

# 스크린샷 모듈(screenshooter) 사용 가이드

좋은징조 - 이무송

## 모듈 개요

qr코드 미리보기 서비스를 제공하기 위해서는 해당 사이트의 정보를 클라이언트가 직접 접속 하지 않고 해당 사이트의 정보를 제공해야 한다. 미리보기 서비스 중 제공되어야 하는 기능 중 하나는 해당 사이트의 겉 모습이라고 생각했기 때문에. 이 모듈을 개발했다.

qr코드를 찍어서 url값이 서버에 전송이 되고 나면 해당 url의 사이트에 접속을 하여 화면 캡처 후 이미지 경로를 제공하여 클라이언트 측에서 이미지를 확인할 수 있게 만들어야 한다.

## 모듈 개발 환경

모듈을 개발하기 위해선 다음과 같은 파이썬 모듈 패키지와 프로그램이 필요하다

- selenium - 웹 테스트용으로 개발된 모듈로서 스크린샷 캡처 뿐만 아니라 웹 크롤링, 클릭 이벤트 처리등 다양한 기능을 제공한다, 현재 역할을 클라이언트 대신, 사이트에 접속하여 스크린샷을 캡처하는 기능을 담당한다.
- webdriver(chrome) - selenium 모듈을 사용하기 위해서는 webdriver 객체가 필요한데, 실제 웹 사이트를 접속하여 내가 작성한 스크립트를 처리하는 프로그램이다. Chrome, firefox, phantomJS 등이 있으며 이 모듈에선 chrome webdriver를 사용한다.

## 개발환경 구축

**selenium 모듈 설치** Selenium 모듈을 설치하기 위해서는 아래와 같은 명령어를 입력해 준다.

```
(env)$pip install selenium
```

이렇게 하면 손 쉽게 selenium 모듈을 설치할 수 있다.

**webbrowser와 webdriver 설치** Webdriver를 원활하게 동작시키기 위해서는 webbrowser를 설치한 뒤 webdriver를 설치해야 한다. 스크린샷 모듈은 chrome webdriver를 사용할 것이고 chrome버전에 따라 <http://chromedriver.chromium.org/downloads> 에서 해당 driver를 다운 받는다.

**환경 변수 설정** selenium이 webdriver를 온전히 인식하기 위해서는 환경변수를 설정할 필요가 있다. webdriver를 압축을 풀어서 해당 디렉토리를 PATH 변수에 추가 해준다. Linux 기준으로 홈 디렉토리 안의 자신이 bash 셸을 사용한다면 .bashrc에 다음과 같은 줄을 추가하자

```
PATH=$PATH:/path/to/webdriver/directory
```

## 모듈 정보

**기능** Screenshooter 객체를 이용해 스크린샷을 저장하고 스크린샷 경로를 반환해주는 역할을 한다.

**객체 정보**

## property

- url - 접속할 url을 저장한다. http:// 나 https:// 로 시작해야 한다.
- dir\_path - 이미지를 저장할 directory 경로를 설정하기 위해 필요하다. 스크린샷을 찍으면 해당 directory안에 img\_name으로 이미지가 저장된다.
- img\_name - 스크린샷을 찍은 뒤 timestamp로 시작하는 이름이 저장된다. Shot 메소드에서 최종적으로 img\_name을 반환해준다.

## method

- shot - 설정한 url에 접속해 스크린샷을 찍어 설정한 디렉토리 안에 저장하여 이미지 이름을 반환 한다.

다음은 screenshot 모듈의 소스코드 이다. **screenshooter.py**

```
import os
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from datetime import datetime
mobile_emulation = {

    "deviceMetrics": { "width": 360, "height": 640, "pixelRatio": 3.0 },

    "userAgent": "Mozilla/5.0 (Linux; Android 4.2.1; en-us; Nexus 5 Build/JOP40D)
    AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.166 Mobile Safari/535.19" } #사용할
웹 드라이버의 창 크기와 디바이스 정보를 수정하는 코드다

class Screenshooter:
    def __init__(self,url,dir_path):
        self.url = url
        self.dir_path = dir_path
        self.img_name = None

    def shot(self):
        try:
            # 드라이버에서 사용할 옵션을 설정하는 코드다
            chrome_options = Options()

            # 웹 드라이버를 실행할 때 브라우저가 보이지 않게 해준다.
            chrome_options.add_argument("headless")

            # 웹 드라이버의 창을 설정해주는 옵션이다
            chrome_options.add_argument("window-size=360,640")

            # 불필요한 자원을 소비하지 않기 위해 gpu 자원을 쓰지 않기로 했다.
            chrome_options.add_argument("disable-gpu")

            #모바일 사이트나 모바일로 접속 할때 화면을 찍기 위해서 설정해 준 옵션이다.
            chrome_options.add_experimental_option("mobileEmulation", mobile_emulation)

            driver = webdriver.Chrome(chrome_options = chrome_options) # 웹 드라이버를 실행시
            #키는 코드이다 chrome_options에서 위에서 설정한 옵션들을 적용해주고 있다
            print("webdriver up!")
            driver.get(self.url) #실제 사이트에 접속하는 코드다
```

```

        now = str(datetime.now().timestamp()) #스크린샷 이름을 timestamp로 하기 위해
timestamp 값을 문자열로 변환하고 있다.
        self.img_name = ''.join([now, '_image.png']) #스크린샷 이름을 timestamp_image.png
로 설정하여 img_name에 저장한다.
        full_path = os.path.join(self.dir_path, self.img_name)
        driver.implicitly_wait(3)
        what = driver.save_screenshot(full_path) #실제 스크린샷을 찍는 코드이다 스크린샷 전체
저장경로를 설정하고 해당 경로에 스크린샷을 저장한다.

        print(what) # 저장경로를 찍어준다
        driver.close() # 실행한 웹 드라이버를 종료시켜준다.
        return self.img_name
    except Exception as e:
        print("[debug] driver load ERROR :",e)

```

## 사용 예제

현재 프로젝트에 screenshoter는 다음과 같이 적용되어 있다.

```

import os
from django.shortcuts import render
from django.conf import settings
from .screenshoter import Screenshoter

def processing(request, url):
    ... (중략) ...
    #3. get screenshot
    if virInfo.positives:
        sinfo = ScreenshotInfo(qrInfo=qrinfo, imgPath="mal site") #악성 사이트면 따로 스
크린샷을 저장 안한다.
        sinfo.save()
    else:
        saveScreenshotInfo(qrinfo) #screenshoter 작동

    #4 generate serializer
    result = TestSerializer(qrinfo)
    print('data is :', result.data)
    return Response(result.data)
except Exception as e:
    _, _, tb = sys.exc_info()
    print("[DEBUG] validate error : ", e)
    print("[DEBUG] error line no =", tb.tb_lineno)
    qrinfo.delete()
    return Response(qrseri.errors, status=status.HTTP_400_BAD_REQUEST)

... (중략) ...

def saveScreenshotInfo(qrinfo):
    try:
        dir_path = 'app/static/images' #스크린샷이 저장될 디렉토리 경로를 지정 해줘야한다.
        shoter = Screenshoter(qrinfo.url, dir_path)

```

```
imgPath= shoter.shot() #screenshot을 찍으면 image이름을 얻는다.  
  
#DB에 스크린샷 정보를 저장  
sinfo = ScreenshotInfo(qrInfo=qrinfo,imgPath=imgPath)  
sinfo.save()  
except Exception as e:  
    print("[debug] screenshot error",e)
```