

Selenium

Chrome Extension 사용 주의

: Chromedriver는 실행할 때 마다 임시 Profile을 사용하여 새로운 Instance로 Open됨. 이를 해결하기 위해 기존 사용중인 Chrome Profile을 사용하면 된다.

1. Chrome에서 특정 페이지로 이동하는 샘플 스크립트

```
# -*- coding: utf-8 -*-
import time
import sys
from selenium import webdriver
from selenium.webdriver.chrome.options import Options

# Chrome WebDriver를 이용해 Chrome을 실행합니다.
driver = webdriver.Chrome('D:\WAutoTest\Wchromedriver.exe')

# www.google.com으로 이동합니다.
driver.get("http://127.0.0.1:8003")
time.sleep(2)

continue_link = driver.find_element_by_partial_link_text('export_yk.html')
time.sleep(5)

# 해당 링크를 클릭합니다.
continue_link.click()
time.sleep(5)

# WebDriver를 종료합니다. (브라우저 닫기)
#driver.quit()
```

2. Chrome Extension 사용 자동화 테스트

Download the **Chrome Extension Source Viewer** from the Google Web Store.



홈 > 확장 프로그램 > Chrome extension source viewer



Chrome extension source viewer

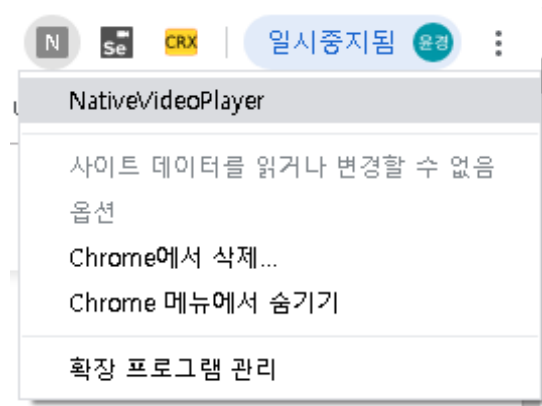
제공자: RobW

★★★★★ 339

개발자 도구

사용자 74,086명

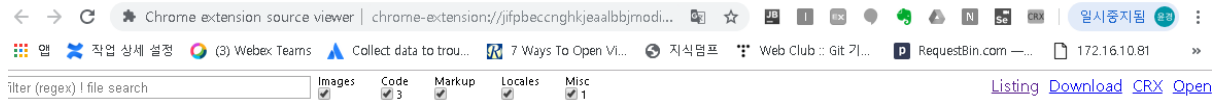
설치할 Chrome Extension 페이지로 이동한다



Chrome Extension Source Viewer 을 실행하여, View source 메뉴를 클릭한다.



설치한 Chrome Extension Source Viewer를 이용해 설치할 ChromeExtension의 .crx 파일을 다운로드 받는다.



Selenium 스크립트에서 아래와 같이 chromedriver 옵션에 다운로드한 .crx Extension 파일을 추가한다.

.crx Extension 파일 추가

```
# Chrome WebDriver를 이용해 Chrome을 실행합니다.
opt = webdriver.ChromeOptions()
opt.add_extension('D:/AutoTest/cmijipmellgcnlagimdoncdjidacklc.crx')
driver = webdriver.Chrome('D:\AutoTest\chromedriver.exe', options=opt)
```

전체스크립트

```
# -*- coding: utf-8 -*-
import time
import sys
from selenium import webdriver
from selenium.webdriver.chrome.options import Options

# Chrome WebDriver를 이용해 Chrome을 실행합니다.
opt = webdriver.ChromeOptions()
opt.add_extension('D:/AutoTest/cmijipmellgcnlagimdoncdjidacklc.crx')
driver = webdriver.Chrome('D:\AutoTest\chromedriver.exe', options=opt)

# www.google.com으로 이동합니다.
driver.get("http://127.0.0.1:8003")
time.sleep(2)

continue_link = driver.find_element_by_partial_link_text('export_yk.html')
time.sleep(5)

# 해당 링크를 클릭합니다.
continue_link.click()
time.sleep(5)

# WebDriver를 종료합니다. (브라우저 닫기)
driver.quit()
```

