

Naver Café Crawling

CLab.

Byeongjo Park



*Computer
communication Lab.*



Index

- ◆ Setting
- ◆ Program
- ◆ Result
- ◆ Reference





Setting(1/16)

◆ PyCharm 2019.3.3 x64

◆ Python 3.8

◆ Chromedriver 80.0.3987.132

- ❖ Chrome > Chrome://settings/help > version check
- ❖ <https://chromedriver.chromium.org/downloads> > now version download

◆ Install Packages

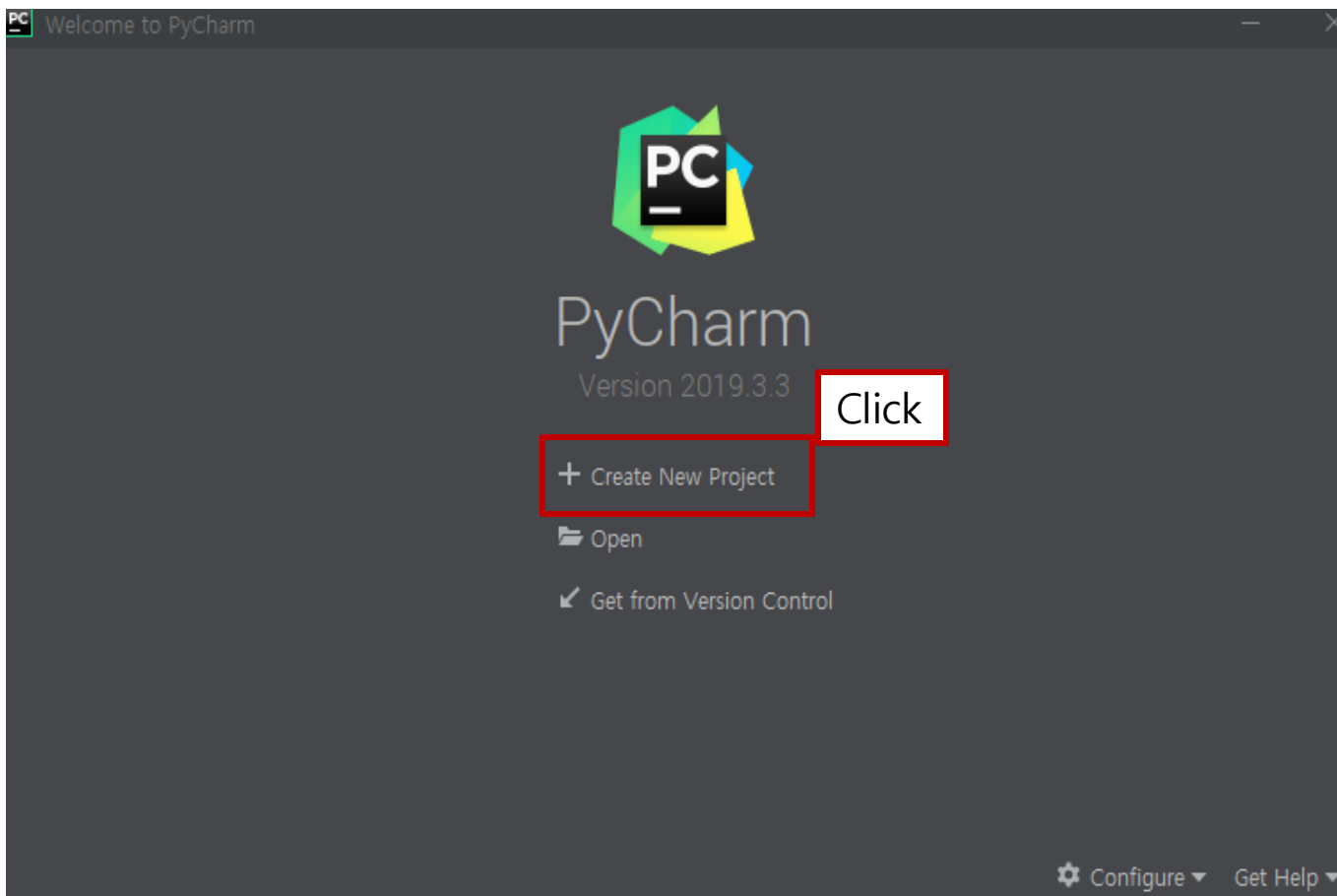
- ❖ Selenium 3.141
- ❖ Django 3.0.4
- ❖ Bs4 0.0.1





Setting(2/16)

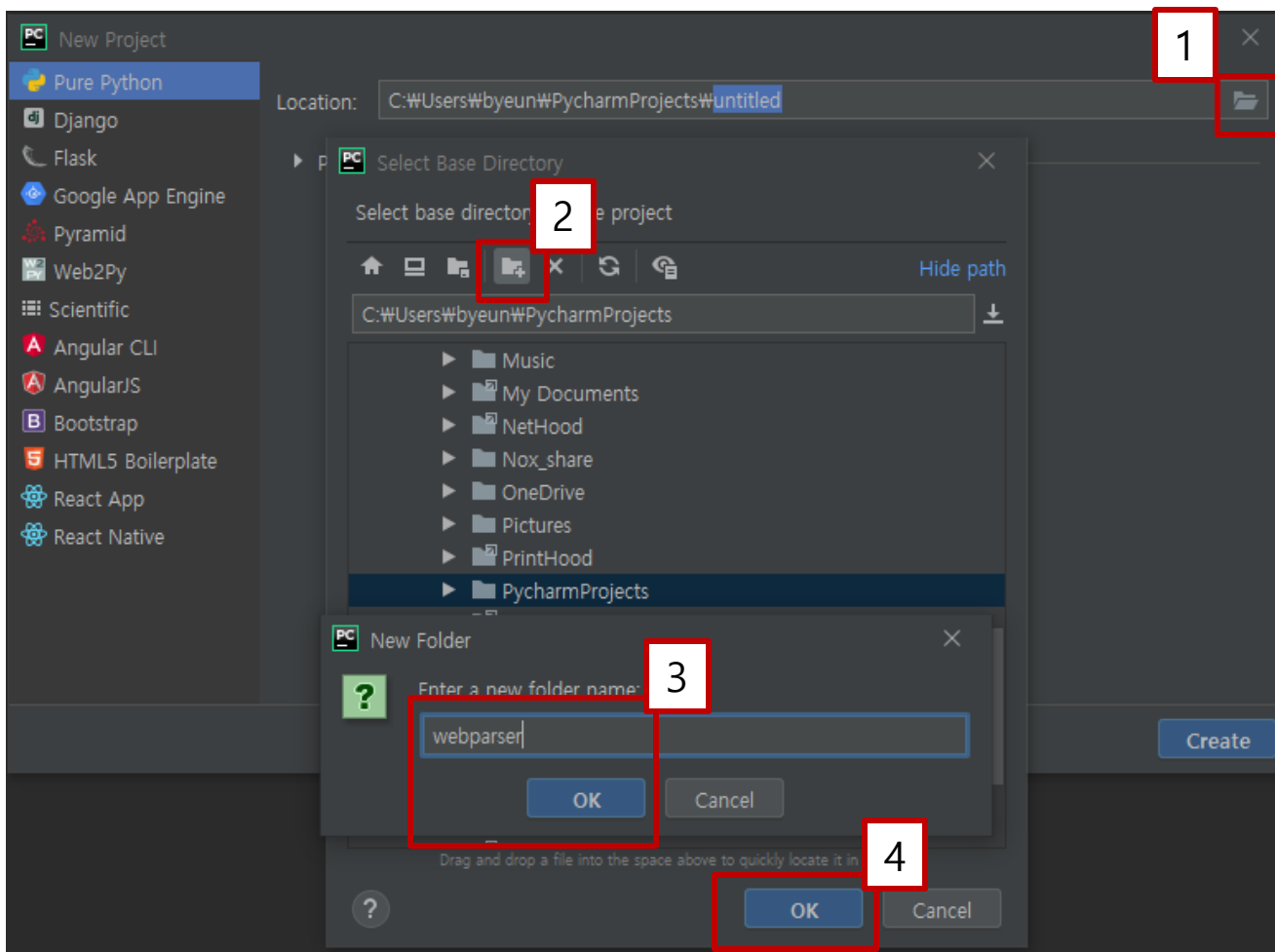
◆ Create Project





Setting(3/16)

