

Web Scraping (2)

Jehyuk Lee

Department of AI, Big Data & Management

Kookmin University

Contents

- 동적 웹페이지 (Dynamic Web Page) 스크래핑
- API 활용
- (실습) 네이버 뉴스 스크래핑
- Web Scraping시 주의 사항

1. 동적 웹페이지 스크래핑

Dynamic Web Page

• 정적 웹페이지 vs 동적 웹페이지

– 정적 웹페이지

- 서버에 미리 저장된 파일(HTML, Javascript등)이 그대로 전달되는 웹페이지
- Client가 요청하면 이 페이지를 그대로 전송(응답)
- 누가, 언제 접속해도 동일한 내용을 보여줌
 - (서버에 저장된 데이터가 변경되지 않는다는 가정하에)

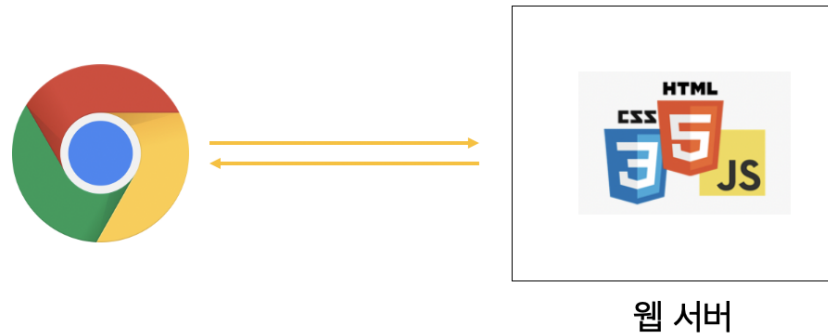
– 동적 웹페이지

- 서버에 있는 데이터를 스크립트에 의해 가공하여 전달하는 웹페이지
- Client의 요청을 해석하여 데이터를 가공하여 전송(응답)
- 누가, 언제 접속하는가에 따라 다른 내용을 보여줌

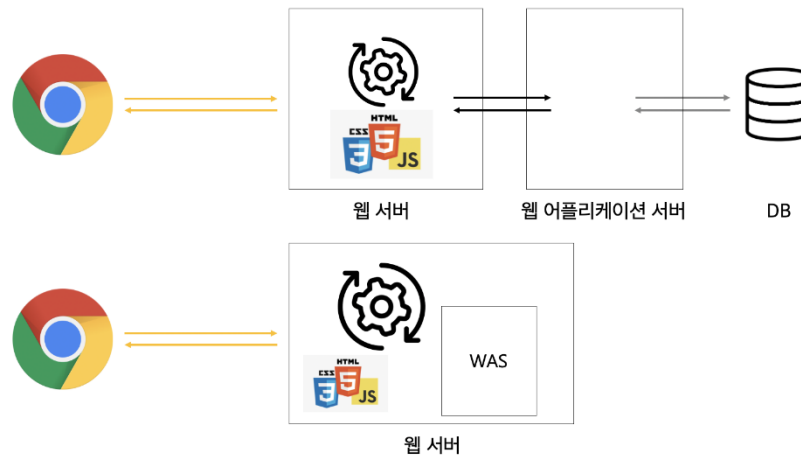
Dynamic Web Page

- 정적 웹페이지 vs 동적 웹페이지

- 정적 웹페이지



- 동적 웹페이지



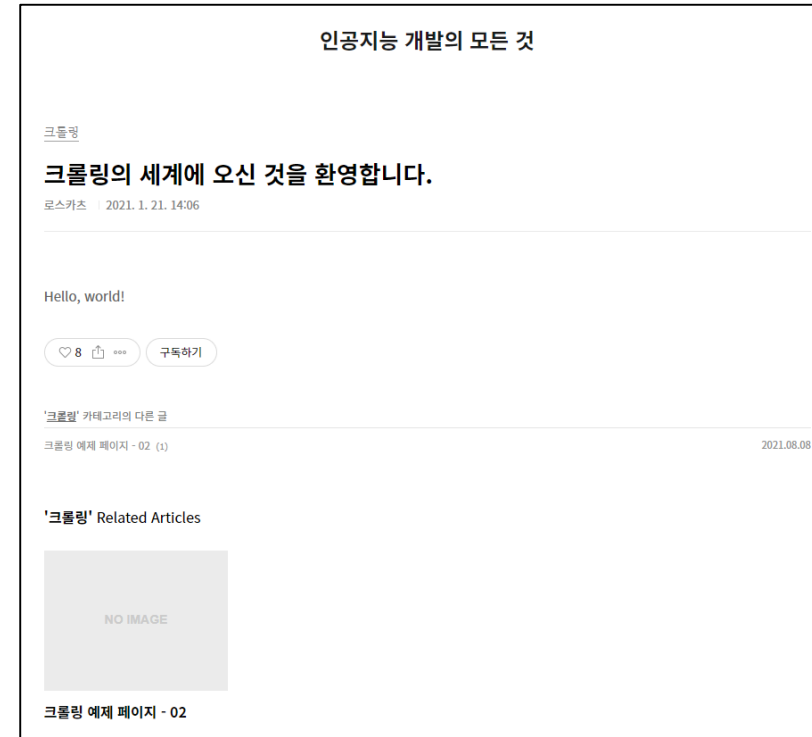
(Source: <https://maily.so/grabnews/posts/ce76c9>)

Dynamic Web Page

- 정적 웹페이지 vs 동적 웹페이지

- 정적 웹페이지

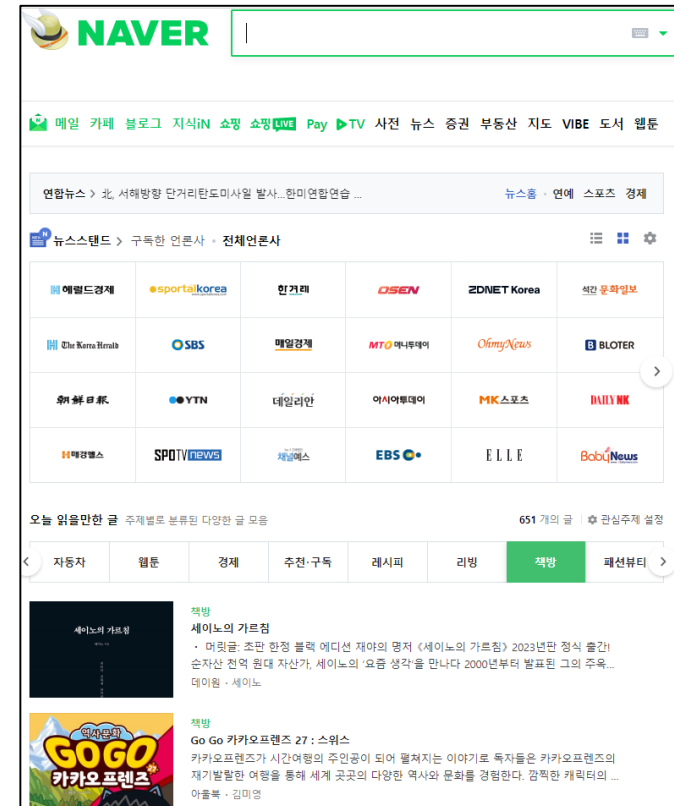
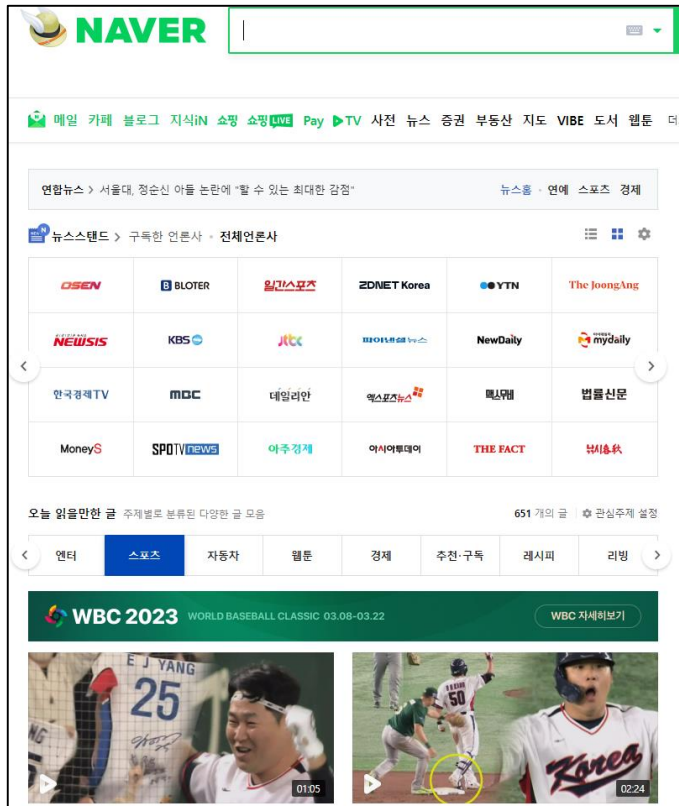
- 동일한 페이지를 보여줍니다



(Source: <https://ai-dev.tistory.com/1>)

Dynamic Web Page

- 정적 웹페이지 vs 동적 웹페이지
 - 동적 웹페이지
 - 보여지는 화면이 계속 바뀝니다



(Source: <https://www.naver.com>)

Dynamic Web Page

- 이제까지 우리는 정적 웹사이트의 정보를 scrap
- 그런데, 동적 웹사이트는 이전과 동일한 방법으로 scrap 불가
 - Why?
- 대다수의 웹사이트는 동적 웹사이트입니다.
 - 동적 웹사이트를 scrap하는 방법에 대해서 알아보시다.

Dynamic Web Page

- 예시) 네이버 웹툰 댓글 가져오기
 - '재벌집 막내아들' 49회의 베스트 댓글들을 가져와봅시다.



< 재벌집 막내아들 49화

본 작품은 가상으로 만들어진 허구의 이야기입니다. 특정 인물이나 단체, 종교, 지명, 사건 등과는 무관합니다.

의견쓰기 265

댓글을 작성하려면 로그인 해주세요.

✓ BEST댓글 전체댓글

클린봇이 악성댓글을 감지합니다.

