

크롤링 차근차근 시작하기 (2/E) - 중급편


# 인스타그램에서 로그인하기

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

# Instagram

Log In


OR


 Log in with Facebook

[Forgot password?](#)

Don't have an account? [Sign up](#)

Get the app.

 Download on the App Store

 GET IT ON Google Play

## 시작하기에 앞서 ...

본 과정은 어디까지나 사람이 하는 과정을 자동화하는 것 뿐입니다.

인스타그램에서의 UI 흐름은 언제라도 변경될 수 있습니다. 본 에피소드를 통해, 같이 분석해보고 인스타그램의 인증과정에 대해서 이해해봅시다.

Selenium 로직을 살펴보기 전에, 먼저 Instagram 로그인을 직접 수없이 진행해보시면서, 그 절차를 차근차근 이해하셔야 합니다.

저는 하나의 "예"를 보여드릴 따름입니다.

<https://gist.github.com/allieus/3f0c709d869c462399e8e0eb1d491fc9>

Instagram 클래스에 기능을 덧붙여 Instagram2 클래스를 정의했습니다.  
→ 편의상 클래스를 구분했을 뿐, 실적용은 단일 클래스로 쓰세요.

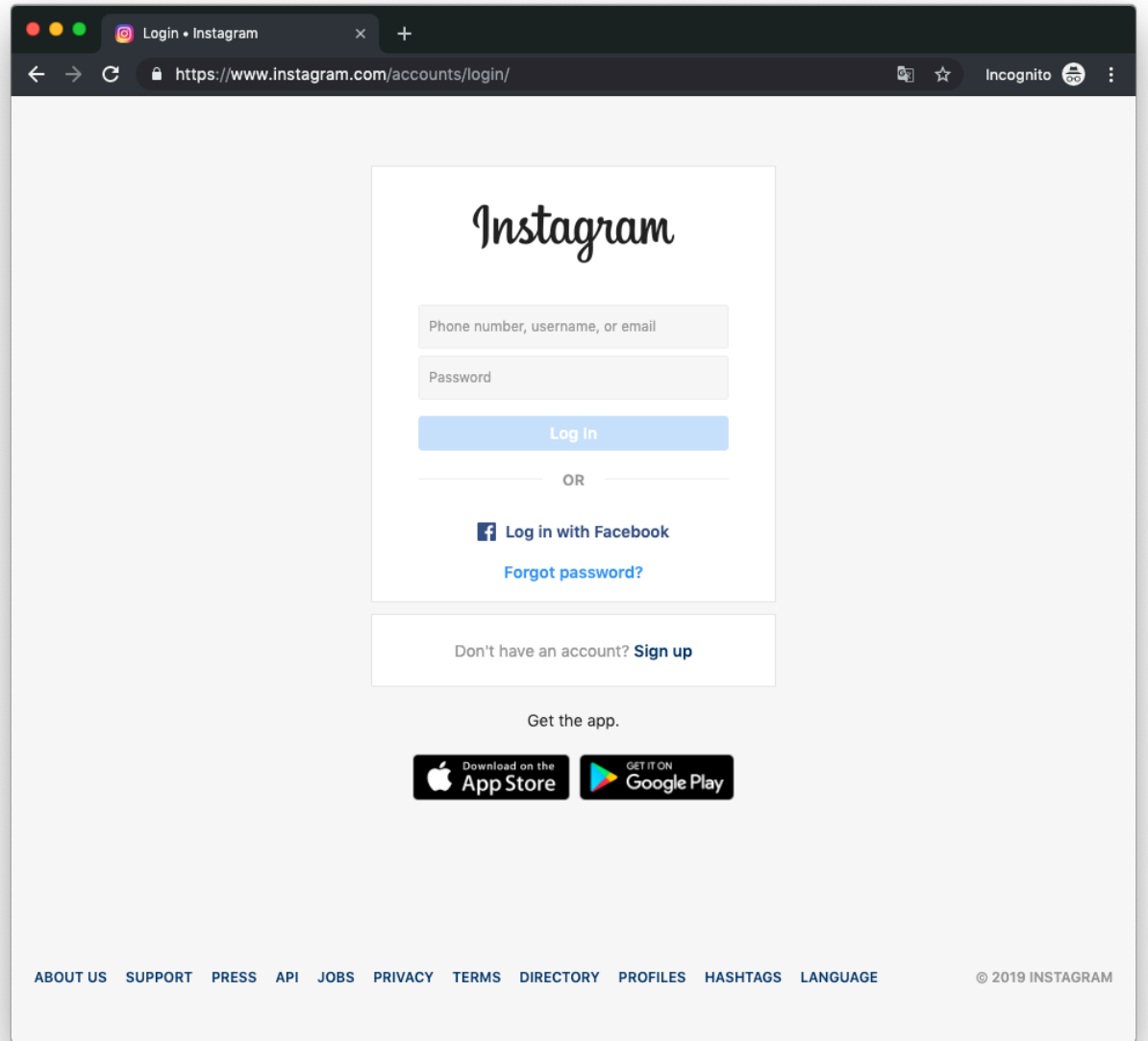
# 로그인을 하고 싶습니다.