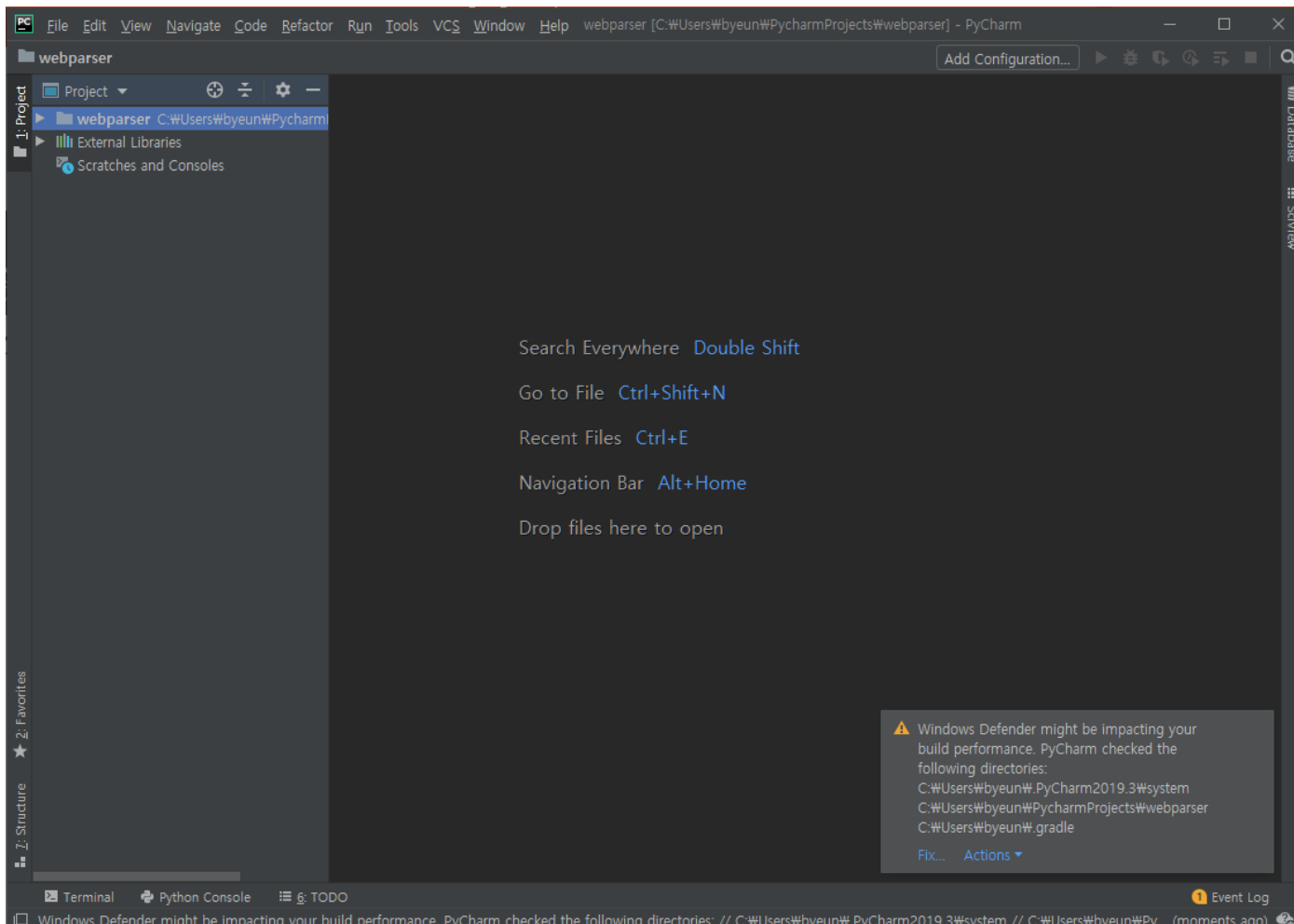
◆ Create Project





Setting(4/16)

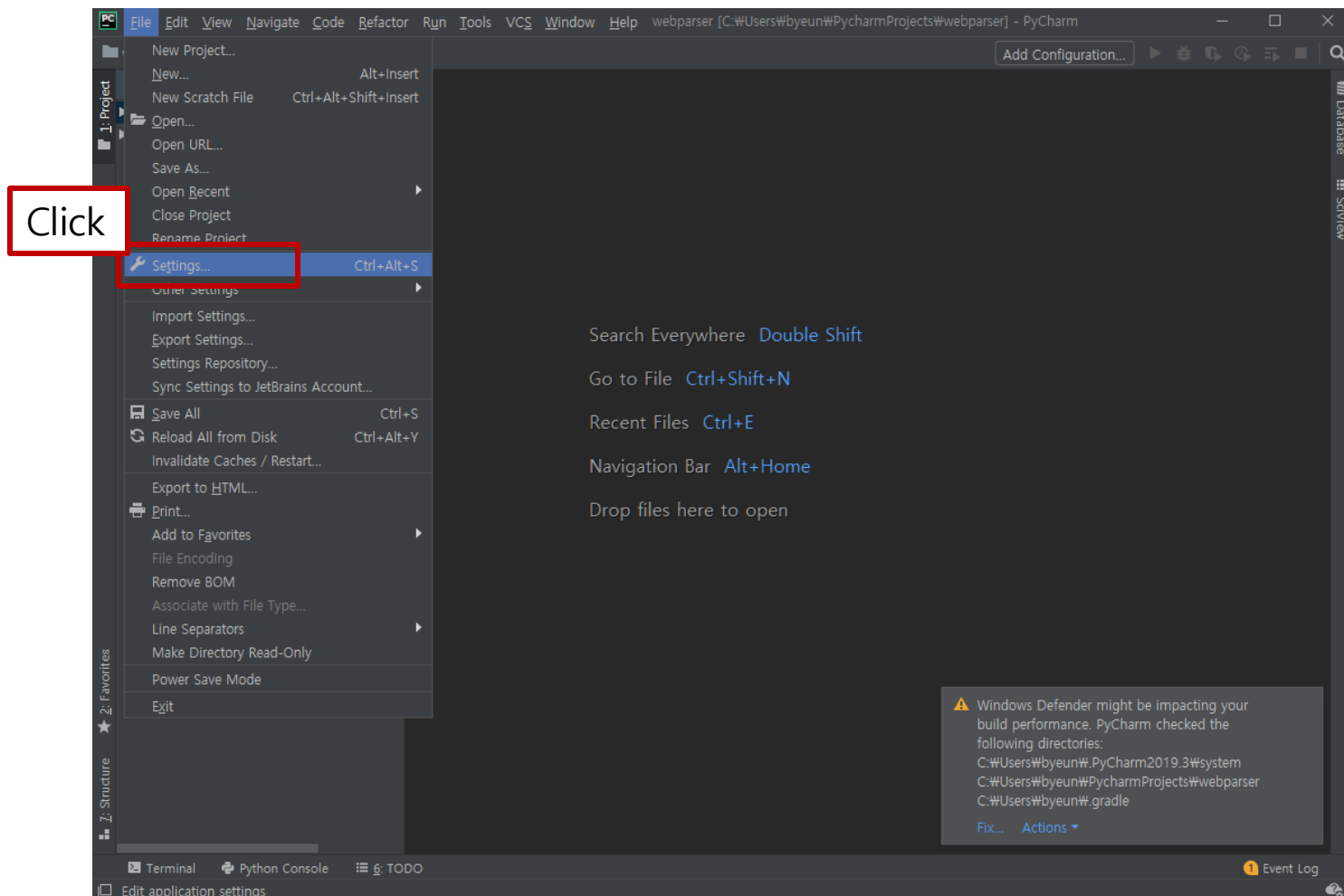
◆ Create Project





Setting(5/16)