블랙박스(guda****)

BEST 의료원이랑 인력개발원이 어짜보면 계열사들중 보잘것 없어보이지만 나를 진회장의 큰 그림이 있다는것만 알아두시면 되십니다
2023-05-31 22:57
답글 45 2623 94

배러타임(tkai****)

BEST 표정이 왜 그러냐? 부족한 게냐? 예 많이 부족합니다 한번만 더 올려주세요
2023-05-31 23:00
답글 10 2186 11

(Source: <https://comic.naver.com/webtoon/detail?titleId=800770&no=49>)

Dynamic Web Page

- 예시) 네이버 웹툰 댓글 가져오기

- Source code를 봅시다 (개발자도구 or F12)

- tag에 내용이 있음
 - class 속성: u_cbox_contents



Dynamic Web Page

• 예시) 네이버 웹툰 댓글 가져오기

- 이대로 scraping 코드를 작성하고 보면, 결과가 이상합니다. (결과가 없어요!)

```
url = 'https://comic.naver.com/webtoon/detail?titleId=800770&no=49'
html = urlopen(url)
bs_obj = BeautifulSoup(html, 'html.parser')
results = bs_obj.find_all('span', {'class': "u_cbox_contents"})

print(results)

[]
```

- bs_obj객체를 확인해보니 상당히 많은 내용이 누락되어 있습니다.

```
print(bs_obj)
ape.veta.naver.com/fxview?eu=EU10040901&calp=800770&oj=gWYX8UfU7u0ThzIEDzM100EkIVQ%2B1Mnk:teAdXhY89yNtgBzXYXvAqDDo0v0ZvwI&ac=8832240&src=6667012&evtcd=Y900&x_tj=1397&tb=&oid=&sid1=&sid2=&rk=w1-cvPlrvjZA00WgS-wo8A&elts=w%2F4kI6Xa9gFkLdGwvEDcGA%3D%3D&brs=Y&p0=p&eid=Y900&uaf=p&wtitle=800770"}},
    pplAdsbyUrl: 'https://comic.naver.com/business/proposalGuide'
  },
  exposureStoreBanner: true
};
})();
</script>
<script src="/runtime-d8071ba84cee8e12105b.js" type="text/javascript"></script>
<script src="/vendor-old-5c521b9b95d2e9c81f35.js" type="text/javascript"></script>
<script src="/vendor-react-d37d9c657a271200d9cf.js" type="text/javascript"></script>
<script src="/vendor-react-common-39f644b98f3af612d766.js" type="text/javascript"></script>
<script src="/vendor-common-4c04532899aef03d14c.js" type="text/javascript"></script>
<script src="/vendor-log-feb99cf7b041c7e3b64d.js" type="text/javascript"></script>
<script src="/detail-scrolltoon-ff35e45d434fe3d65c54.js" type="text/javascript"></script>
</body>
</html>
```

tag내의
class 속성값이 u_cbox_contents인
부분이 아예 누락됨

Dynamic Web Page

- 다른 예시) 스포츠팀의 승패 기록 계산하기

- 두 종류의 source 코드를 봅시다

- 페이지 소스 보기 – 웹 서버에서 최초로 전달받은 내용
 - 개발자 도구 – 기존 html코드에서 (데이터 등을) 새로 가공한 내용

- 참고

- 네이버 웹툰에서
페이지 소스보기가 막혀
다른 예제로 보여드립니다

```
<div class="container__fsbody" id="fsbody">
<div id="live-table">
  <script>
    document.body.classList.toggle("loading", true);
  </script>
  <div class="loadingOverlay">
  <div class="loadingAnimation">
    <div class="loadingAnimation__text">Loading...</div>
  </div>
</div>
<div class="sk">
  <div class="sk__bl">
    <div class="sk__w">
      <div></div>
      <div></div>
      <div></div>
      <div></div>
      <div></div>
      <div></div>
      <div></div>
      <div></div>
      <div></div>
      <div></div>
    </div>
    <div class="sk__h"></div>
  </div>
</div>
```

(페이지 소스 보기)

가공 후

```
<div class="container__fsbody" id="fsbody">
<div id="live-table">
  <section class="event event--live event--summary">
    <div class="leagues--live">
      <div class="tabs tabs--live">
        <div class="tabs__group">
          <div class="tabs__ear">오늘의 경기</div>
        </div>
      </div>
      <div class="sportName basketball">
        <div class="event__header top event__header--noExpand">
          <div id="g_3_MVd8yftL" title="경기세부사항을 보려면 클릭" elementtiming="SpeedCurveFRP"
            class="event__match event__match--last event__match--twoLine">
            ::before
            <div class="eventSubscriber eventSubscriber__star eventSubscriber__star--event">
            </div>
          </div>
          <a class="event__more event__more--static" href="/kr/team/golden-state-warriors/SxUtXqch/c
            esults/">더 많은 경기 보기</a>
        </div>
      </div>
    </section>
    <section class="event event--summary">
    <div class="notificationsDialog">
    <div class="loadingOverlay">
    <div class="sk">
      <div class="sk__bl">
        <div class="sk__l">
        <div class="sk__w">
        <div class="sk__h">
```

(개발자 도구)

Dynamic Web Page

• 예시) 스포츠팀의 승패 기록 계산하기

- 페이지 소스를 검색해서는 찾을 수 없는 html코드들이 동적 웹 페이지 코드들
- 이러한 내용을 가져오기 위해서는 별도의 도구가 필요 → **Selenium**

- 참고

네이버 웹툰에서
페이지 소스보기가 막혀
다른 예제로 보여드립니다



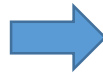
(Source: <https://www.livesport.com>)

Selenium을 활용한 동적 웹페이지 scraping

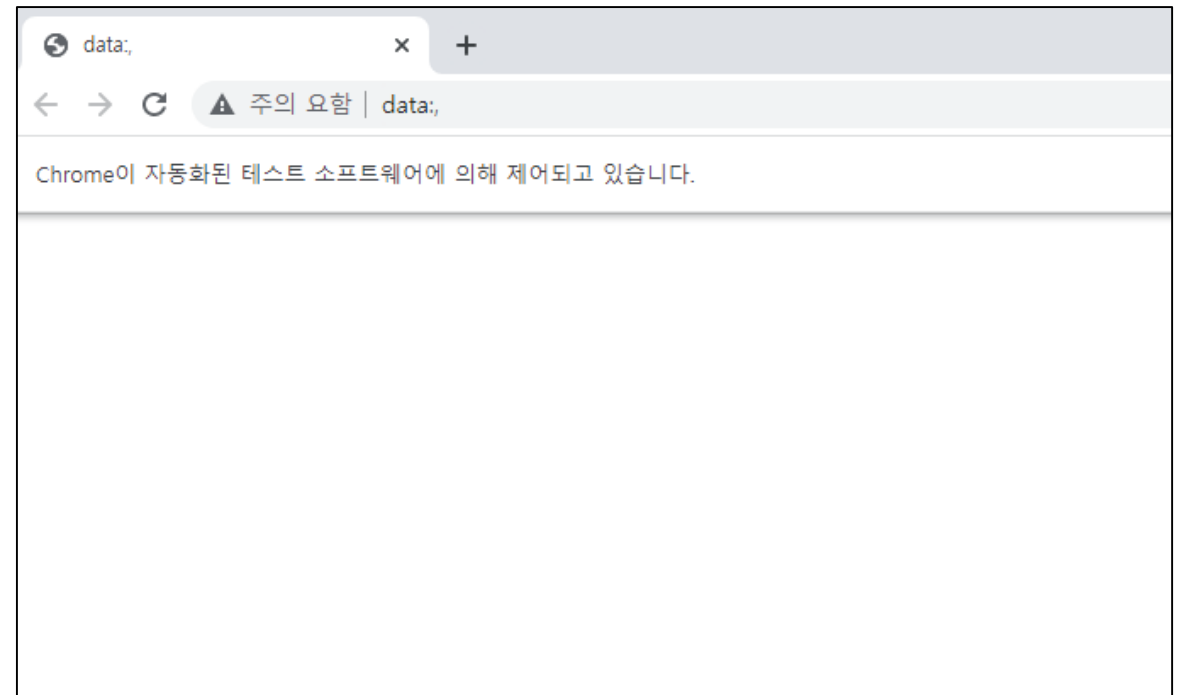
- Jupyter notebook을 실행하고 selenium 불러오기
 - Webdriver 클래스의 객체인 driver 생성

```
from selenium import webdriver
```

```
driver = webdriver.Chrome('D:\MyWorkspace\chromedriver.exe')
```



앞에서 기억한 폴더의 절대 경로에
\chromedriver.exe를 붙인 경로

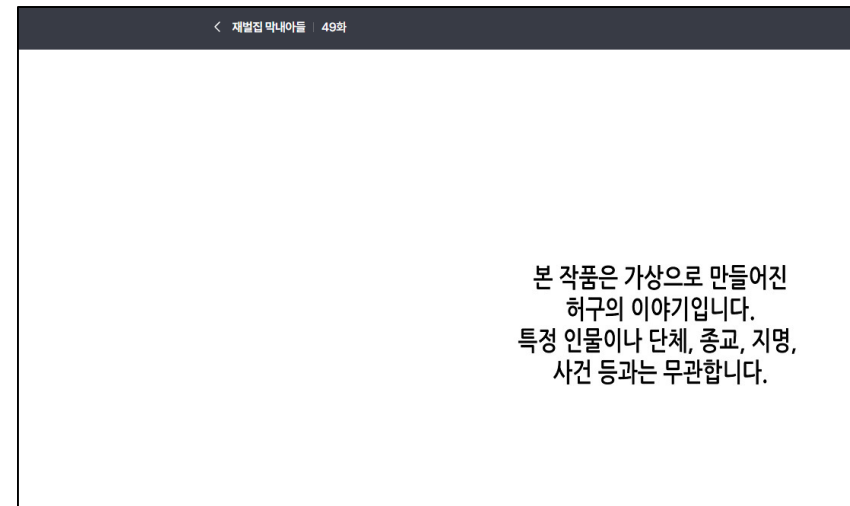
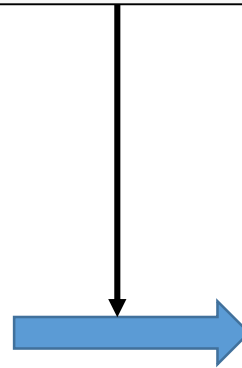
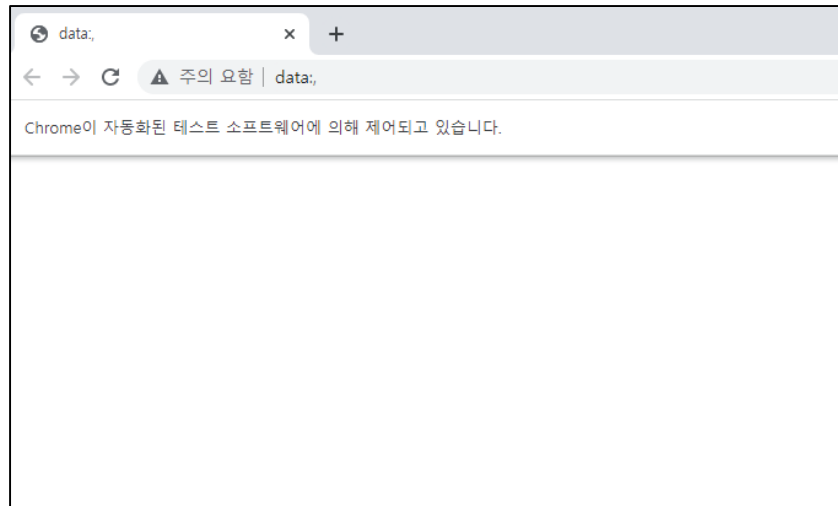


이 빈페이지에서 동적 웹 페이지를 불러올 예정

Selenium을 활용한 동적 웹페이지 scraping

- driver의 `get()` method에 원하는 웹페이지 주소를 전달하여 접속

```
# 페이지 로드를 위해 기다리는 시간  
# Wait for loading...  
driver.implicitly_wait(3)  
  
# scraping하려는 웹페이지 주소를 get()에 전달  
# Enter the webpage address into the get() method  
url = 'https://comic.naver.com/webtoon/detail?titleId=800770&no=49'  
driver.get(url)
```