내 타임라인 읽기

좋아요, 팔로우 등의 처리

2가지 인증 지원

- email/password 인증
  - 이메일 등을 통한 추가 인증을 요구하기도 합니다.
- 페이스북 OAuth 인증



# 필요한 패키지

selenium

requests

beautifulsoup4

imbox

chromedriver 파일

# 아이디/패스워드 인증

# 로그인 기본 로직

인스타그램이 파이썬/장고로 구현되어있다는 증거. 로그아웃은 /accounts/logout/

```
driver.get("https://www.instagram.com/accounts/login/")
```

```
instagram_username = '*****' # FIXME: 인스타그램 계정명
```

```
instagram_password = '*****' # FIXME: 인스타그램 계정암호
```

```
driver.find_element_by_css_selector('[name=username]').send_keys(instagram_username)
```

```
driver.find_element_by_css_selector('[name=password]').send_keys(instagram_password)
```

```
driver.find_element_by_css_selector('[type=submit]').click()
```

# 본 에피소드 최종 코드 - 아이디/패스워드 로그인 + 인증코드

```
driver_path = './drivers/chromedriver-mac64-74.0.3729.6'  
instagram_username = '*****'  
instagram_password = '*****'  
gmail_username = '*****'  
gmail_password = '*****'
```

```
with Instagram2(driver_path) as insta:  
    try:  
        insta.try_login(instagram_username, instagram_password, gmail_username, gmail_password)  
    except RequestSecurityCodeException:  
        insta.try_security_code()  
  
    print("is_logged :", insta.is_logged)  
    if insta.is_logged:  
        session_id = insta.get_session_id()  
        print("session_id :", session_id)
```