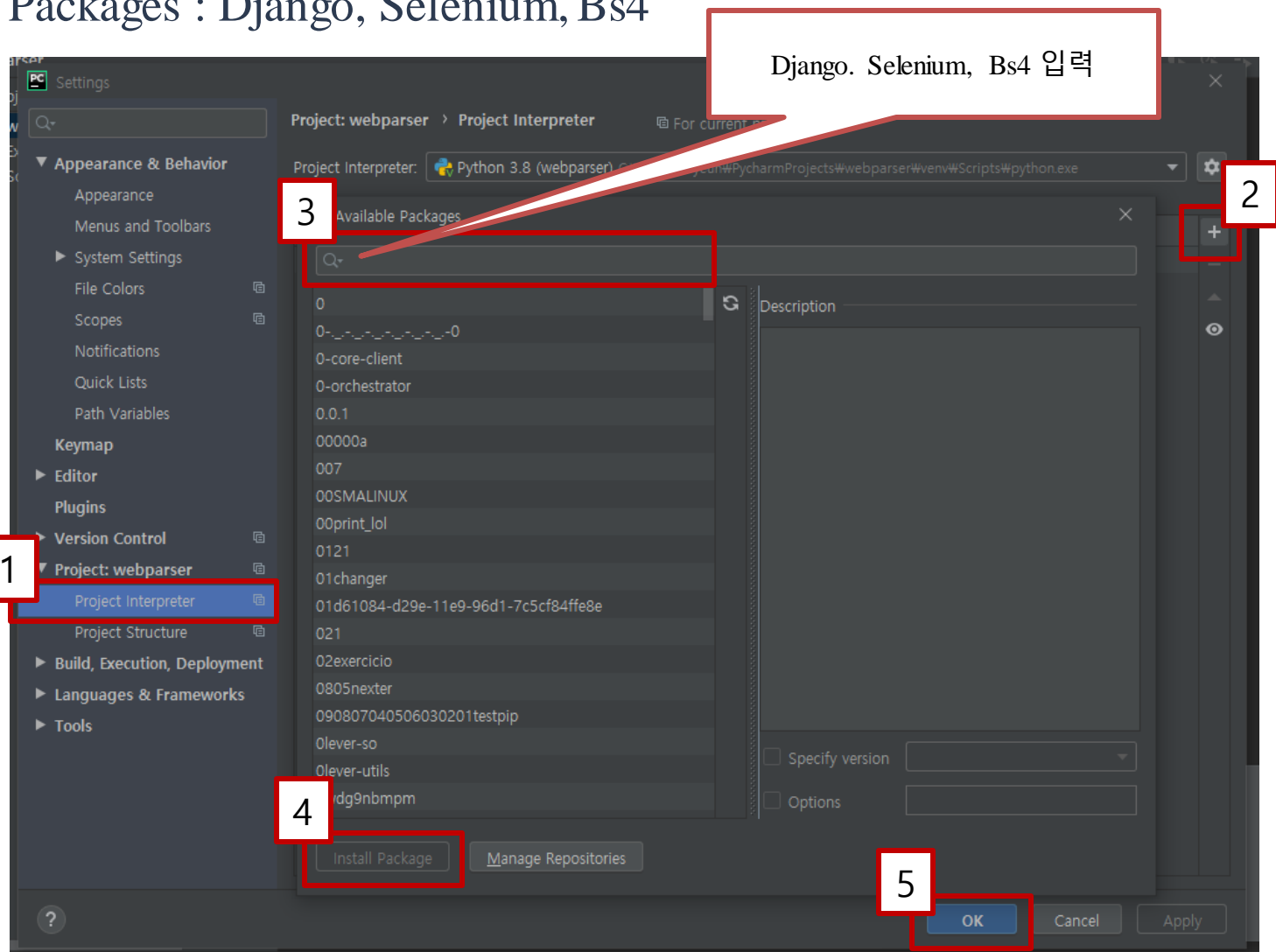
◆ Install Packages





Setting(6/16)

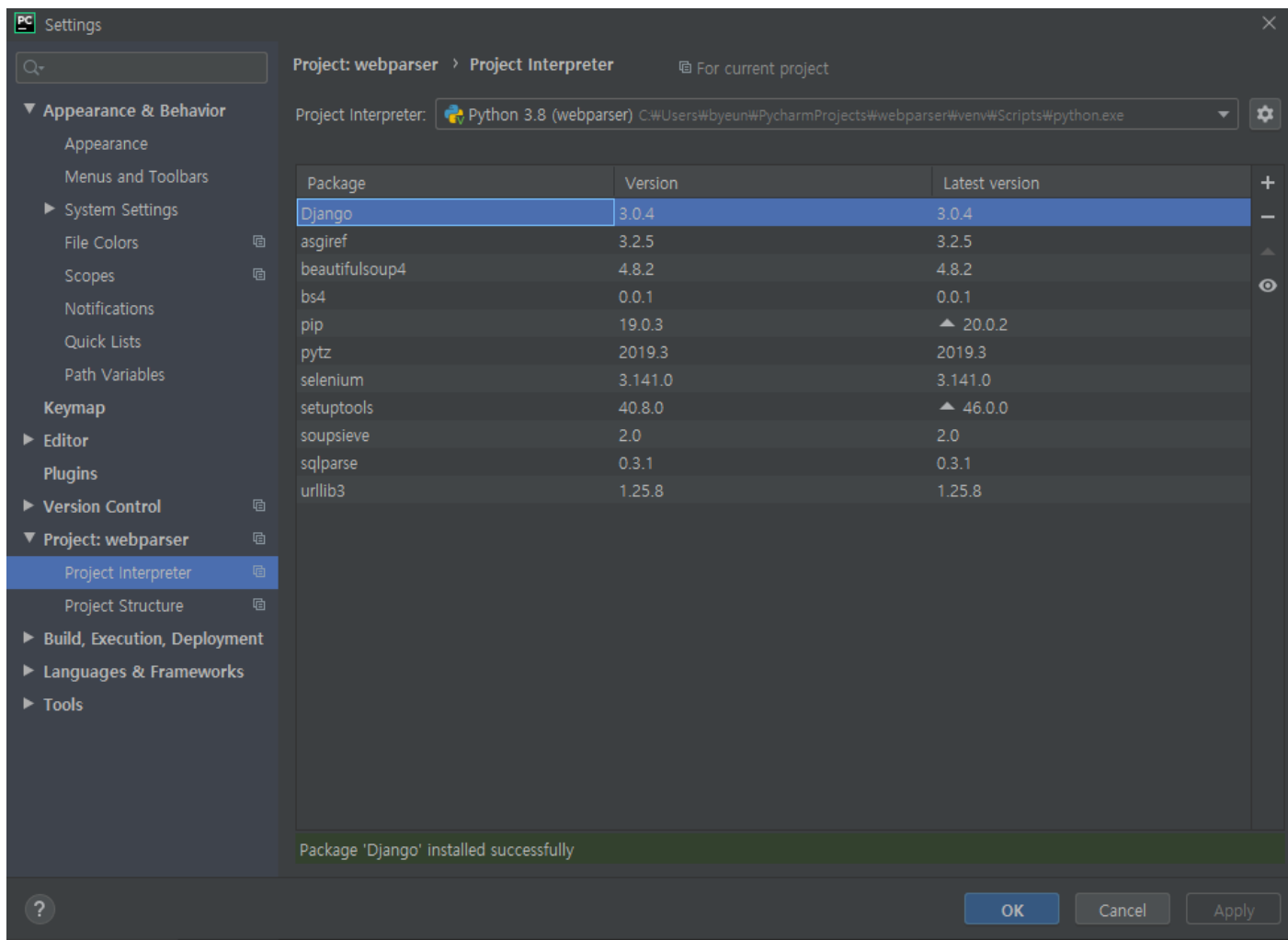
◆ Install Packages : Django, Selenium, Bs4