Selenium을 활용한 동적 웹페이지 scraping

- driver의 page_source 속성에서 문자열 형식의 html 파일내용 확인

```
In [36]: page = driver.page_source
```

```
In [37]: type(page)
```

```
Out[37]: str
```

```
In [38]: page
```

```
cbox_id_area"><span class="u_cbox_id">(guda****)</span></span></span></span></span><span class="u_cbox_info_sub"><span class="u_cbox_w
ork_sub"><a href="#" class="u_cbox_btn_open" data-action="list#toggleButtons" data-param="@event"><span class="u_cbox_ico_open"></span>
<span class="u_cbox_in_open">옵션 열기</span></a><span class="u_cbox_work_box"><span class="u_cbox_work_inner"><a href="#" class="u_c
box_btn_userreport" data-action="report#request" data-param="commentNo:₩'455014378₩',objectId:₩'800770_49₩'" data-log="RPC.report" tar
get="_blank"><span class="u_cbox_ico_block"></span><span class="u_cbox_in_block">신고</span></a><a href="#" class="u_cbox_btn_userbloc
k" data-action="userBlock#requestBlock" data-param="commentNo:₩'455014378₩',objectId:₩'800770_49₩',nickName:₩'블랙박스₩',maskedId:₩'gu
da****₩',idNo:₩'21Yd₩',profileImageSrc:₩'https://ssl.pstatic.net/static.cbox/20230829152227/img/img_default_profile.gif₩'"><span clas
s="u_cbox_ico_block"></span><span class="u_cbox_in_block">차단</span></a></span></span></span></span></div><div class="u_cbox_text_wra
p"><span class="u_cbox_ico_best" style="">BEST</span><span class="u_cbox_contents" style="" data-lang="ko">의료원이랑 인력개발원이 어
찌보면 계열사들중 보잘것 없어보이지만 나름 진회장의 큰 그림이 있다는것만 알아두시면 되십니다</span><span class="u_cbox_ico_exclamatio
n" style="display:none;"></span><span class="u_cbox_delete_contents" style="display:none;">내가 차단한 이용자의 댓글입니다.</span></di
v><div class="u_cbox_info_base"><span class="u_cbox_date" data-value="2023-05-31T22:57:26+0900">2023-05-31 22:57</span></div><div clas
s="u_cbox_tool"><a href="#" role="button" aria-expanded="false" class="u_cbox_btn_reply" data-action="reply#toggle" data-param="455014
378" data-log="RPC.replyopen#RPC.replyclose"><strong class="u_cbox_reply_txt">답글</strong><span class="u_cbox_reply_cnt">47</span></a>
<div class="u_cbox_recomm_set"><strong class="u_vc">좋아요/싫어요</strong><a href="#" data-action="vote" data-param="mine:false,comme
ntNo:₩'455014378₩',voteStatus:₩'SYMPATHY₩',objectId:₩'800770_49₩',ticket:₩'comic₩'" data-log="RPC.sym#RPC.unsym" class="u_cbox_btn_rec
omm"><span class="u_cbox_ico_recomm">좋아요</span><em class="u_cbox_cnt_recomm">2665</em></a><a href="#" data-action="vote" data-param
="mine:false,commentNo:₩'455014378₩',voteStatus:₩'ANTIPATHY₩',objectId:₩'800770_49₩',ticket:₩'comic₩'" data-log="RPC.dis#RPC.undis" cl
ass="u_cbox_btn_unrecomm"><span class="u_cbox_ico_unrecomm">싫어요</span><em class="u_cbox_cnt_unrecomm">96</em></a></div></div><div><span
class="u_cbox_comment_frame"><span class="u_cbox_ico_tit"></span><span class="u_cbox_comment_frame_top"><span class="u_cbox_comment_bg
```

tag내의
class 속성값이
u_cbox_contents인
부분 확인 가능

Selenium을 활용한 동적 웹페이지 scraping

- driver로 불러온 웹페이지에서 BeautifulSoup을 활용한 scraping

```
from bs4 import BeautifulSoup
```

```
bs_obj = BeautifulSoup(page, 'html.parser')
```

```
bs_obj
```

```
og=RPC.report" data-param=commentNo: 455014378 ,objectId: 800770_49" href="#" target="_blank"><span class="u_cbox_ico_block"></span>
<span class="u_cbox_in_block">신고</span></a><a class="u_cbox_btn_userblock" data-action="userBlock#requestBlock" data-param="comment
No: '455014378',objectId: '800770_49',nickName: '블랙박스',maskedId: 'guda****',idNo: '21YaD',profileImageSrc: 'https://ssl.pstatic.net/stat
ic.cbox/20230829152227/img/img_default_profile.gif'" href="#"><span class="u_cbox_ico_block"></span><span class="u_cbox_in_block">차단
</span></a></span></span></span></div><div class="u_cbox_text_wrap"><span class="u_cbox_ico_best" style="">BEST</span><span cla
ss="u_cbox_contents" data-lang="ko" style="">의료원이랑 인력개발원이 어찌보면 계열사들중 보잘것 없어보이지만 나름 진회장의 큰 그림이
있다는것만 알아두시면 되십니다</span><span class="u_cbox_ico_exclamation" style="display:none;"></span><span class="u_cbox_delete_cont
ents" style="display:none;">내가 차단한 이용자의 댓글입니다.</span></div><div class="u_cbox_info_base"><span class="u_cbox_date" data-
value="2023-05-31T22:57:26+0900">2023-05-31 22:57</span></div><div class="u_cbox_tool"><a aria-expanded="false" class="u_cbox_btn_repl
y" data-action="reply#toggle" data-log="RPC.replyopen#RPC.replyclose" data-param="455014378" href="#" role="button"><strong class="u_c
box_reply_txt">답글</strong><span class="u_cbox_reply_cnt">47</span></a><div class="u_cbox_recomm_set"><strong class="u_vc">좋아요/싫
어요</strong><a class="u_cbox_btn_recomm" data-action="vote" data-log="RPC.sym#RPC.unsym" data-param="mine:false,commentNo:'45501437
8',voteStatus:'SYMPATHY',objectId:'800770_49',ticket:'comic'" href="#"><span class="u_cbox_ico_recomm">좋아요</span><em class="u_cbox_
cnt_recomm">2665</em></a><a class="u_cbox_btn_unrecomm" data-action="vote" data-log="RPC.dis#RPC.undis" data-param="mine:false,comment
No:'455014378',voteStatus:'ANTIPATHY',objectId:'800770_49',ticket:'comic'" href="#"><span class="u_cbox_ico_unrecomm">싫어요</span><em
class="u_cbox_cnt_unrecomm">96</em></a></div></div><span class="u_cbox_comment_frame"><span class="u_cbox_ico_tip"></span><span class
="u_cbox_comment_frame_top"><span class="u_cbox_comment_bg_r"></span><span class="u_cbox_comment_bg_l"></span></span></span><span class="u_cb
ox_comment_frame_bottom"><span class="u_cbox_comment_bg_r"></span><span class="u_cbox_comment_bg_l"></span></span></span></div></div><
div class="u_cbox_reply_area" style="display:none;"></div></li></li><li class="u_cbox_comment cbox_module__comment_455015145 _user_id_no_2a
wBg" data-info="commentNo: '455015145',deleted:false,best:true,visible:true,secret:false,manager:false,mine:false,report:undefined,blin
```

Selenium을 활용한 동적 웹페이지 scraping

- 태그의 class 속성 값이 'u_box_contents'인 부분 추출

```
results = bs_obj.find_all('span', {'class':"u_box_contents"})
```

```
results
```

```
[<span class="u_box_contents" data-lang="ko" style="">의료원이랑 인력개발원이 어찌보면 계열사들중 보잘것 없어보이지만 나름 진회장의 큰 그림이 있다는것만 알아두시면 되십니다</span>,  
<span class="u_box_contents" data-lang="ko" style="">표정이 왜 그러냐? 부족한 게냐?  
예 많이 부족합니다
```

```
한편만 더 올려주세요</span>,  
<span class="u_box_contents" data-lang="ko" style="">시간이 할아버지편은 아닌데 막내손주한테는 치트키를 주더라</span>,  
<span class="u_box_contents" data-lang="ko" style="">인력개발원은 모든 인재가 모이는곳이니 원하는 사람들과 연을 쌓기 제일 좋아보이고  
인재에 대해 누구보다 빨리 선점할수 있지 않을까 싶군</span>,  
<span class="u_box_contents" data-lang="ko" style="">진짜가치있는것들은 전부 가장이쁜손주에게 준다는것..이런분야에 어둡고 아들을사랑  
하는 아버지의모습</span>,  
<span class="u_box_contents" data-lang="ko" style="">국내 최고 재벌가의 의료원과 인력 개발원이 황금알을 낳는 거워지..의료원은 전국에  
서 정재계 가릴것 없이 VIP들의 모든 정보와 인력개발원은 차세대 임원진들 즉 회사를 장악하기 위한 측근들 양성소인데 이게 알짜중에 알짜임</  
span>,  
<span class="u_box_contents" data-lang="ko" style="">돈은 도준이가 많아 가지고 있으니 사람을 선물한거네요 할배가..이번에 받은 재단으  
로 사람을 얻었고..  
돈은 도준이가 가지고있고..  
아버지랑 형은 미디어쪽에서 착실하게 커리어 쌓구있고..  
점점 도준이가 후계구도 싸움을위한 패를 만들고 있는것 같네요..  
나중에 망나니 형이 저렇게 노는것도 미디어에 던지니 마니하면서 카드로 써먹들듯 ㅋㅋ</span>,  
<span class="u_box_contents" data-lang="ko" style="">홍 회장이 겁은 먹어도 마냥 호락호락하진 않네요. 괜히 제1의 신문사 회장이 아니군  
요.</span>,  
<span class="u_box_contents" data-lang="ko" style="">??? : 저는 여기 분당땅으로 주세요~</span>,  
<span class="u_box_contents" data-lang="ko" style="">달려로 쿠키 넌테니 5개 정도만 더 보고싶다..</span>,  
<span class="u_box_contents" data-lang="ko" style="">맛있어 맛있어여기는 항상 맛있어</span>,  
<span class="u_box_contents" data-lang="ko" style="">크흐 다른세계 아부지랑 다르지! 크흐 이게 아부지지!</span>,  
<span class="u_box_contents" data-lang="ko" style="">스피드웨건: 소설 81화 </span>,  
<span class="u_box_contents" data-lang="ko" style="">소설 81화 43%부터 이어보기 가능</span>,
```

Selenium을 활용한 동적 웹페이지 scraping

- 태그의 class 속성 값이 'u_box_contents'인 부분 추출
 - 좀 더 이쁘게 뽑아봅시다

```
[x.text.replace('\\\\n',' ') for x in results]
```

['의료원이랑 인력개발원이 어찌보면 계열사들중 보잘것 없어보이지만 나름 진회장의 큰 그림이 있다는것만 알아두시면 되십니다',
'표정이 왜 그러냐? 부족한 게냐? 예 많이 부족합니다 한편만 더 올려주세요',
'시간이 할아버지편은 아닌데 막내손주한테는 치트키를 주더라',
'인력개발원은 모든 인재가 모이는곳이니 원하는 사람들과 연을 쌓기 제일 좋아보이고 인재에 대해 누구보다 빨리 선점할수 있지 않을까 싶군',
'진짜가치있는것들은 전부 가장이쁜손주에게 준다는것..이런분야에 어둡고 아들을사랑하는 아버지의모습',
'국내 최고 재벌가의 의료원과 인력 개발원이 황금알을 낳는 거워지..의료원은 전국에서 정재계 가릴것 없이 VIP들의 모든 정보와 인력개발원은 차세대 임원진들 즉 회사를 장악하기 위한 측근들 양성소인데 이게 알짜중에 알짜임',
'돈은 도준이가 많아 가지고 있으니 사람을 선물한거네요 할배가..이번에 받은 재단으로 사람을 얻었고.. 돈은 도준이가 가지고있고.. 아버지랑 형은 미디어쪽에서 착실하게 커리어 쌓구있고.. 점점 도준이가 후계구도 싸움을위한 패를 만들고 있는것 같네요.. 나중에 망나니 형이 저렇게 노는것도 미디어에 던지니 마니하면서 카드로 써먹들듯 ㅋㅋ',
'홍 회장이 겁은 먹어도 마냥 호락호락하진 않네요. 괜히 제1의 신문사 회장이 아니군요.',
'??? : 저는 여기 분당땅으로 주세요~',
'달려로 쿠키 날테니 5개 정도만 더 보고싶다..',
'맛있어 맛있어여기는 항상 맛있어',
'크흐 다른세계 아버지랑 다르지! 크흐 이게 아버지지!',
'스피드웨건: 소설 81화 ',
'소설 81화 43%부터 이어보기 가능',