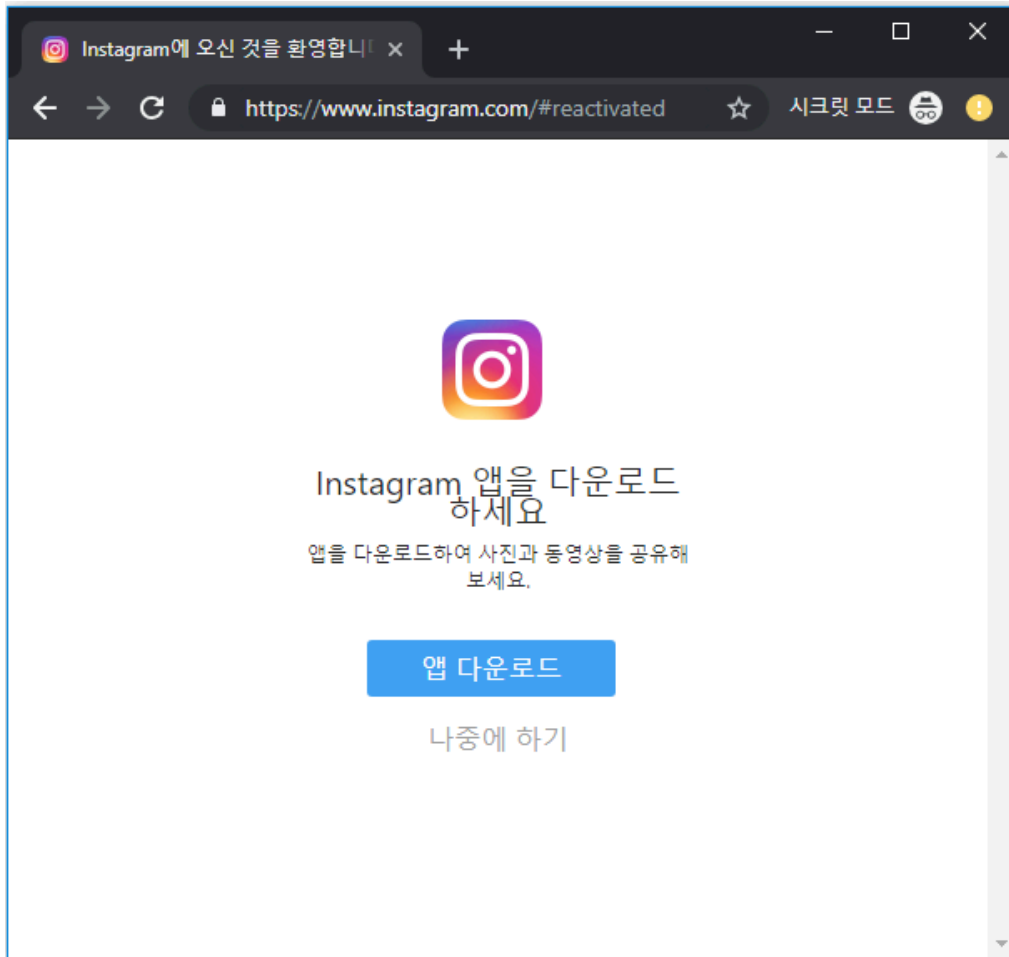
# 본 에피소드 최종 코드 - 세션 재활용

```
driver_path = './drivers/chromedriver-mac64-74.0.3729.6'  
session_id = '*****'
```

```
with Instagram2(driver_path) as insta:  
    insta.set_session_id(session_id)  
    print("is_loggd :", insta.is_logged)
```



# 인스타그램 앱 다운로드 안내가 나올 수도 있습니다.



"나중에 하기" 버튼을 클릭하거나,  
홈으로 강제 이동을 하여 해결

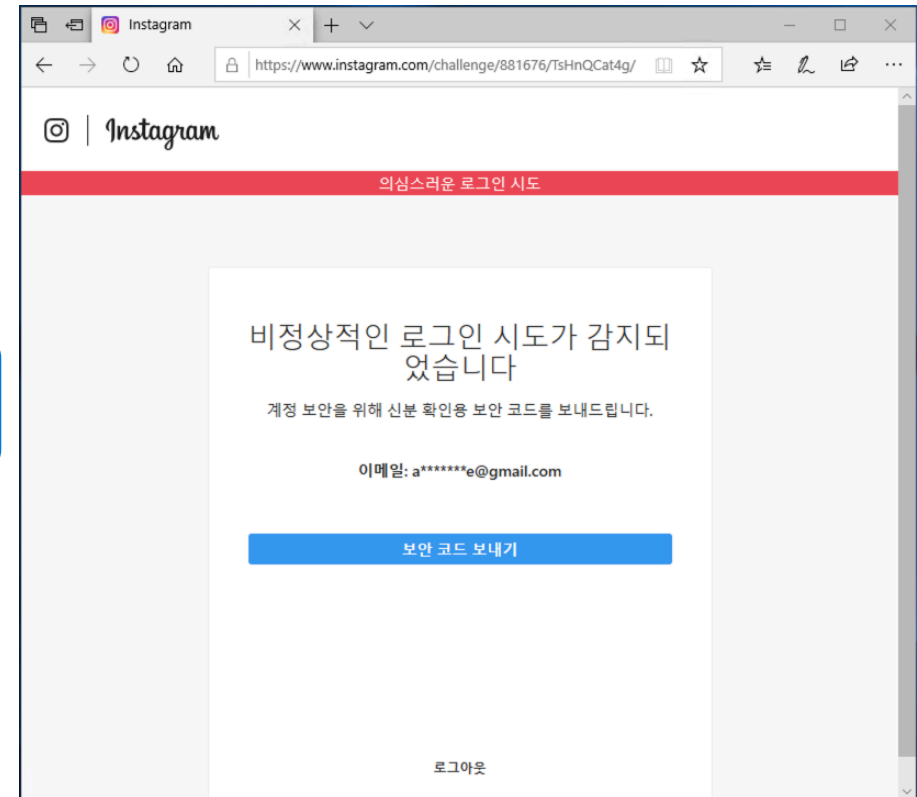
# 비정상 로그인시도가 탐지되면 → Security Code를 요구 (1/3)