Setting(7/16)

◆ Install Packages





Setting(8/16)

◆ Create django project and Django app

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The left sidebar shows the Project view with the following structure:

- Project
- webparser (C:\Users\byeun\PycharmP)
 - venv library root
 - webparser
 - parsed_data
 - webparser
 - manage.py
 - External Libraries
 - Scratches and Consoles

The right sidebar shows the Search Everywhere section with the following options:

- Search Everywhere (Double Shift)
- Go to File (Ctrl+Shift+N)
- Recent Files (Ctrl+F)

The bottom panel shows the Terminal view with the following commands and output:

```
Microsoft Windows [Version 10.0.18363.720]
(c) 2019 Microsoft Corporation. All rights reserved.

(venv) C:\Users\byeun\PycharmProjects\webparser>django-admin startproject webparser

(venv) C:\Users\byeun\PycharmProjects\webparser>cd webparser

(venv) C:\Users\byeun\PycharmProjects\webparser\webparser>python manage.py startapp parsed_data

(venv) C:\Users\byeun\PycharmProjects\webparser\webparser>
```

The bottom status bar shows the following tabs: 6: TODO, Terminal, and Python Console.





Setting(9/16)

◆ Create django project and Django app

```
25 # SECURITY WARNING: don't run with debug...
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'parsed_data',
41 ]
42
```





Setting(10/16)

◆ Add path

Settings

Project: webparser > Project Interpreter For current project

Project Interpreter: Python 3.8 (webparser) C:\Users\Wbyeun\PycharmProjects\Wwebparser\venv\Scripts\python.exe

Package	Version	Latest version
Django	3.0.4	3.0.4
asgiref	3.2.5	3.2.5
beautifulsoup4	4.8.2	4.8.2
bs4	0.0.1	0.0.1
pip	19.0.3	▲ 20.0.2
pytz	2019.3	2019.3
selenium	3.141.0	3.141.0
setuptools	40.8.0	▲ 46.0.0
soupsieve	2.0	2.0
sqlparse	0.3.1	0.3.1
urllib3	1.25.8	1.25.8

Package 'Django' installed successfully

OK Cancel Apply

1

2

Add...

Show All...

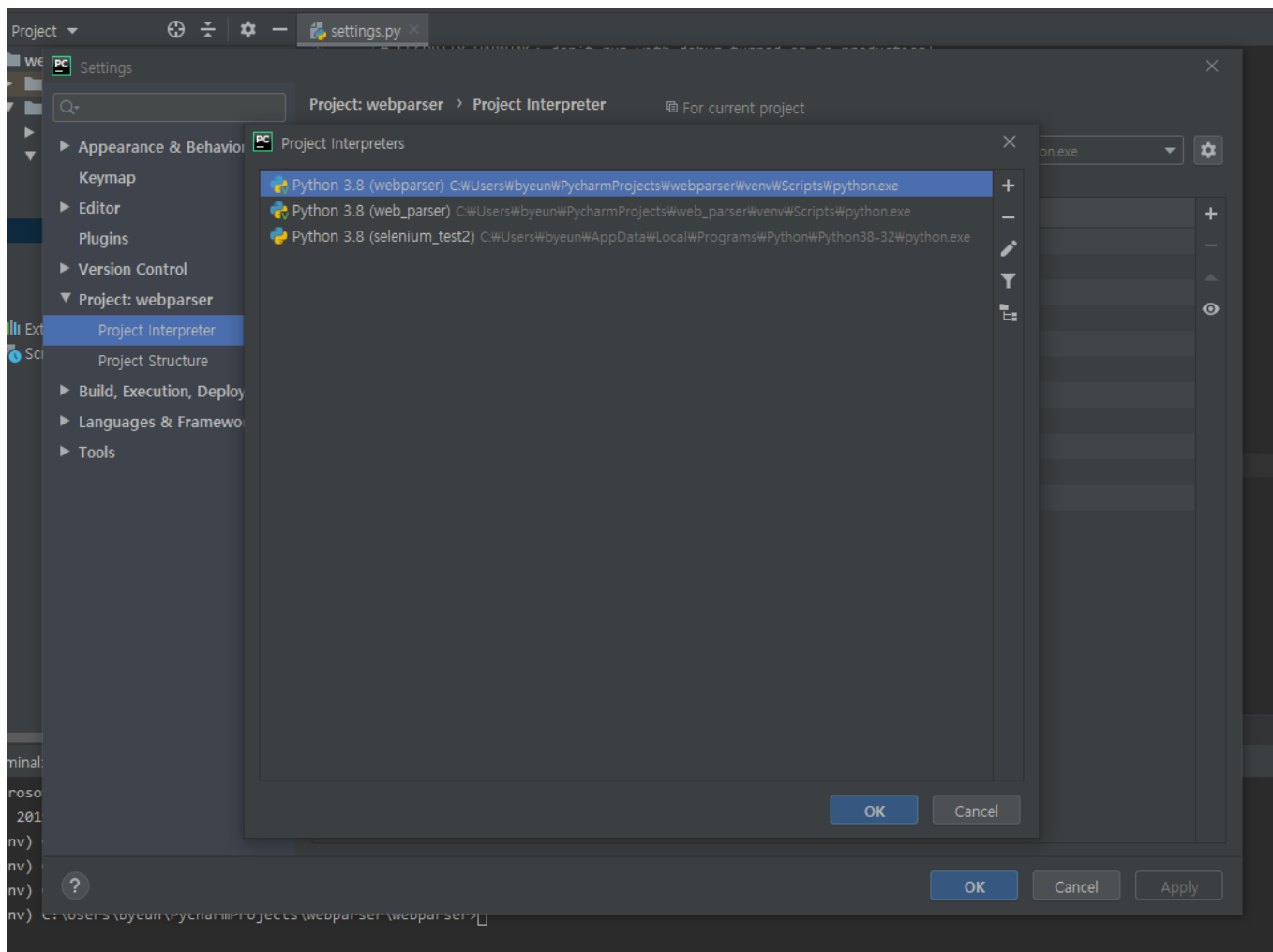
+





Setting(11/16)