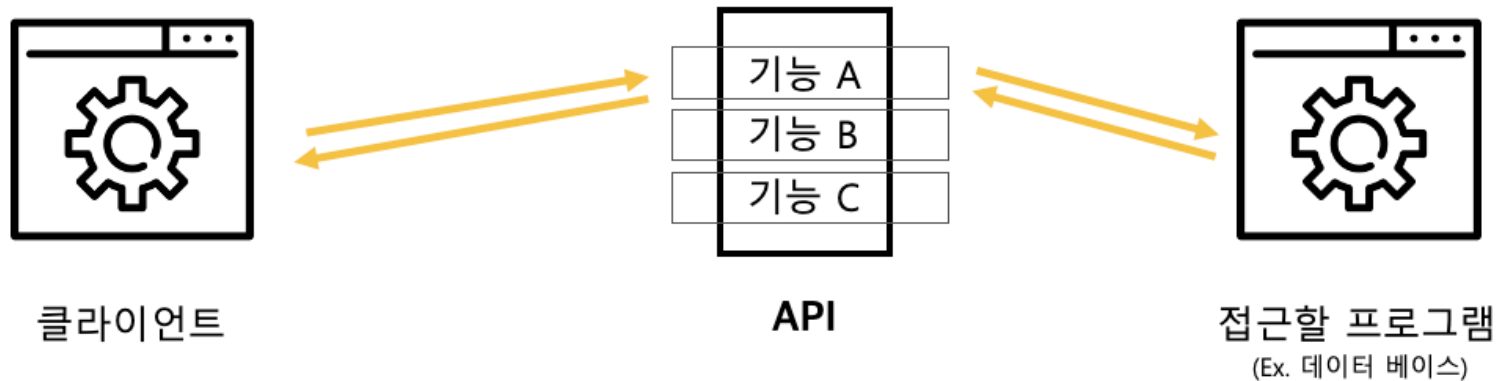
2. API활용

What is API?

- Application Programming Interface (API)

- 프로그램 간 데이터를 주고 받기 위한 방법

프로그램과 프로그램을 연결시켜주는 매개체

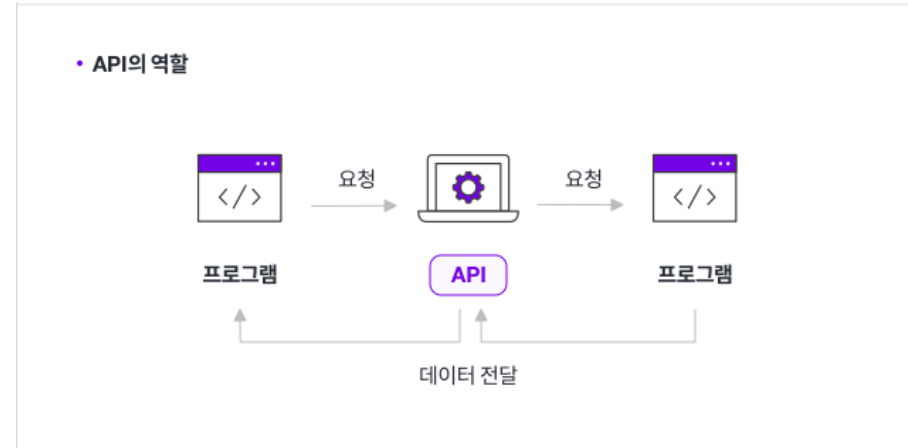
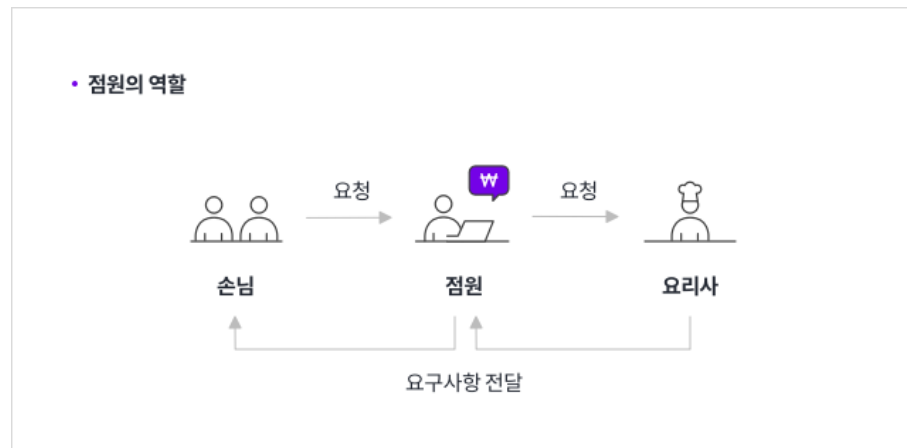


(Source: <https://maily.so/grabnews/posts/b2341a>)

What is API?

- **Application Programming Interface (API)**

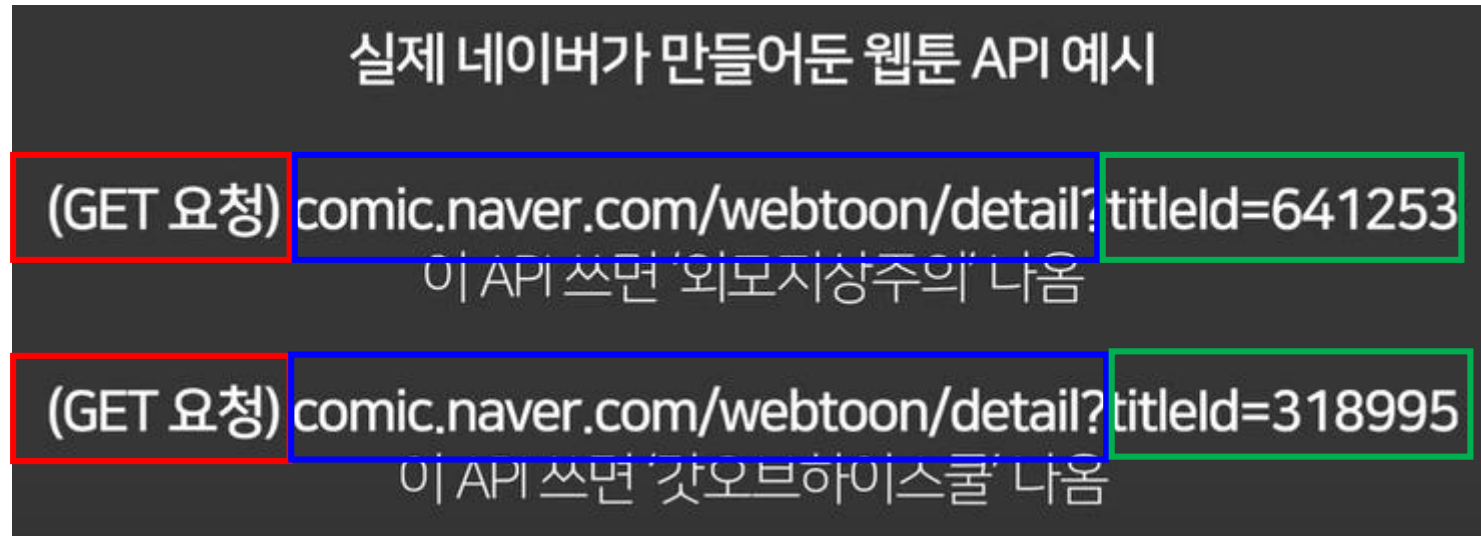
- 프로그램 간 데이터를 주고 받기 위한 방법
- 레스토랑에서 점원과 같은 역할을 수행
 - **점원:** 손님에게 메뉴를 '요청'받고 이를 요리사에게 '요청' → 요리사의 결과물을 전달
 - **API:** 프로그램에게 데이터를 '요청'받고 이를 타 프로그램에 '요청' → 데이터를 전달



(Source: <https://blog.wishket.com/api%EB%9E%80-%EC%89%BD%EA%B2%8C-%EC%84%A4%EB%AA%85-%EA%B7%B8%EB%A6%B0%ED%81%B4%EB%9D%BC%EC%9D%B4%EC%96%B8%ED%8A%B8/>)

What is API?

- Application Programming Interface (API)
 - API가 가져야 할 내용



1. 요청방식
(GET, POST 등)

2. 요청할 자료
(endpoint)

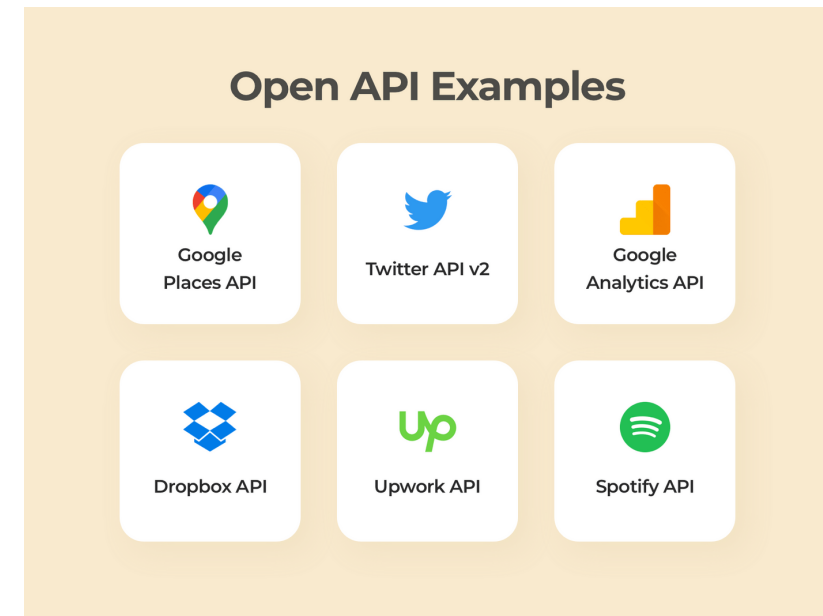
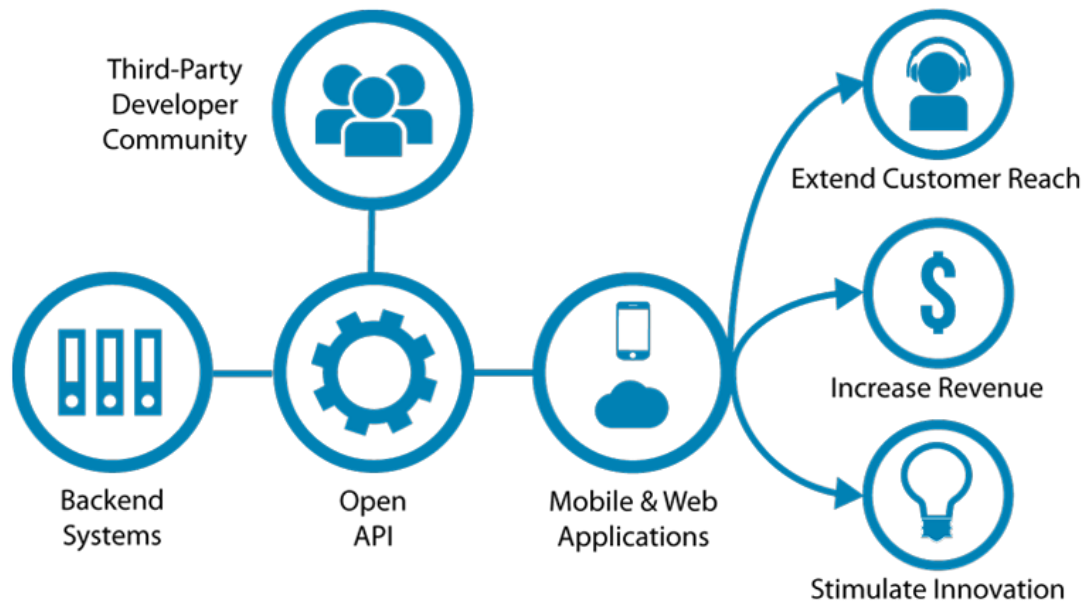
3. 자료요청에 필요한 추가정보
(검색 조건, API key 등)

(Source: <https://www.youtube.com/watch?v=ckSdPNKM2pY>)

What is API?