이메일을 통해 Security Code를 전송 (다양한 요구 중의 하나)

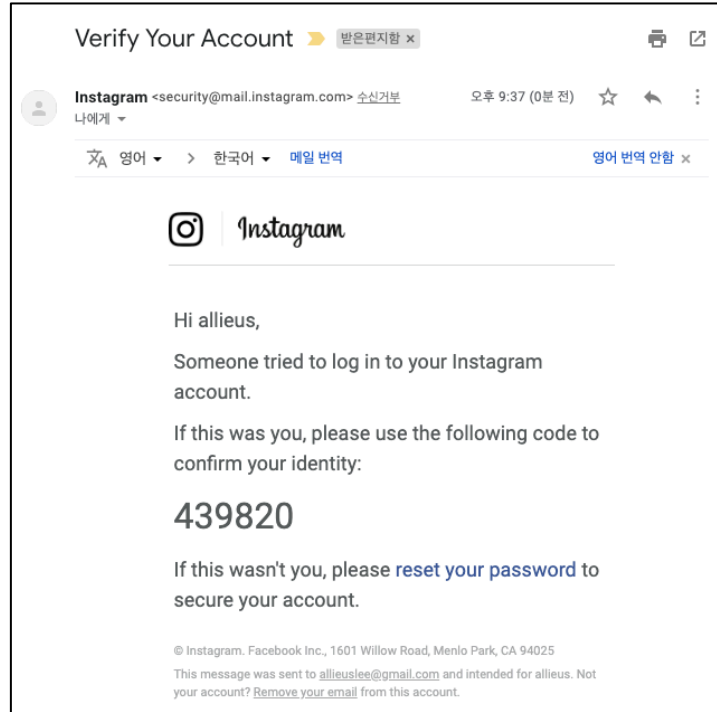
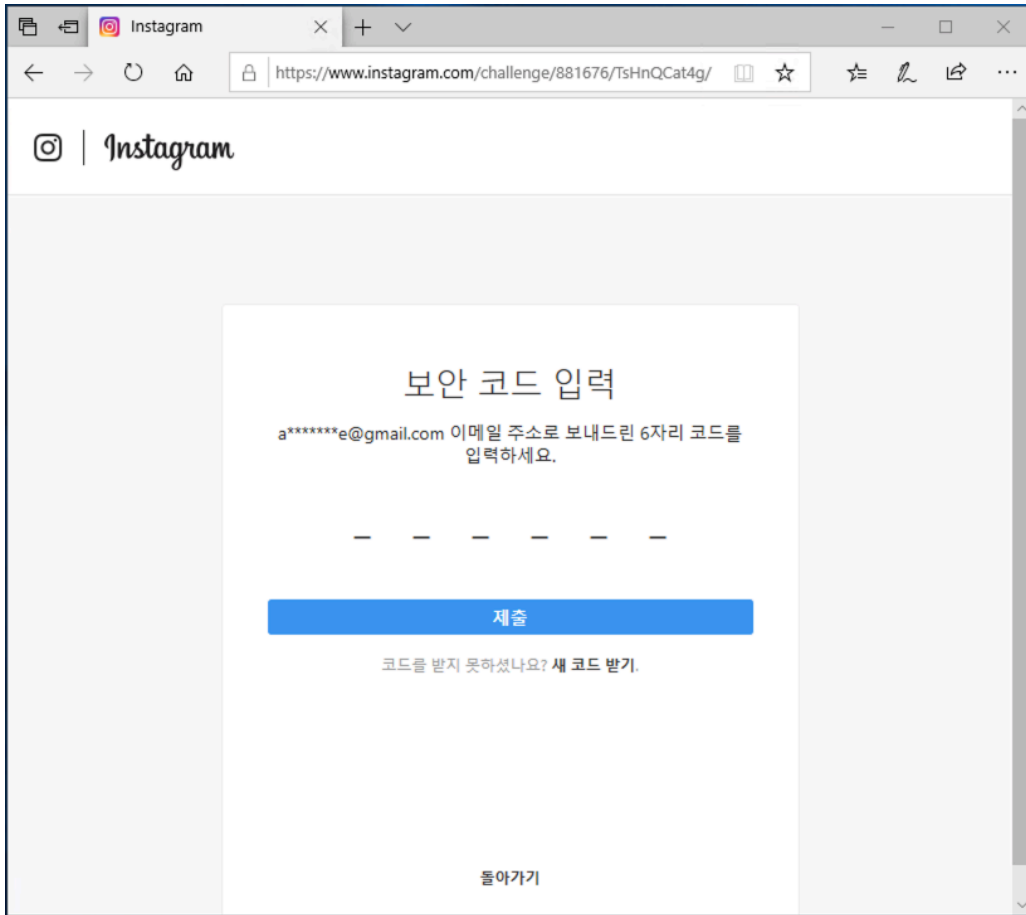
security@mail.instagram.com에서 발송한 보안코드 메일을 읽어서, 보안코드 자동 기입

