next")]')`
      - ❖ <ul class="paging"><li>~</li></ul> 태그 안의 내용 중, <li>의 class 가 btn btn\_next 가 아닌 것들

# Questions?

---