- **Open API**

- 개발자라면 누구나 무료로 사용 가능하도록 공개된 API
- Open API를 배포하여 자사 서비스를 활용한 영역 확대
 - 결국, 자사의 영향력을 높일 수 있음



(Source: https://ko.wikipedia.org/wiki/%EC%98%A4%ED%94%88_API, <https://velog.io/@gil0127/API%EB%9E%80-%EA%B0%9C%EB%85%90-%EC%A0%95%EB%A6%AC%EC%99%80-%ED%8F%AC%ED%8A%B8%ED%8F%B4%EB%A6%AC%EC%98%A4%EC%97%90-%EC%9C%A0%EC%9A%A9%ED%95%9C-%EB%8C%80%EB%B0%95-%EC%82%AC%EC%9D%B4%ED%8A%B8-%EA%B3%B5%EC%9C%A0>)

What is API?

• API를 활용한 비즈니스 예시

 **CLOVA OCR** Update


인쇄물 속 글자를 추출하여 디지털 데이터로 변환해 주는 서비스

[이용 신청하기](#) [요금 계산하기](#) 

특징

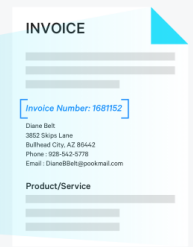

상세 기능

요금

사용 가이드 

이미지 속 문자 추출하여 컴퓨터 데이터로 변환

복잡하고 다양한 문서나 이미지 속 문자를 추출하여 데이터화하고 관리할 수 있는 서비스입니다.



정확한 데이터 추출

Optical character recognition(OCR, 광학 문자 인식)은 이미지(사진)에서 글자 위치를 찾고 인식하여 컴퓨터 텍스트로 변환하는 기술입니다. CLOVA OCR은 OCR 분야에서 가장 권위 있는 글로벌 챌린지인 ICDAR2019 47개 분야에서 1위, CVPR 및 ICCV 국제 학회 논문예 선정되는 등 독보적인 기술력을 자랑합니다. 특히 인식 대상의 레이아웃을 분석하고 글자를 읽는 순서와 방향을 추정하여 문자를 인식할 수 있습니다. 또한 국선으로 배열되거나 기울어진 문자, 필기체 등도 인식할 수 있어 정확한 데이터를 추출할 수 있습니다.

요금 안내

요금 옵션 설정하기

필터를 적용하여 요구사항에 맞게 가격 옵션을 설정할 수 있습니다.

한국 ▼

KRW - ₩ ▼

General OCR

| 구분 | 서비스 | 과금구간 | 과금기준 | 요금 |
|----------------|-------|---------|-------|------|
| 프리미엄 - General | 글자 추출 | 100회 이하 | 호출수 당 | 무료 |
| 프리미엄 - General | 글자 추출 | 100회 초과 | 호출수 당 | 3 원 |
| 프리미엄 - General | 표 추출 | 100회 이하 | 호출수 당 | 무료 |
| 프리미엄 - General | 표 추출 | 100회 초과 | 호출수 당 | 22 원 |

(VAT 별도)

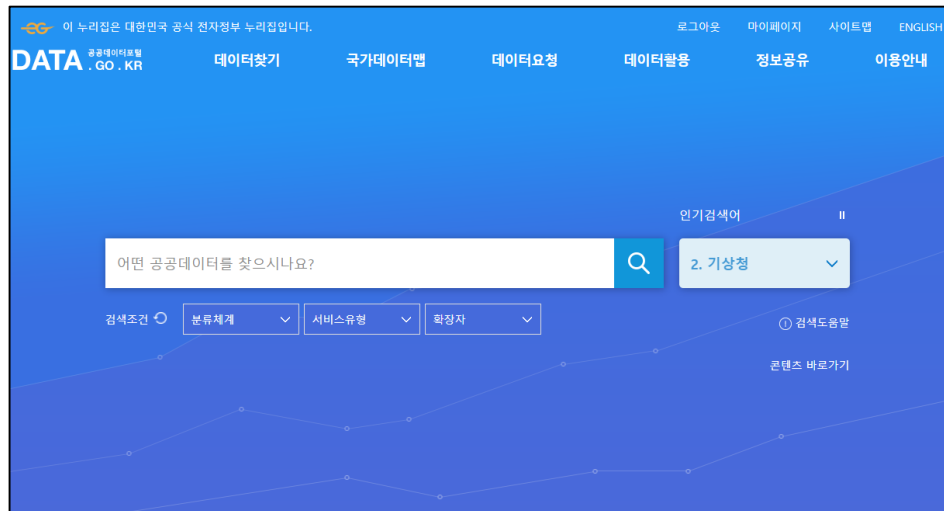
• 표 추출 이용시 글자 추출이 동시에 호출되며 1건당 25원이 청구됩니다.

• 최대 100만 건까지 호출 가능하며, 대용량 사용을 원하시는 경우 고객센터로 문의해 주시기 바랍니다.

(Source: <https://www.ncloud.com/product/aiService/ocr>)

API를 활용한 데이터 수집

- 예제: 데이터 포털에서 api를 활용하여 부동산 거래내역 데이터 수집
 - <https://www.data.go.kr>
 - 회원 가입 → 로그인 → '부동산 거래 현황 통계 조회 서비스' 검색 → 활용신청
 - 신청 상세 내역을 적당히 작성하고 제출



(Source: <https://www.data.go.kr>)

API를 활용한 데이터 수집

- 예제: 데이터 포털에서 api를 활용하여 부동산 거래내역 데이터 수집
 - <https://www.data.go.kr>
 - 회원 가입 → 로그인 → '부동산 거래 현황 통계 조회 서비스' 검색 → 활용신청
 - 신청 상세 내역을 적당히 작성하고 제출

개발계정

신청 0건 >

신청중인 단계

· 보류 0건

· 반려 0건

활용 1건 >

승인되어 활용중인 단계

· 변경신청 0건

중지0건 >

중지신청하여 운영이 중지된 단계

상세검색 열기 ▾

총1건

폐기된 목록 포함 보기 ☐ OFF

공공행정

한국부동산원

활용신청 [승인] 한국부동산원_부동산 거래 현황 통계 조회 서비스

신청일 2023-03-15 만료예정일 2025-03-15

마이페이지

인증키 발급현황

오픈API ▾

개발계정

운영계정

인증키 발급현황

총 1건

| 구분 | 발급일자 | 재발급여부 |
|-------|------------|-------|
| plain | 2023/03/15 | new |

DATA

나의 문의 >

나의 관심

나의 제공신청

나의 분쟁조정

회원정보 수정 >

(Source: <https://www.data.go.kr>)

API를 활용한 데이터 수집

• 예제: 데이터 포털에서 api를 활용하여 부동산 거래내역 데이터 수집

기본정보

| | | | |
|-------|---------------------------------|------|------|
| 데이터명 | 한국부동산원_부동산 거래 현황 통계 조회 서비스 상세설명 | | |
| 서비스유형 | REST | 심의여부 | 자동승인 |
| 신청유형 | 개발계정 활용신청 | 처리상태 | 승인 |
| 활용기간 | 2023-03-15 ~ 2025-03-15 | | |

서비스정보

| | |
|-------------|--------------------------------------------------------------------|
| 데이터포맷 | JSON+XML |
| Base URL | api.odcloud.kr/api |
| Swagger URL | https://infuser.odcloud.kr/api/stages/27795/api-docs?1647332194450 |

API 환경 또는 API 호출 조건에 따라 인증키가 적용되는 방식이 다를 수 있습니다.
포털에서 제공되는 Encoding/Decoding 된 인증키를 적용하면서 구동되는 키를 사용하시기 바랍니다.
* 향후 포털에서 더 명확한 정보를 제공하기 위해 노력하겠습니다.

| | |
|----------------------|--|
| 일반 인증키 (Encoding) | |
| 일반 인증키 (Decoding) | |

활용신청 상세기능정보

활용 명세 Open API 명세 확인 가이드

입력
파라
미터

조회
가능
데이터

부동산 거래 통계 조회 서비스¹
[Base URL: api.odcloud.kr/api]
<https://infuser.odcloud.kr/api/stages/27795/api-docs?1647332194450>
한국부동산원(구.한국감정원)에서 제공하는 부동산 거래 통계를 조회할 수 있는 서비스로 토지거래현황, 순수토지 거래현황, 건축물 거래현황, 주택 거래현황, 아파트 거래현황, 주택매매거래현황, 아파트매매거래현황, 토지매매거래현황 데이터를 제공합니다.

인증키 입력

Schemes
HTTPS
인증키 설정

API 목록

| | | |
|-----|-------------------------------------------------------------|---------------------|
| GET | /RealEstateTradingSvc/v1/getRealEstateTradingArea | 부동산 거래 면적 조회 |
| GET | /RealEstateTradingSvc/v1/getRealEstateTradingAreaResidence | 매입자거주지별 부동산 거래면적 조회 |
| GET | /RealEstateTradingSvc/v1/getRealEstateTradingCount | 부동산 거래 건수 조회 |
| GET | /RealEstateTradingSvc/v1/getRealEstateTradingCountResidence | 매입자거주지별 부동산 거래건수 조회 |
| GET | /RealEstateTradingSvc/v1/getRealEstateTradingAreaBuildType | 건물유형별 부동산 거래 면적 조회 |
| GET | /RealEstateTradingSvc/v1/getRealEstateTradingCountDealer | 거래주체별 부동산 거래건수 조회 |

(Source: <https://www.data.go.kr>)

API를 활용한 데이터 수집

- 예제: 데이터 포털에서 api를 활용하여 부동산 거래내역 데이터 수집
 - 부동산 거래 건수 조회 데이터를 얻어봅시다 (예: 2021년 1월 성북구 아파트 거래)
 - (1) 먼저 웹페이지에서 데이터 바로 얻기

| Name | Description |
|------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| page Integer(Integer) (query) | page index 1 |
| perPage Integer(Integer) (query) | page size 10 |
| returnType string (query) | 응답의 데이터 타입을 선택할 수 있습니다. (기본값: JSON) XML 형태의 응답결과를 얻기 위해서는 XML 값으로 설정 returnType - 응답의 데이터 타입을 선택할 수 있음 |
| cond[RESEARCH_DATE:LT] string(String) (query) | 조사일자(YYYYMM) 202102 |
| cond[RESEARCH_DATE:LTE] string(String) (query) | 조사일자(YYYYMM) cond[RESEARCH_DATE:LTE] - 조사일자(YYYYMM) |
| cond[RESEARCH_DATE:GT] string(String) (query) | 조사일자(YYYYMM) cond[RESEARCH_DATE:GT] - 조사일자(YYYYMM) |
| cond[RESEARCH_DATE:GTE] string(String) (query) | 조사일자(YYYYMM) 202101 |
| cond[REGION_CD:EQ] string(String) (query) | 지역코드 11290 |
| cond[DEAL_OBJ:EQ] string(String) (query) | 거래유형(01:보지, 02:순수보지, 03:건축물, 04:주택, 05:아파트, 06:주택 매매, 07:아파트매매, 08:보지매매) 05 |