→ imbox 라이브러리를 통해 IMAP 프로토콜로 이메일 읽기

홈으로 강제 이동을 하면, 재로그인 요청  
"보안 코드 보내기" 를 무조건 수행해야함



# Security Code 확인 순서



계정/위치 등의 요인에 따라  
추가 확인 과정이 더 있을 수도 있습니다.

font 107.81 x 38  
439820

If this wasn't you, please [reset your password](#) to

ty Audits

```
please use the following code to confirm your identity:</p>
<p style="margin:10px 0 10px 0;color:#565a5c;font-size:18px">
  <font size="6">439820</font> == $0
</p>
```

→ `r'^\d+(\d{6})^\d'`

# 비정상 로그인시도가 탐지되면 → Security Code를 요구 (2/3)

아래 코드는 Gmail에서 보안 코드 읽어오기

```
class Instagram2(Instagram):
    SECURITY_CODE_PATTERN = r'^\d+(\d{6})^\d{'

# 중략

def get_security_code_by_gmail(self, username, password, **timedelta_kwargs):
    with Imbox('imap.gmail.com', username=username, password=password,
               ssl=True, ssl_context=None, starttls=False) as imbox:

        time = datetime.now() - timedelta(**timedelta_kwargs)
        messages = imbox.messages(sent_from='security@mail.instagram.com',
                                   # subject='Verify Your Account',
                                   date__gt=time)
        for uid, message in messages:
            body_list = message.body['html'] or message.body['plain']
            if body_list:
                content = ''.join(body_list)
                matched = re.search(self.SECURITY_CODE_PATTERN, content)
                if matched:
                    return matched.group(1)

    return None
```

네이버 메일의 경우, imap.naver.com → imap 활성화 설정 필요

비밀번호 혹은 앱 비밀번호  
Gmail에 2단계 인증을 걸어둔 경우, 앱 비밀번호를 별도로 할당받아 지정  
<https://support.google.com/accounts/answer/185833>

# 비정상 로그인시도가 탐지되면 → Security Code를 요구 (3/3)

네이버 메일의 경우, imap 설정을 확인해주세요.

환경 설정 | 메일로 돌아가기

기본 환경 설정

메일함 관리

메일 자동 분류

서명/빠른답장

부재 중 설정

새 메일 알림 설정

스팸 설정

외부 메일 가져오기

✓ POP3/IMAP 설정

단축키

POP3/SMTP 설정

IMAP/SMTP 설정

스마트폰에서 네이버 메일을 더욱 잘 관리하고 쓸 수 있도록 IMAP/SMTP를 설정합니다.

allieuslee님은 현재 IMAP/SMTP를 사용하고 있습니다.