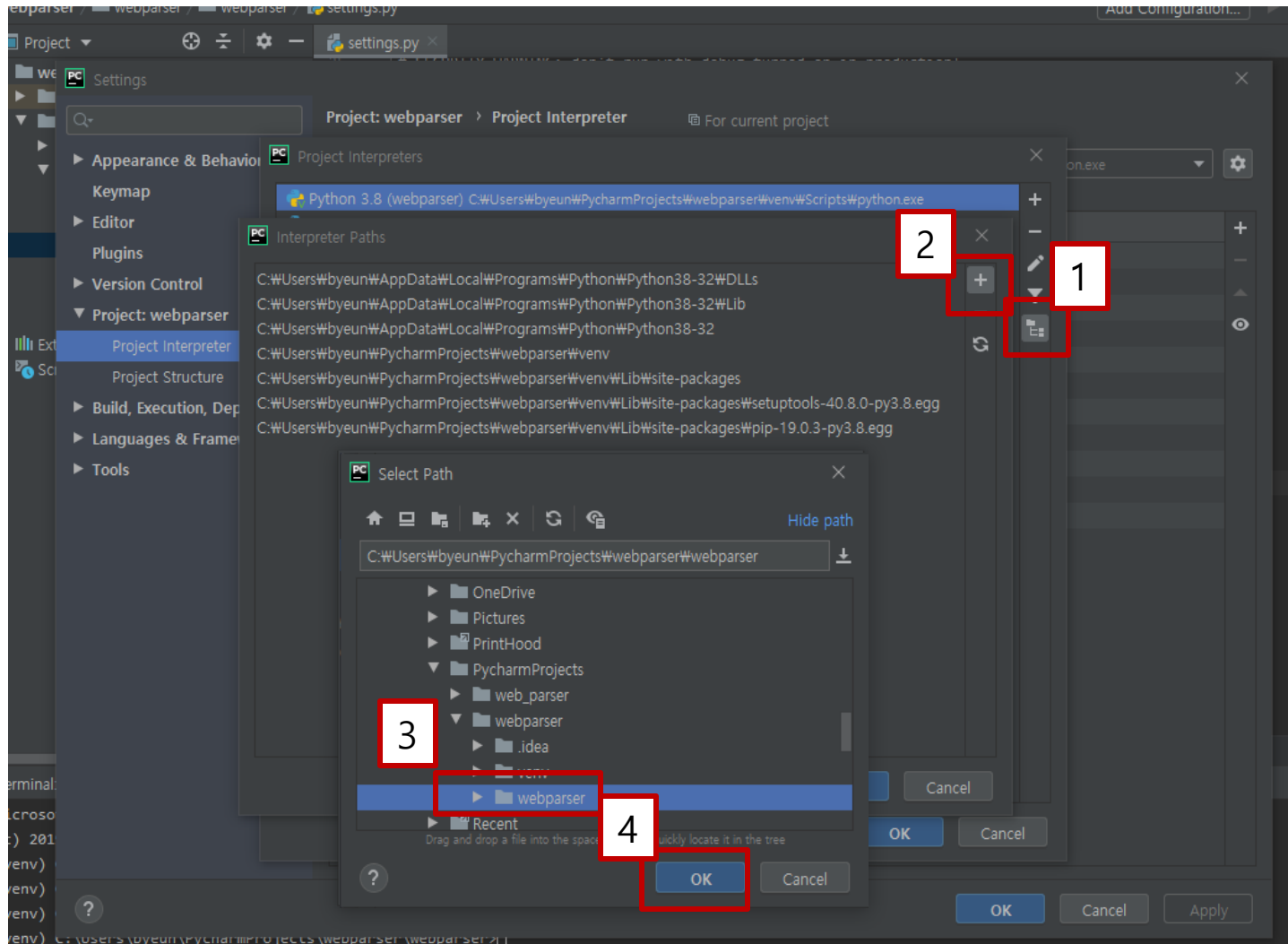
◆ Add path





Setting(12/16)

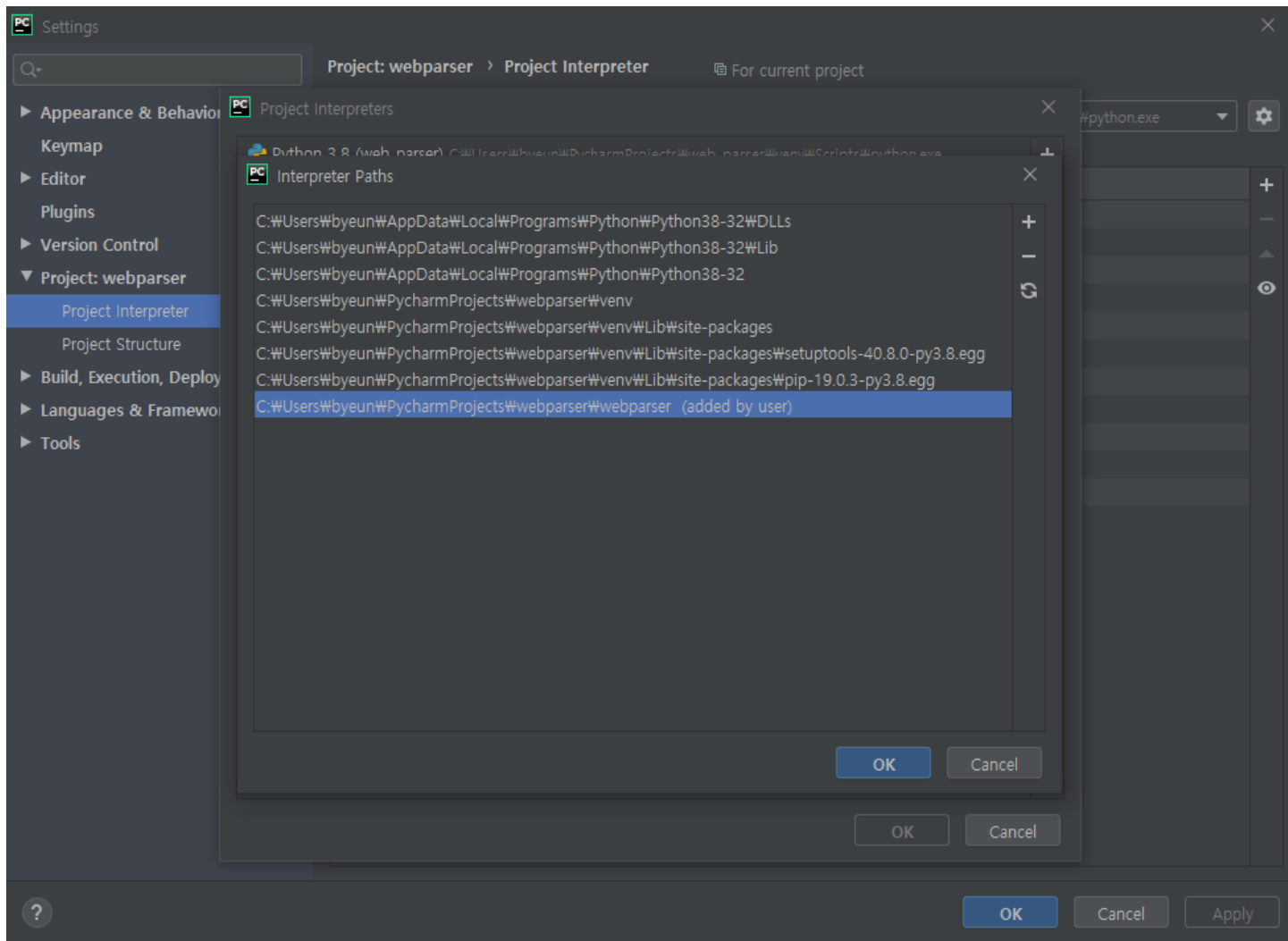
◆ Add path





Setting(13/16)

◆ Add path





Setting(14/16)

◆ Create DB

❖ models.py

```
from django.db import models
```

#카페 글 목록을 저장하기 위한 DB Model

```
class CafeData(models.Model):
```

```
    # title = Title of post list
```

```
    title = models.CharField(max_length=200)
```

```
    # link = Post link address
```

```
    link = models.URLField()
```

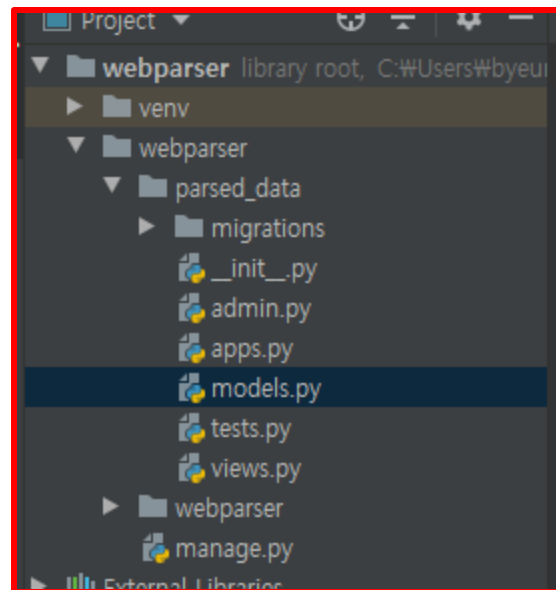
```
    # time = Posted time
```

```
    time = models.TimeField()
```

```
#django admin page title overload
```

```
def __str__(self):
```

```
    return self.title
```