[참고]
<https://jamdol.tistory.com/91>

Request URL

`https://api.odcloud.kr/api/RealEstateTradingSvc/v1/getRealEstateTradingCount?page=1&perPage=10&cond%5BRESEARCH_DATE%3AX3AL%5D=202102&cond%5BRESEARCH_DATE%3AX3AGITE%5D=202101&cond%5BREGION_CD%3AX3AEQ%5D=11290&cond%5BDEAL_OBJ%3AX3AEQ%5D=05`

Server response

| Code | Details |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 200 | Response body Download <pre>{ "currentCount": 1, "data": [{ "ALL_CNT": 609, "DEAL_OBJ": "05", "LEVEL_NO": "1", "REGION_CD": "11290", "REGION_NM": "성북구", "RESEARCH_DATE": "202101" }], "matchCount": 1, "page": 1, "perPage": 10, "totalCount": 411531 }</pre> Response headers content-length: 201 content-type: application/json; charset=UTF-8 |

(Source: <https://www.data.go.kr>)

API를 활용한 데이터 수집

- 예제: 데이터 포털에서 api를 활용하여 부동산 거래내역 데이터 수집
 - 부동산 거래 건수 조회 데이터를 얻어봅시다 (예: 2021년 1월 성북구 아파트 거래)
 - (2) python으로 얻기

Request URL

```
https://api.odcloud.kr/api/RealEstateTradingSvc/v1/getRealEstateTradingCount?
page=1&perPage=10&condX5BRESEARCH_DATEX3AX3ALTIX5D=202102&condX5BRESEARCH_DATEX3AX3AGTEX5D=202101&condX5BRESEARCH_DATEX3AX3AEQX
```

Server response

Code

200

Details

Response body

Download

```
{
  "currentCount": 1,
  "data": [
    {
      "ALL_CNT": 609,
      "DEAL_OBJ": "05",
      "LEVEL_NO": "1",
      "REGION_CD": "11290",
      "REGION_NM": "성북구",
      "RESEARCH_DATE": "202101"
    }
  ],
  "matchCount": 1,
  "page": 1,
  "perPage": 10,
  "totalCount": 411531
}
```

Response headers

```
content-length: 201
content-type: application/json; charset=UTF-8
```

이 URL으로 접속해보면 실제로 데이터를 볼 수 있습니다.

Gmail YouTube 지도 번역

```
{ "currentCount": 1, "data": [ { "ALL_CNT": 609, "DEAL_OBJ": "05", "LEVEL_NO": "1", "REGION_CD": "11290", "REGION_NM": "성북구", "RESEARCH_DATE": "202101" } ], "matchCount": 1, "page": 1, "perPage": 10, "totalCount": 411531 }
```

DevTools is now available in Korean! Always match Chrome's language Switch DevTools to Korean Don't show again

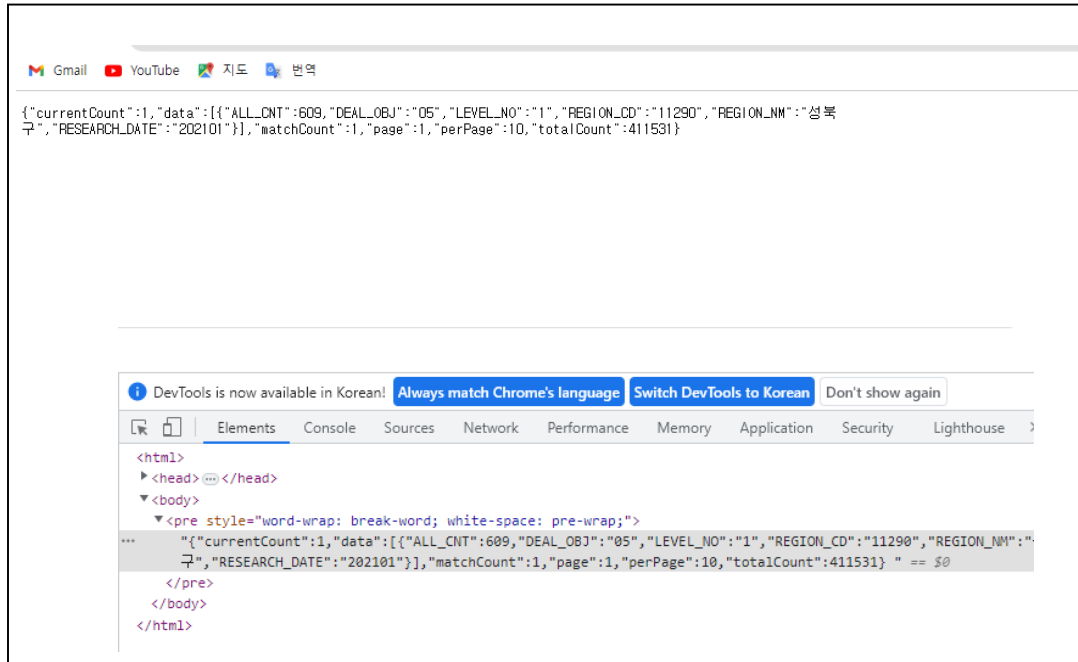
Elements Console Sources Network Performance Memory Application Security Lighthouse

```
<html>
<head>
<body>
  <pre style="word-wrap: break-word; white-space: pre-wrap;">
    { "currentCount": 1, "data": [ { "ALL_CNT": 609, "DEAL_OBJ": "05", "LEVEL_NO": "1", "REGION_CD": "11290", "REGION_NM": "성북구", "RESEARCH_DATE": "202101" } ], "matchCount": 1, "page": 1, "perPage": 10, "totalCount": 411531 }
  </pre>
</body>
</html>
```

(Source: <https://www.data.go.kr>)

API를 활용한 데이터 수집

- 예제: 데이터 포털에서 api를 활용하여 부동산 거래내역 데이터 수집
 - 부동산 거래 건수 조회 데이터를 얻어봅시다 (예: 2021년 1월 성북구 아파트 거래)
 - (2) python으로 얻기
 - 이런 '정적' 홈페이지를 scraping하는 것은 어렵지 않죠



```
from bs4 import BeautifulSoup
from urllib.request import urlopen

my_url = 'https://api.odcloud.kr/api/RealEstateTradingSvc/v1/getRealEstateTradingCo

html = urlopen(my_url)
bs_obj = BeautifulSoup(html, 'html.parser')

bs_obj

{"currentCount": 1, "data": [ { "ALL_CNT": 609, "DEAL_OBJ": "05", "LEVEL_NO": "1", "REGION_CD": "11290", "REGION_NM": "성북구", "RESEARCH_DATE": "202101" } ], "matchCount": 1, "page": 1, "perPage": 10, "totalCount": 411531 }
```

(Source: <https://www.data.go.kr>)

API를 활용한 데이터 수집

- 예제: 데이터 포털에서 api를 활용하여 부동산 거래내역 데이터 수집

- 부동산 거래 건수 조회 데이터를 얻어봅시다 (예: 2021년 1월 성북구 아파트 거래)

- (2) python으로 얻기

- 문제는 '어떻게 이 URL을 어떻게 자동으로 얻을 수 있는가?'
 - 이 웹페이지에 접속해서 다른 조건을 입력하면서 하나하나 알아볼 수는 없죠.
 - 그런데, 자세히 보면 이 URL에 패턴이 존재

```
https://api.odcloud.kr/api/RealEstateTradingSvc/v1/getRealEstateTradingCount?
page=1&perPage=10&returnType=JSON&
cond%5BRESEARCH_DATE%3A%3ALT%5D=202102&
cond%5BRESEARCH_DATE%3A%3AGTE%5D=202101&
cond%5BREGION_CD%3A%3AEQ%5D=11290&
cond%5BDEAL_OBJ%3A%3AEQ%5D=05&
serviceKey=~~~
```

(Source: <https://www.data.go.kr>)

API를 활용한 데이터 수집

- 예제: 데이터 포털에서 api를 활용하여 부동산 거래내역 데이터 수집
 - 부동산 거래 건수 조회 데이터를 얻어봅시다 (예: 2021년 1월 성북구 아파트 거래)
 - (2) python으로 얻기
 - URL주소 생성은 'OPEN API 명세 확인 가이드'를 참조해봅시다.

```
https://api.odcloud.kr/api/RealEstateTradingSvc/v1/getRealEstateTradingCount?
page=1&perPage=10&returnType=JSON&
cond%5BRESEARCH_DATE%3A%3ALT%5D=202102&
cond%5BRESEARCH_DATE%3A%3AGTE%5D=202101&
cond%5BREGION_CD%3A%3AEQ%5D=11290&
cond%5BDEAL_OBJ%3A%3AEQ%5D=05&
serviceKey=~~~
```

01. API 호출 URL 주소 생성

The screenshot shows the 'API 호출 URL 주소 생성' (Generate API Call URL) page. It includes a text box for the Base URL, a dropdown for the protocol (HTTPS), a list of API endpoints, and a text box for the endpoint. Numbered callouts 1 through 4 guide the user through the process: 1. Confirm the Base URL, 2. Confirm the protocol, 3. Confirm the endpoint, and 4. Confirm the API call URL. A summary table at the bottom shows the components of the final URL.

| 호출 URL 주소 | 통신 프로토콜 | 호출 주소 | 통신 프로토콜 메소드 |
|-------------------------------------------------------|---------|-----------------------------|-------------|
| https://api.odcloud.kr/api/15077093/v1/file-data-list | HTTPS | /15077093/v1/file-data-list | GET |

(Source: <https://www.data.go.kr>)

API를 활용한 데이터 수집

- 예제: 데이터 포털에서 api를 활용하여 부동산 거래내역 데이터 수집

- 부동산 거래 건수 조회 데이터를 얻어봅시다 (예: 2021년 1월 성북구 아파트 거래)

- (2) python으로 얻기

- 문제는 '어떻게 이 URL을 어떻게 자동으로 얻을 수 있는가?'
 - 이 웹페이지에 접속해서 다른 조건을 입력하면서 하나하나 알아볼 수는 없죠.
 - 그런데, 자세히 보면 이 URL에 패턴이 존재

```
https://api.odcloud.kr/api/RealEstateTradingSvc/v1/getRealEstateTradingCount?  
page=1&perPage=10&returnType=JSON&  
cond%5BRESEARCH_DATE%3A%3ALT%5D=202102&  
cond%5BRESEARCH_DATE%3A%3AGTE%5D=202101&  
cond%5BREGION_CD%3A%3AEQ%5D=11290&  
cond%5BDEAL_OBJ%3A%3AEQ%5D=05&  
serviceKey=~~~
```

The screenshot shows an API interface with the following parameters and options:

- page**: integer(\$int64) (query). Value: 1.
- perPage**: integer(\$int64) (query). Value: 10.
- returnType**: string (query). Value: JSON.

Below the parameters, there is a note: "응답의 데이터 타입을 선택할 수 있습니다. (기본값: JSON) XML형태의 응답결과를 얻기 위해서는 XML 값으로 설정".