IMAP/SMTP 사용

☒ 사용함 ☐ 사용 안 함 ?

IMAP 동기화 메일 제한

1,000통 ↕

IMAP으로 동기화할 수 있는 최대 메일 통수를 지정할 수 있습니다.  
너무 많은 메일을 동기화하는 경우 데이터 과금을 주의해 주세요.

기본 설정으로

확인

취소

스마트폰 메일 애플리케이션 계정 설정

스마트폰의 메일 계정 설정에 아래와 같이 등록해 주세요.

IMAP 서버명 : imap.naver.com

SMTP 서버명 : smtp.naver.com

IMAP 포트 : 993, 보안연결(SSL) 필요

SMTP 포트 : 587, 보안 연결(TLS) 필요  
(TLS가 없는 경우 SSL로 연결)

아이디 : allieuslee

비밀번호 : 네이버 로그인 비밀번호

TIP

IMAP/SMTP 설정 및 해제 방법

IMAP이란 무엇인가?

# 로그인 로직

```
class Instagram2(Instagram):
    # 중략

    def try_login(self, username, password, gmail_username, gmail_password):
        self.browser.visit("https://www.instagram.com/accounts/login/")

        self.browser.webdriver_select_one('[name=username]', 1).send_keys(username)
        self.browser.webdriver_select_one('[name=password]', 1).send_keys(password)
        self.browser.webdriver_select_one('[type=submit]', 1).click()

        try:
            self.browser.webdriver_until(
                lambda driver: '/accounts/login/' not in driver.current_url,
                explicitly_wait_time=3)
        except TimeoutException:
            pass

        current_url = self.browser.driver.current_url
        # Instagram 앱 다운로드 안내 페이지
        if '#reactivated' in current_url:
            self.browser.visit('https://www.instagram.com')

        elif '/challenge/' in current_url:
            html = self.browser.driver.page_source
            if ('보안 코드 보내기' in html) or ('Send Security Code' in html):
                raise RequestSecurityCodeException

        # 다른 상황이 있을 수도 있습니다.

        if not self.is_logged:
            raise UnexpectedException('Unexpected URL : {}'.format(current_url))
        else:
            self.dismiss_notification()

    # 생략
```

# 로그인 여부 판단 로직

```
class Instagram2(Instagram):
```

```
    @property
```

```
    def is_logged(self):
```

```
        # 문자열로 획득 (비교: BeautifulSoup4의 Tag['class'] 반환값은 리스트 타입)
```

```
        class_str = insta.browser.webdriver_select_one('html').get_attribute('class')
```

```
        return 'not-logged-in' not in class_str
```

'logged-in' 혹은 'nog-logged-in'

# 보안코드 로직

```
class Instagram2(Instagram):
    def try_security_code(self):
        html = self.browser.driver.page_source

        for we in driver.find_elements_by_css_selector('button'):
            if we.text in ('보안 코드 보내기', 'Send Security Code'):
                we.click()
                print("보안 코드 전송을 요청했습니다.")
                break
        else:
            raise UnexpectedException('보안 코드 보내기 버튼을 찾을 수 없습니다.')

        for retry in range(1, 6):
            print("Gmail 계정 {}에서 보안코드 확인 : {}번째 시도".format(gmail_username, retry))
            security_code = self.get_security_code_by_gmail(gmail_username, gmail_password, minutes=5)
            if security_code:
                print("보안코드 {} 획득 성공".format(security_code))
                break
            else:
                time.sleep(5)
        else:
            raise UnexpectedException('Gmail 계정 {}에서 보안코드 확인 실패'.format(gmail_username))

        print("보안코드 기입")
        self.browser.webdriver_select_one('[name=security_code]').send_keys(security_code)

        for we in self.browser.webdriver_select('button'):
            if we.text in ('제출', 'Submit'):
                print("보안코드 제출")
                we.click()
                break
        else:
            raise UnexpectedException()
```