Setting(15/16)

◆ Create DB

❖ Typing in Terminal

python manage.py migrate

python manage.py makemigrations parsed_data

python manage.py migrate

```
Terminal: Local x +
(venv) C:\Users\byeun\PycharmProjects\webparser\webparser>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
```

```
Terminal: Local x +
(venv) C:\Users\byeun\PycharmProjects\webparser\webparser>python manage.py makemigrations parsed_data
Migrations for 'parsed_data':
  parsed_data\migrations\0001_initial.py
    - Create model CafeData

(venv) C:\Users\byeun\PycharmProjects\webparser\webparser>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, parsed_data, sessions
Running migrations:
  Applying parsed_data.0001_initial... OK
```





Setting(16/16)

◆ Create DB SuperUser

❖ Typing in Terminal

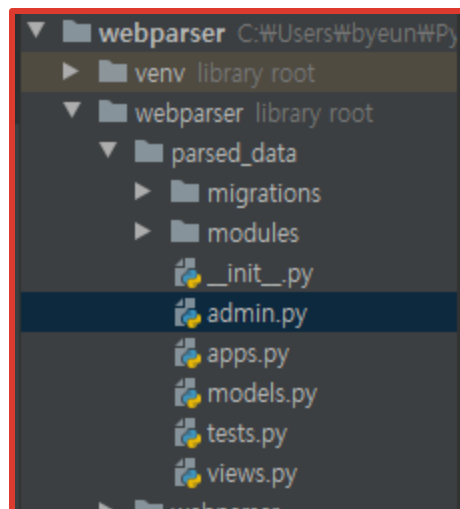
python manage.py createsuperuser

```
Terminal: Local x +
(venv) C:\Users\byeun\PycharmProjects\webparser\webparser>python manage.py createsuperuser
Username (leave blank to use 'byeun'): admin
Email address:
Password:
Password (again):
The password is too similar to the username.
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

◆ Register Admin to the app

```
from django.contrib import admin
from models import CafeData
```

```
# Register your models here.
admin.site.register(CafeData)
```





Programming(1/10)

◆ parser.py

```
from selenium import webdriver
```

```
if __name__ == '__main__':
```

```
    # Download chromedriver location setting
```

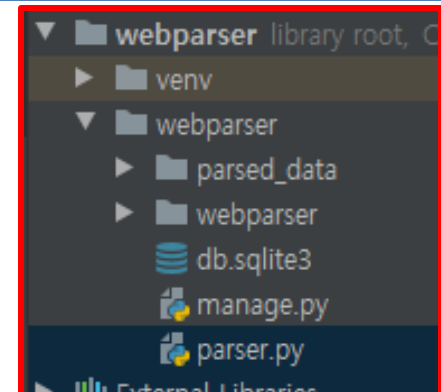
```
    driver = webdriver.Chrome('/Users/byeun/Downloads/chromedriver')
```

```
    # Delay for web loading
```

```
    driver.implicitly_wait(3)
```

```
    #url access
```

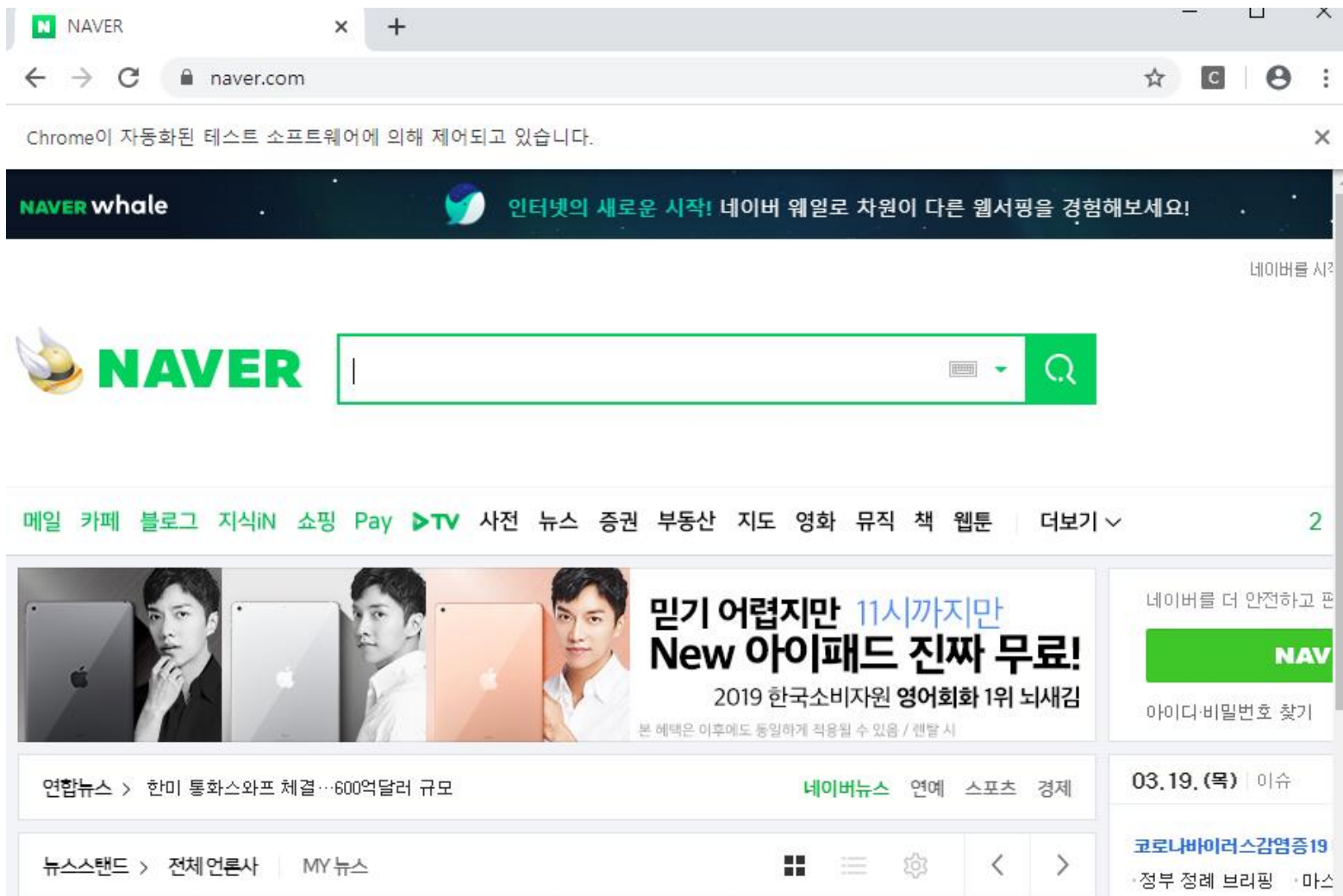
```
    driver.get('https://www.naver.com')
```





Programming(2/10)

◆ parser.py : result





Programming(3/10)

◆ café_crawling.py(1/6)

```
from bs4 import BeautifulSoup
import time
```

```
# crawling data format
```

```
class article_data:
```

```
# cafe-menu search and page change
```

```
def parse_cafe(driver, query, data_time):
```

```
# change pages until find the data_time want
```

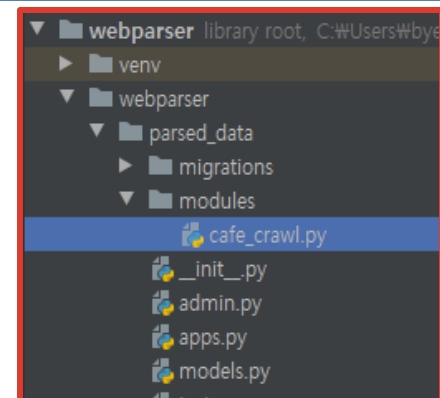
```
def page_cafe(driver, data, data_time):
```