(Source: <https://www.data.go.kr>)

API를 활용한 데이터 수집

- 예제: 데이터 포털에서 api를 활용하여 부동산 거래내역 데이터 수집

- 부동산 거래 건수 조회 데이터를 얻어봅시다 (예: 2021년 1월 성북구 아파트 거래)

- (2) python으로 얻기

- 문제는 '어떻게 이 URL을 어떻게 자동으로 얻을 수 있는가?'
 - 이 웹페이지에 접속해서 다른 조건을 입력하면서 하나하나 알아볼 수는 없죠.
 - 그런데, 자세히 보면 이 URL에 패턴이 존재

```
https://api.odcloud.kr/api/RealEstateTradingSvc/v1/getRealEstateTradingCount?
page=1&perPage=10&returnType=JSON&
cond%5BRESEARCH_DATE%3A%3ALT%5D=202102&
cond%5BRESEARCH_DATE%3A%3AGTE%5D=202101&
cond%5BREGION_CD%3A%3AEQ%5D=11290&
cond%5BDEAL_OBJ%3A%3AEQ%5D=05&
serviceKey=~~~
```

```
cond[RESEARCH_DATE::LT]
string($string)
(query)
```

```
cond[RESEARCH_DATE::LTE]
string($string)
(query)
```

```
cond[RESEARCH_DATE::GT]
string($string)
(query)
```

```
cond[RESEARCH_DATE::GTE]
string($string)
(query)
```

조사일자(YYYYMM)

202102

조사일자(YYYYMM)

cond[RESEARCH_DATE::LTE] - 조사일자(YYYYMM)

조사일자(YYYYMM)

cond[RESEARCH_DATE::GT] - 조사일자(YYYYMM)

조사일자(YYYYMM)

202101

(Source: <https://www.data.go.kr>)

API를 활용한 데이터 수집

- 예제: 데이터 포털에서 api를 활용하여 부동산 거래내역 데이터 수집

- 부동산 거래 건수 조회 데이터를 얻어봅시다 (예: 2021년 1월 성북구 아파트 거래)

- (2) python으로 얻기

- 문제는 '어떻게 이 URL을 어떻게 자동으로 얻을 수 있는가?'
 - 이 웹페이지에 접속해서 다른 조건을 입력하면서 하나하나 알아볼 수는 없죠.
 - 그런데, 자세히 보면 이 URL에 패턴이 존재

```
https://api.odcloud.kr/api/RealEstateTradingSvc/v1/getRealEstateTradingCount?
page=1&perPage=10&returnType=JSON&
cond%5BRESEARCH_DATE%3A%3ALT%5D=202102&
cond%5BRESEARCH_DATE%3A%3AGTE%5D=202101&
cond%5BREGION_CD%3A%3AEQ%5D=11290&
cond%5BDEAL_OBJ%3A%3AEQ%5D=05&
serviceKey=~~~
```

```
cond[REGION_CD::EQ]
string($string)
{query}
```

```
cond[DEAL_OBJ::EQ]
string($string)
{query}
```

지역코드

11290

거래유형(01:토지, 02:순수토지, 03:건축물, 04:주택, 05:아파트, 06:주택
매매, 07:아파트매매, 08:토지매매)

05

(Source: <https://www.data.go.kr>)

API를 활용한 데이터 수집

- 예제: 데이터 포털에서 api를 활용하여 부동산 거래내역 데이터 수집
 - 부동산 거래 건수 조회 데이터를 얻어봅시다 (예: 2021년 1월 성북구 아파트 거래)
 - (2) python으로 얻기
 - (Left): 검색 조건에 맞게 URL을 생성하도록 코드 작성
 - (Right upper): HTML문서를 scraping
 - (Right lower): ALL_CNT만 추출하기 위한 코드

```
endpoint = 'https://api.odcloud.kr/api/RealEstateTradingSvc/v1/getRealEstateTradingCount'  
service_key =  
page = 1  
perpage = 10  
start_month = '202101'  
end_month = '202102'  
region = '11290'  
trading_type='05'
```

```
cond1 = f'cond%5BRESEARCH_DATE%3A%3ALT%5D={end_month}'  
cond2 = f'cond%5BRESEARCH_DATE%3A%3AGTE%5D={start_month}&  
cond3 = f'cond%5BREGION_CD%3A%3AEQ%5D={region}'  
cond4 = f'cond%5BDEAL_OBJ%3A%3AEQ%5D={trading_type}'  
  
url = f'{endpoint}?page={page}&perPage={perpage}&{cond1}&{cond2}&{cond3}&{cond4}&serviceKey={service_key}'
```

```
html = urlopen(url)  
bs_obj = BeautifulSoup(html, 'html.parser')
```

bs_obj

```
{ "currentCount": 1, "data": [ { "ALL_CNT": 609, "DEAL_OBJ": "05", "LEVEL_NO": "1", "REGION_CD": "11290", "REGION_NM": "성북구", "RESEARCH_DATE": "202101" } ], "matchCount": 1, "page": 1, "perPage": 10, "totalCount": 411531 }
```

```
result = eval(bs_obj.text)  
print(result['data'][0]['ALL_CNT'])
```

609

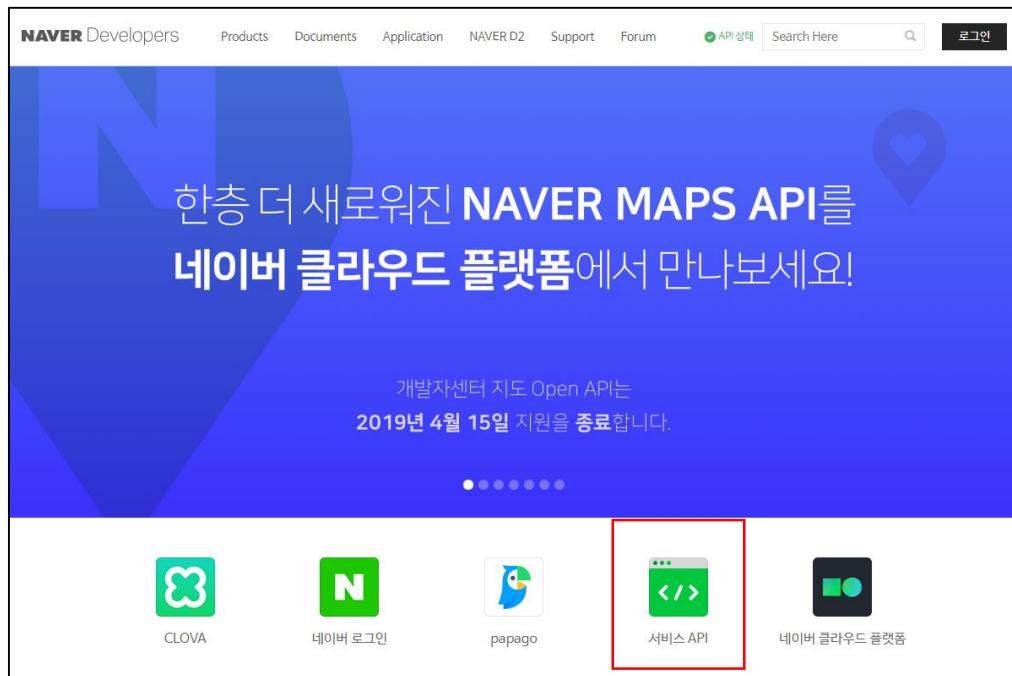
(Source: <https://www.data.go.kr>)

3. (실습) 네이버 뉴스 scraping

API를 활용한 데이터 수집

• 예제 (1) Naver API를 활용한 scraping

– <https://developers.naver.com> → 로그인 → 서비스 API → 검색 → API신청



애플리케이션 등록 (API 이용신청)

애플리케이션의 기본 정보를 등록하면, 좌측 **내 애플리케이션** 메뉴의 서브 메뉴가 표시됩니다.

(4) 신청하기를 누르면 같은 화면이 뜬. 당황하지 말고, 내 애플리케이션 메뉴로 이동

| | | |
|--------------------|----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| 애플리케이션 이름 | <input type="text" value="jaylee"/> | (1) 애플리케이션 이름 작성 |
| 사용 API | <div>선택하세요. <input type="text" value="검색"/></div> | (2) 사용 API선택 ('검색'은 꼭 포함) |
| 비로그인 오픈 API 서비스 환경 | <div>환경 추가</div> <div>WEB 설정</div> <div>웹 서비스 URL (최대 10개)</div> <div><input type="text" value="https://localhost"/></div> | (3) WEB 설정 환경 추가 (별도 URL이 없으면 https://localhost) |

• 데이터 로그인할 때 사용자에게 표시되는 이름이므로 서비스 브랜드를 대표할 수 있는 이름으로 가급적 10자 이내로 간결하게 설정해주세요.
• 40자 이내의 영문, 한글, 숫자, 공백문자, 점표(.)

• 텍스트 폼 우측 끝의 '+' 버튼을 누르면 행이 추가되며, '-' 버튼을 누르면 행이 삭제됩니다.
• http와 https는 구분하지 않습니다.
• www는 빼고 입력해 주세요. 예) http://naver.com
• 서브 도메인이 있으면 대표 도메인명만 입력해 주세요. (예: http://naver.com)
• 하이브라드 앱은 location.href 객체 출력 값을 입력하면 됩니다. (예: file://로컬 URL)

(Source: <https://developers.naver.com/main/>)

API를 활용한 데이터 수집

- 예제 (1) Naver API를 활용한 scraping

- <https://developers.naver.com> → 로그인 → 서비스 API → 검색 → API신청

jaylee

개요 API 설정 멤버관리 로그인 통계 API 통계 Playground(Beta)

애플리케이션 정보 ID, Secret을 별도로 저장해주세요(메모장 등)

Client ID

Client Secret

API 호출 안내

지도 API 인증실패나 네이버 로그인 이용 제한이 걸렸다면 [API 설정] 탭에서 URL 관련 설정을 수정하시면 정상 이용 가능합니다 !!!

비로그인 오픈 API 당일 사용량 API호출량/일일허용량

검색 0/25000

데이터랩 (검색어트렌드) 0/1000

(Source: <https://developers.naver.com/main/>)

API를 활용한 데이터 수집

- 예제 (1) Naver API를 활용한 scraping
 - Documentation을 봅시다.



Documents

네이버 오픈 API를 이용해 창의적인 애플리케이션을 제작해 보세요.

Documents > 서비스 API > 검색

블로그

뉴스

책

성인 검색어 판별

백과사전

영화

카툰

지식iN

지역

오타변환

웹문서

이미지

쇼핑

전문자료

검색) 뉴스

- 뉴스 검색 개요
 - 개요
 - 사전 준비 사항
- 뉴스 검색 API 레퍼런스
 - 뉴스 검색 결과 조회
 - 오류 코드
- 검색 API 뉴스 검색 구현 예제

뉴스 검색 개요

- 개요
- 사전 준비 사항

개요

검색 API와 뉴스 검색 개요

검색 API는 네이버 검색 결과를 뉴스, 백과사전, 블로그, 쇼핑, 영화, 웹 문서, 전문정보, 지식iN, 책, 카툰 등 분야별로 볼 수 있는 API입니다. 그 외에 지역 검색 결과와 성인 검색어 판별 기능, 오타 변환 기능을 제공합니다.

뉴스 검색은 검색 API를 사용해 네이버 검색의 뉴스 검색 결과를 반환하는 RESTful API입니다. 뉴스 검색 결과를 XML 형식 또는 JSON 형식으로 반환합니다. API를 호출할 때는 검색어와 검색 조건을 쿼리 스트링(Query String) 형식의 데이터로 전달합니다.

뉴스 검색은 검색 API를 사용하며, 검색 API의 하루 호출 한도는 25,000회입니다.

(Source: <https://developers.naver.com/main/>)

API를 활용한 데이터 수집

- 예제 (1) Naver API를 활용한 scraping

- Documentation에서 예제 코드도 볼 수 있습니다. (blog이긴 하지만 유사한 형식)

검색 API 뉴스 검색 구현 예제 [↗](#)

검색 API로 뉴스 검색 결과를 조회하는 방법은 블로그 검색 결과를 조회하는 방법과 유사합니다. 뉴스 검색 결과 조회를 구현하는 방법은 [검색 API 블로그 검색 구현 예제](#)를 참고합니다.