# 다시 최종 코드를 확인해봅시다. 😊

```
driver_path = './drivers/chromedriver-mac64-74.0.3729.6'
instagram_username = '*****'
instagram_password = '*****'
gmail_username = '*****'
gmail_password = '*****'
```

```
with Instagram2(driver_path) as insta:
    try:
        insta.try_login(instagram_username, instagram_password, gmail_username, gmail_password)
    except RequestSecurityCodeException:
        insta.try_security_code()

print("is_logged :", insta.is_logged)
if insta.is_logged:
    session_id = insta.get_session_id()
    print("session_id :", session_id)
```

다음 챕터에서 설명합니다.

```
driver_path = './drivers/chromedriver-mac64-74.0.3729.6'
session_id = '*****'
```

```
with Instagram2(driver_path) as insta:
    insta.set_session_id(session_id)
    print("is_loggd :", insta.is_logged)
```

(1번의 수동 과정과 함께 가장 손쉬운 방법) → 추천  
session id 재활용

# 로그인이 되어있을 때, sessionid를 획득해두세요.

인스타그램은 django로 구현된 서비스 → sessionid 쿠키값으로 세션 구분

```
>>> driver.get_cookie('sessionid')
```

```
{'domain': '.instagram.com',  
'expiry': 1588664778.629663,  
'httpOnly': True,  
'name': 'sessionid',  
'path': '/',  
'secure': True,  
'value': '901179%3AUHkt1k2uxtpIUH%3A28'}
```

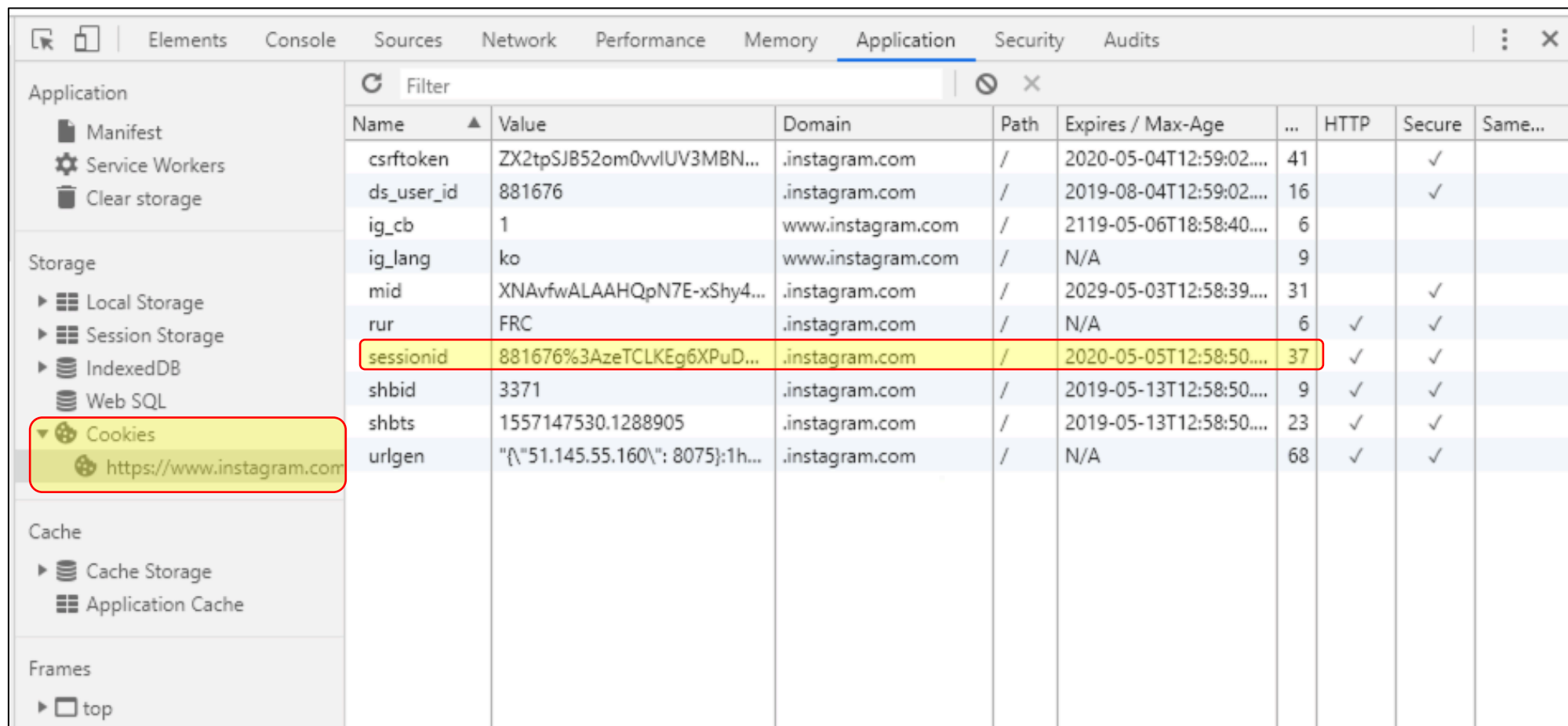
```
>>> from datetime import datetime  
>>> dt = datetime.datetime.fromtimestamp(1588664778.629663)  
>>> print(dt.strftime('%Y-%m-%d %H:%M:%S'))  
datetime.datetime(2020, 5, 5, 7, 46, 18, 629663)
```