```
# page title and link and time crawling
```

```
def crawling_page(driver, data, data_time):
```

```
# check until data_time
```

```
def time_check(cafe_time, data_time):
```





Programming(4/10)

◆ café_crawling.py(2/6)

crawling data format

class article_data:

def __init__(self, title, link, time):

title is article

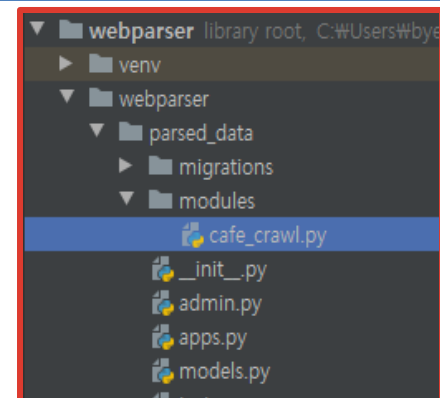
link is article link

time is article time

self.title = title

self.link = link

self.time = time





Programming(5/10)

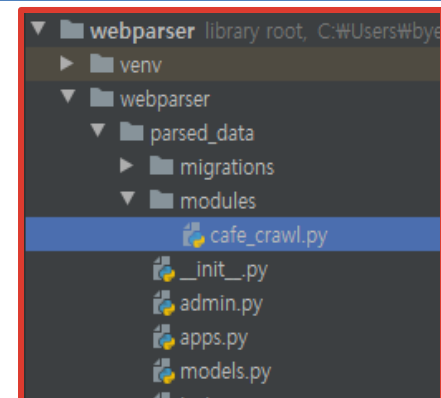
◆ café_crawling.py(3/6)

```
# cafe-menu search and page change
def parse_cafe(driver, query, data_time):
    # query is cafe-menu / data_time is time chosen
    # url access
    driver.get("https://cafe.naver.com/joonggonara")
    time.sleep(1) # wait for page loading
    # find cafe-menu and click
    driver.find_element_by_partial_link_text(query).click()
    time.sleep(1)
    driver.switch_to.frame('cafe_main')

    data = [] # data storage

    t = True
    while t:
        # page crawl
        t = page_cafe(driver, data, data_time)
        if t != '다음':
            t = False

    return data
```





Programming(6/10)

◆ café_crawling.py(4/6)

change pages until find the data_time want

```
def page_cafe(driver, data, data_time):
```

```
    # data is storing data for article / data_time is time chosen
```

```
    soup = BeautifulSoup(driver.page_source, 'html.parser')
```

```
    # hf is page navigation crawl
```

```
    hf = soup.select('div.prev-next > a')
```

```
    count = 15 # count is maximum number of article
```

```
    for t in hf:
```

```
        if count < 14:
```

```
            return count
```

```
        elif t.text == '다음':
```

```
            driver.find_element_by_link_text(t.text).click()
```

```
            return t.text
```

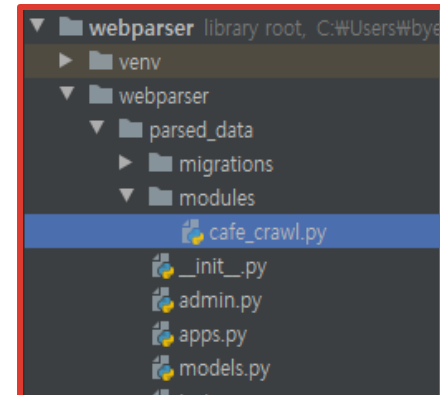
```
        elif t.text == '이전':
```

```
            continue
```

```
    else:
```

```
        driver.find_element_by_link_text(t.text).click()
```

```
        count = crawling_page(driver, data, data_time)
```





Programming(7/10)

◆ café_crawling.py(5/6)

page title and link and time crawling

def crawling_page(driver, data, data_time):

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

 # article html format crawl

 article = soup.select('div:nth-child(6) > table > tbody > tr > td > div
 > div > a.article')

 # article time html format crawl

 article_time = soup.select('div:nth-child(6) > table > tbody > tr > td.td_date')

 count = 0

 for t in article:

 # Comparison of article_time and data_time

 ti = time_check(article_time[count], data_time)

 if ti is True: # if article_time equals data_time

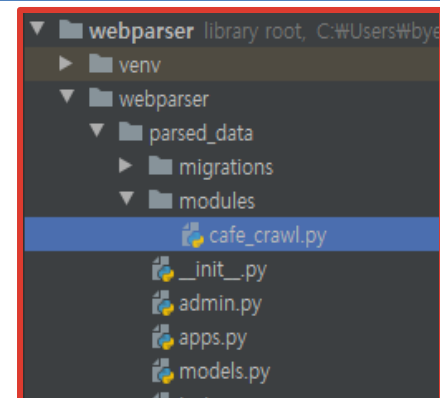
 break

 else:

 data.append(article_data(t.text.strip(), t.get('href'), article_time[count].text))

 count += 1

return count





Programming(8/10)

◆ café_crawling.py(6/6)

```
# check until data_time
```

```
def time_check(cafe_time, data_time):
```

```
    if cafe_time.text.find(':') == -1
```

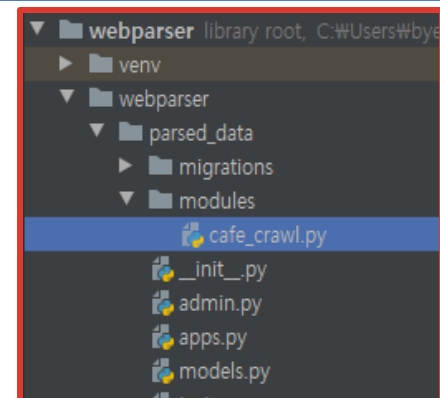
```
        or cafe_time.text < data_time:
```

```
            t = True
```

```
    else :
```

```
        t = False
```

```
    return t
```



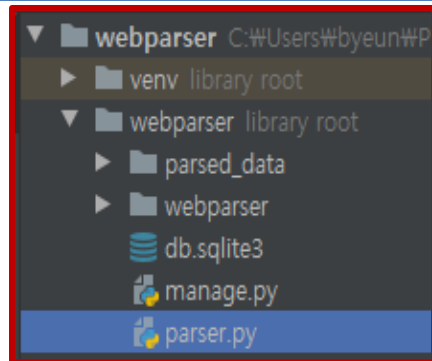


Programming(9/10)

◆ parser.py(1/2)

```
from selenium import webdriver
from parsed_data.modules.cafe_crawl import parse_cafe
import os
import django
os.environ.setdefault("DJANGO_SETTINGS_MODULE", "webparser.settings")
django.setup()
from parsed_data.models import CafeData

if __name__ == '__main__':
    a_list = '컴퓨터'
    c_time = '21:00'
    # Download chromedriver location setting
    driver = webdriver.Chrome('/Users/byeun/Downloads/chromedriver')
    driver.implicitly_wait(3) # Delay for web loading
```





Programming(10/10)

◆ parser.py(2/2)

try:

```
data = parse_cafe(driver, a_list, w_time)
```

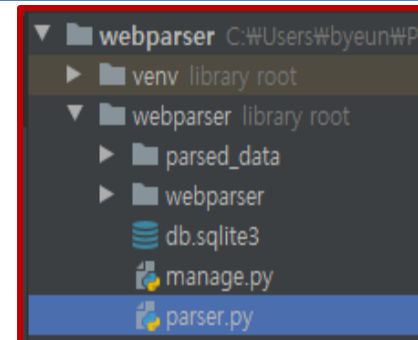
for t in data:

```
CafeData(title=t.title, link=t.link, time=t.time).save()
```

the end of work

finally:

```
driver.quit()
```





Result(1/7)

◆ Run parser.py

- ❖ URL : <https://cafe.naver.com/joonggonara>
- ❖ time : 21:00
- ❖ cafe menu : 컴퓨터

>>When saving DB after crawling

```
if __name__ == '__main__':  
    finally
```

Run: parser ×

C:\Users\byeun\PycharmProjects\webparser\venv\Scripts\python.exe

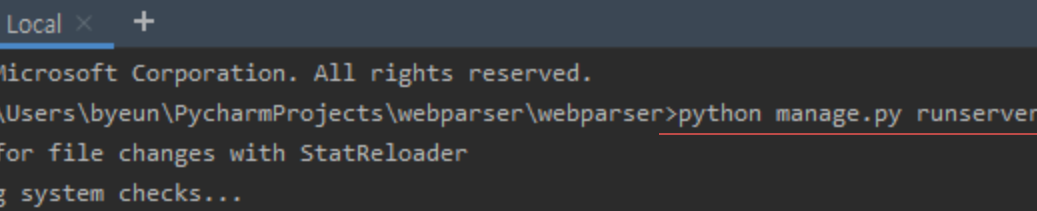
Process finished with exit code 0

4: Run 6: TODO Terminal Python Console





❖ Tying in Terminal



The screenshot shows a PyCharm terminal window with the following output:

```
if __name__ == '__main__':  
  
Terminal: Local x +  
(c) 2019 Microsoft Corporation. All rights reserved.  
(venv) C:\Users\byeun\PycharmProjects\webparser\webparser>python manage.py runserver  
Watching for file changes with StatReloader  
Performing system checks...  
  
System check identified no issues (0 silenced).  
March 22, 2020 - 03:59:36  
Django version 3.0.4, using settings 'webparser.sett  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.  
|
```

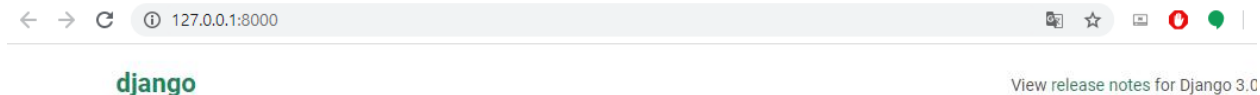
A red box highlights the URL `http://127.0.0.1:8000/`, and a red box with the text "Click" points to it, indicating where to click to open the application in a browser.

Click



Result(3/7)

◆ Web URL : 127.0.0.1:8000



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.





Result(4/7)

- ◆ Web URL : 127.0.0.1:8000/admin
 - ❖ Username and Password insert after Log in

Django administration

Username:

Password:

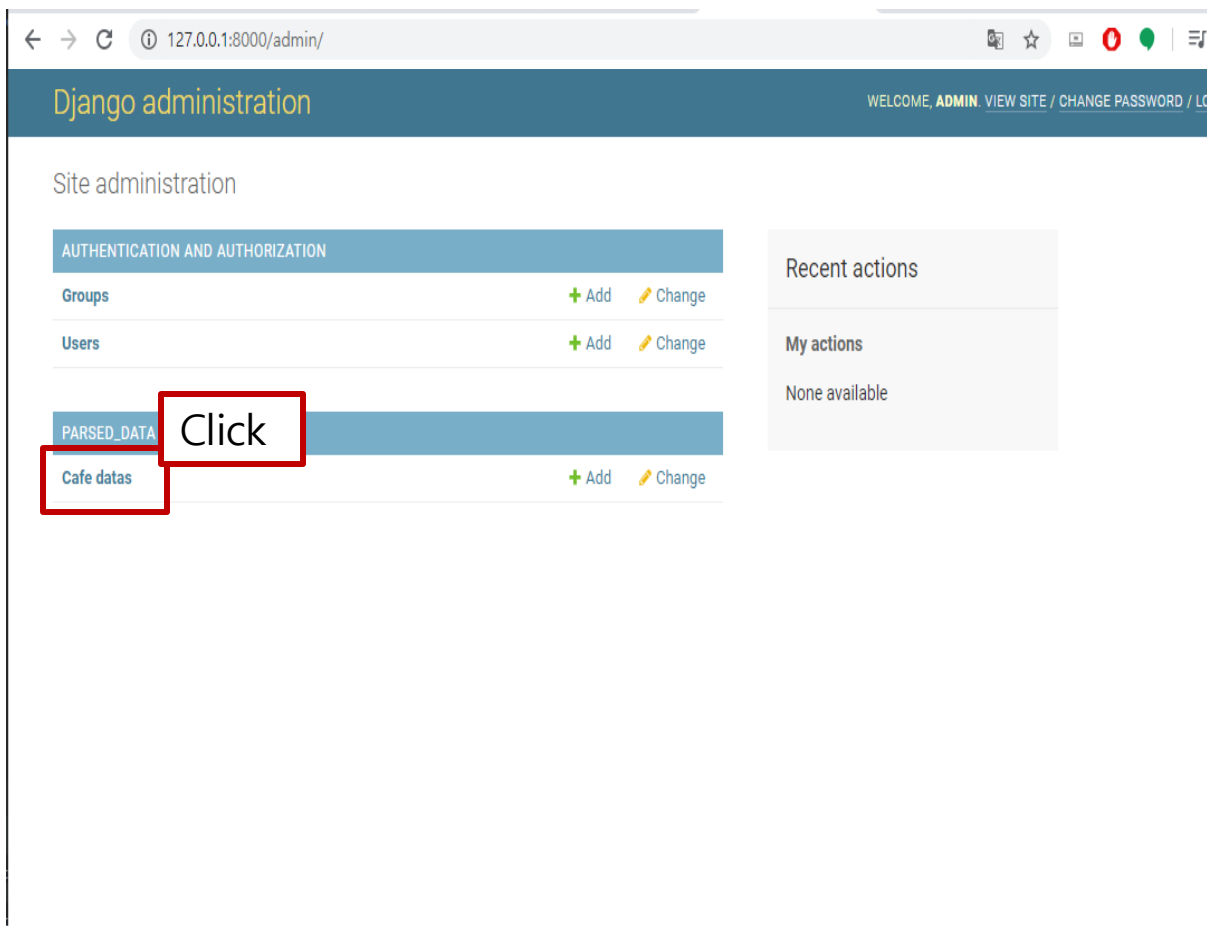
Log in





Result(5/7)

◆ Web URL : 127.0.0.1:8000/admin





Result(6/7)

◆ Web URL : 127.0.0.1:8000/admin

← → ↻ ⓘ 127.0.0.1:8000/admin/parsed_data/cafedata/ 🔍 ☆ 📄 |

Django administration WELCOME, **ADMIN**. [VIEW SITE](#) / [CHANGE PASSWORD](#)

Home › Parsed_Data › Cafe datas

Select cafe data to change

Action: 0 of 1 selected

<input type="checkbox"/>	CAFE DATA
<input type="checkbox"/>	모니터의 패널선택?

1 cafe data





Result(7/7)

◆ Web URL : 127.0.0.1:8000/admin

Django administration

WELCOME, **ADMIN**. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home › [Parsed_Data](#) › [Cafe datas](#) › 모니터의 패널선택?

Change cafe data

HISTORY

Title:

모니터의 패널선택?

Link:

/ArticleRead.nhn?clubid=10050146&page=1&menuid=155&boardty

Time:

02:36:00

Now | ⌚

Note: You are 9 hours ahead of server time.

Delete

Save and add another

Save and continue editing

SAVE





Reference

- ◆ <https://docs.djangoproject.com/ko/3.0/intro/>
- ◆ <https://beomi.github.io/gb-crawling/>