Python [↗](#)

```
# 네이버 검색 API 예제 - 블로그 검색
import os
import sys
import urllib.request
client_id = "YOUR_CLIENT_ID"
client_secret = "YOUR_CLIENT_SECRET"
encText = urllib.parse.quote("검색할 단어")
url = "https://openapi.naver.com/v1/search/blog?query=" + encText # JSON 결과
# url = "https://openapi.naver.com/v1/search/blog.xml?query=" + encText # XML 결과
request = urllib.request.Request(url)
request.add_header("X-Naver-Client-Id", client_id)
request.add_header("X-Naver-Client-Secret", client_secret)
response = urllib.request.urlopen(request)
rescode = response.getcode()
if(rescode==200):
    response_body = response.read()
    print(response_body.decode('utf-8'))
else:
    print("Error Code:" + rescode)
```

(Source: <https://developers.naver.com/main/>)

API를 활용한 데이터 수집

- 예제 (2) BeautifulSoup을 활용한 scraping

- NAVER API를 사용할 수 있는 횟수는 제한 되어 있습니다.

API 호출 안내

지도 API 인증실패나 네이버 로그인 이용 제한이 걸렸다면 [API 설정] 탭에서 URL 관련 설정을 수정하시면 정상 이용 가능합니다 !!!

비로그인 오픈 API 당일 사용량

API호출량/일일허용량



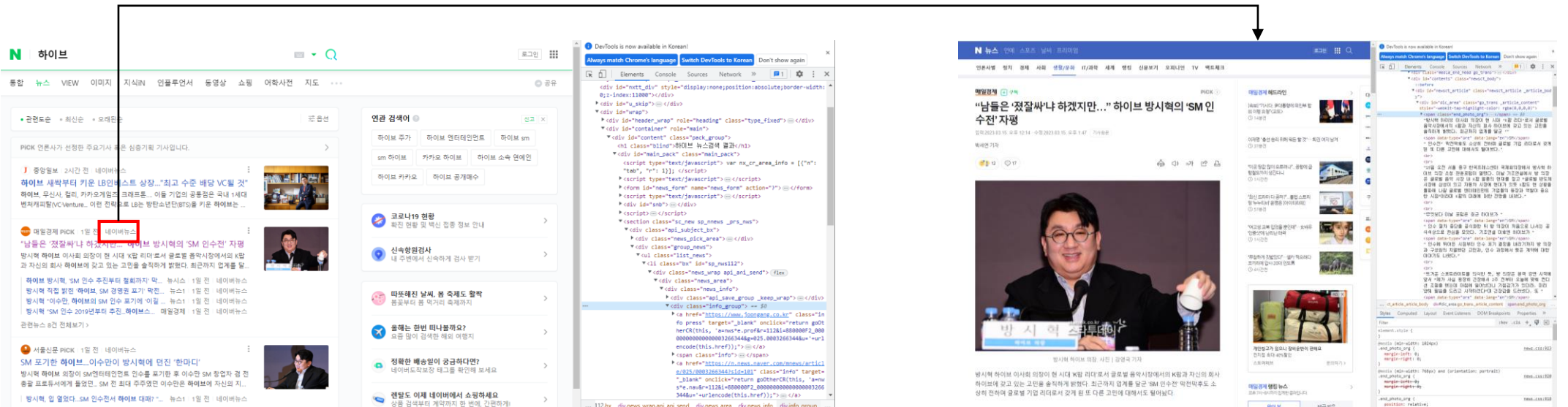
- BeautifulSoup을 활용한 방식을 알아봅시다.
 - 그러나, 공식적인 방법인 API를 사용하는 것이 안전하다는 것을 알려드립니다.

(Source: <https://developers.naver.com/main/>)

API를 활용한 데이터 수집

• 예제 (2) BeautifulSoup을 활용한 scraping

- Step 1. 검색 결과창에 있는 '네이버 뉴스' 링크를 수집
- Step 2. 해당 링크에 접속하여 제목과 본문 수집



(Source: <https://www.naver.com>)

API를 활용한 데이터 수집

• 예제 (2) BeautifulSoup을 활용한 scraping

– Step 1. 검색 결과창에 있는 '네이버 뉴스' 링크를 수집

The screenshot shows the Naver search results for '하이브' (HYBE). The main content area displays several news items. The first item is titled '하이브 새싹부터 키운 LB인베스트 성장...' and includes a snippet about HYBE's investment in K-pop artists. The DevTools window is open, showing the HTML structure of the selected news item. The relevant HTML elements are highlighted, showing the path to the news link: `#sp_nws113 > div.news_wrap.api_ani_send > div > div.news_info > div.info_group > a:nth-child(3)`.

마우스 우클릭 > Copy > Copy Selector 선택

#sp_nws113 >
div.news_wrap.api_ani_send
> div > div.news_info >
div.info_group > a:nth-child(3)

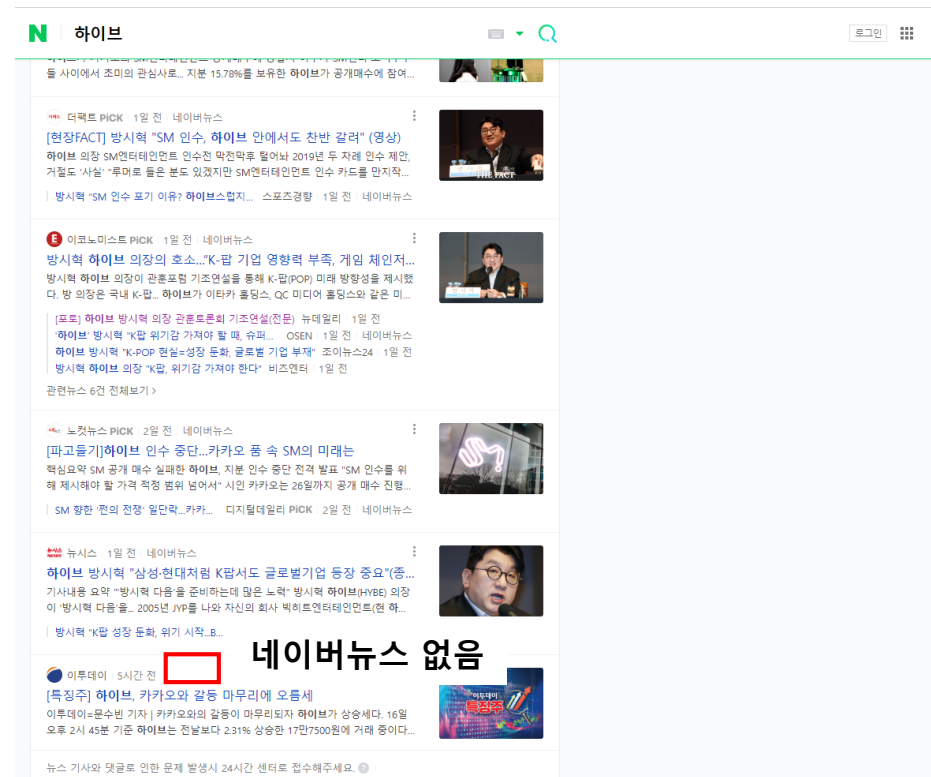
(Source: <https://www.naver.com>)

API를 활용한 데이터 수집

• 예제 (2) BeautifulSoup을 활용한 scraping

– Step 1. 검색 결과창에 있는 '네이버 뉴스' 링크를 수집

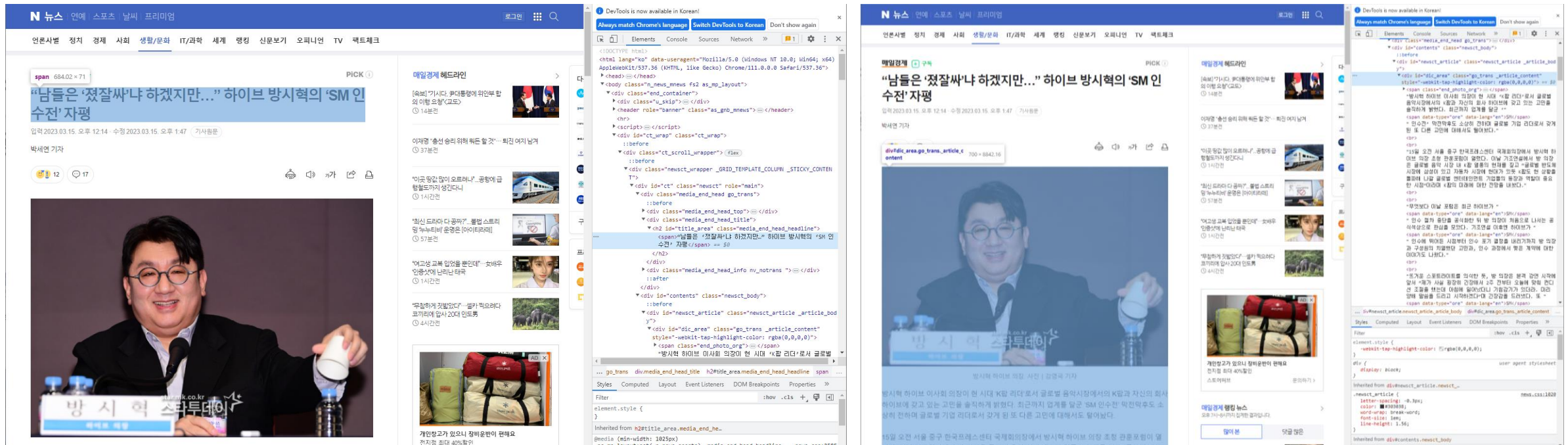
- 가끔 네이버뉴스가 없는 경우도 있다. → 이런 결과는 제외!



(Source: <https://www.naver.com>)

API를 활용한 데이터 수집

• 예제 (2) BeautifulSoup을 활용한 scraping – Step 2. '네이버 뉴스' 링크에서 제목과 본문 수집



마우스 우클릭 > Copy > Copy Selector 선택

#title_area > span

마우스 우클릭 > Copy > Copy Selector 선택

#newsct_article

(Source: <https://developers.naver.com/main/>)

4. Web scraping시 주의사항

(중요) Web Scraping시 주의사항

• 권한 확인(robots.txt)

– API를 사용하지 않으면 반드시 권한을 확인하세요.

- 일반적으로 크롤링하려는 사이트 뒤에 robots.txt를 붙여서 검색하면 볼 수 있음

– 예시) <https://www.naver.com/robots.txt>

그리고 robots.txt 파일의 형식은 아래와 같습니다.

[robots.txt 파일 형식]

User-agent: <= 검색봇 이름

Disallow: <= 접근 설정

Crawl-delay: 다음방문까지의 디레이(초)

[모든 검색봇에 차단]

User-agent: *

Disallow: /

[모든 검색봇에 허용]

User-agent: *

Disallow:

[모든 검색봇에 대해 일부만 허용]

User-agent: *

Disallow: /page1/

Disallow: /page2/

(Source: <https://nick2ya.tistory.com/11>)

(중요) Web Scraping시 주의사항

• 데이터 요청 주기

- 웹 서버에 계속 데이터를 요청하는 행위는 디도스 공격과 유사한 행위
 - 짧은 시간에 수많은 요청 → 웹 서버에 부하를 일으킴
 - time.sleep()으로 적절히 쉬어가면서 데이터 요청
 - 참고) 이를 잘못 사용해서 IP도 block당한 경우가 존재

• 저작권 주의

- 데이터를 받아서 혼자서 사용하는 것은 크게 문제가 없을 수 있음
 - 문제가 있는 경우도 존재하니 반드시 확인
- 그러나 이를 상업적으로 이용하는 것은 문제
 - 반드시 확인

(중요) Web Scraping시 주의사항

- Web Scraping이 활성화 되면서 웹서버에서는 봇을 차단함
 - 꼭 필요하다면 별도의 기술이 필요할 수도
 - 참고) <https://noodle-dev.tistory.com/41>
 - 그러나, 이 경우에도 문제의 소지가 있을 수 있으니 꼭 확인할 것

(Source: <https://nick2ya.tistory.com/11>)

End of the documents