→ 1년의 Expire Time

프론트엔드단 자바스크립트 document.cookie를 통해 조회 불가  
장고의 settings.SESSION\_COOKIE\_HTTPONLY 설정이 디폴트 True

```
from urllib.parse import unquote  
quote(value)를 수행  
→ '901179:UHkt1k2uxtpIUH:28'
```

# sessionid 쿠키는 개발자도구를 통해 확인 가능



The screenshot shows the Chrome DevTools Application tab with the 'Cookies' section expanded. The 'sessionid' cookie is highlighted with a red box. The table below represents the data shown in the 'Cookies' section.

Name	Value	Domain	Path	Expires / Max-Age	...	HTTP	Secure	Same...
csrftoken	ZX2tpSJ852om0vvIUV3MBN...	.instagram.com	/	2020-05-04T12:59:02....	41		✓	
ds_user_id	881676	.instagram.com	/	2019-08-04T12:59:02....	16		✓	
ig_cb	1	www.instagram.com	/	2119-05-06T18:58:40....	6			
ig_lang	ko	www.instagram.com	/	N/A	9			
mid	XNAvfWALAAHQpN7E-xShy4...	.instagram.com	/	2029-05-03T12:58:39....	31		✓	
rur	FRC	.instagram.com	/	N/A	6	✓	✓	
sessionid	881676%3AzeTCLKEg6XPuD...	.instagram.com	/	2020-05-05T12:58:50....	37	✓	✓	
shbid	3371	.instagram.com	/	2019-05-13T12:58:50....	9	✓	✓	
shbts	1557147530.1288905	.instagram.com	/	2019-05-13T12:58:50....	23	✓	✓	
urlgen	"{"51.145.55.160": 8075}:1h...	.instagram.com	/	N/A	68	✓	✓	

# 획득한 session id를 통해, 세션 재활용

새로운 WebDriver세션에서 이미 획득한 session id를 지정

```
class Instagram2(Instagram):
```

```
    # 중략
```

```
    def get_session_id(self): 새로운 WebDriver 세션에서 sessionid 쿠키 지정을 통해, 세션 재활용
```

```
        try:
```

```
            session_id = unquote(self.browser.driver.get_cookie('sessionid')['value'])
```

```
        except TypeError:
```

```
            session_id = None # 로그아웃 상황으로 볼 수 있습니다.
```

```
        return session_id
```

```
    def set_session_id(self, session_id):
```

```
        self.browser.visit("https://www.instagram.com")
```

```
        self.browser.driver.add_cookie({
```

```
            "name": "sessionid",
```

```
            "value": session_id,
```

```
        })
```

```
        self.browser.driver.refresh()
```

```
    # 생략
```

해당 세션이 만료시간을 초과하여 만료되거나,  
직접 로그아웃을 하여 만료시키지 전까지는 유효

인생은 짧습니다.  
파이썬/장고를 쓰세요.

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

- Ask Company