자동화 테스트 샘플 스크립트

```
# -*- coding: utf-8 -*-
import time
import sys
import os.path
import json

from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.common.exceptions import NoSuchElementException
from selenium.common.exceptions import ElementClickInterceptedException

def CaptureTestCase(webdriver) :
    print('Testing Image Capture Start')

    try:
        CaptureFile=webdriver.find_element_by_id('StillImageFilePath').get_attribute('value')
        if os.path.isfile(CaptureFile):
            os.remove(CaptureFile)
    except NoSuchElementException:
        print('Failure! : NoSuchElementException - Capture', )
        CaptureFile="test.jpg"

    try:
        capture= webdriver.find_element_by_name('Capture')
        capture.click()
        time.sleep(2)

        data = json.loads(webdriver.find_element_by_id('message').text)
        # print (data["status"])

        if data["status"]=='success':
            if os.path.isfile(CaptureFile):
                print('Success Capture! :-)')
            else:
                print('Failure Save Capture File : ', CaptureFile)
        else:
            print('Fail Capture! => Response : ', data["status"])
    except NoSuchElementException:
        print('Failure! : NoSuchElementException - Capture', )
    print('Testing Image Capture End')

def ExportTestCase(webdriver) :
    print('Testing Image Export Video Start')

    try:
        ExportVideoFile=webdriver.find_element_by_id('VideoFilePath').get_attribute('value')
        if os.path.isfile(ExportVideoFile):
            os.remove(ExportVideoFile)
```

```

except NoSuchElementException:
    print('Failure! : NoSuchElementException - VideoFilePath', )
    ExportVideoFile="test.mov"

    try:
#Export 시작 시간 설정
        startTime= webdriver.find_element_by_name('Start')
        startTime.click()
        time.sleep(5)
#Export 시작 시간 Response code 확인
        startTimeRes = json.loads(webdriver.find_element_by_id('message').text)
        if
            startTimeRes["action"]=='backup_start_video'
            and
startTimeRes["status"]=='success':
#Export 종료 시간 설정
            endTime= webdriver.find_element_by_name('END')
            endTime.click()
            time.sleep(5)
#Export 종료 시간 Response code 확인
            endTimeRes = json.loads(webdriver.find_element_by_id('message').text)
            if
                endTimeRes["action"]=='backup_end_video'
                and
endTimeRes["status"]=='success':
                if os.path.isfile(ExportVideoFile):
                    print('Success Export Video! :-))')
                else:
                    print('Failure Save Video File : ', ExportVideoFile)
            else:
                print('Failure Set Export EndTime ')
                print('Fail Export Video! => Response : ', endTimeRes["status"])
        else:
            print('Failure Set Export StartTime ')
            print('Fail Export Video! => Response : ', startTimeRes["status"])

except NoSuchElementException:
    print('Failure! : NoSuchElementException - END')
    print('Testing Export Video End')

def LinkToExportPage(webdriver):
    print('Link Export Page Test Start')

    try:
        continue_link = webdriver.find_element_by_partial_link_text('export_yk.html')
        time.sleep(5)

        # 해당 링크를 클릭합니다.
        try:
            continue_link.click()
            time.sleep(5)
        except ElementClickInterceptedException:
            print('FAIL: Can not link to export_yk.html Page ')
            print('Link Export Page Test End')
            return

```

```

except NoSuchElementException:
    print('FAIL: NoSuchElementException (export_yk.html)')
    print('Link Export Page Test End')
    return

print('PASS: NoSuchElementException (export_yk.html)')
print('Link Export Page Test End')

def Capture_Export_TestSuit_Main():
    # Chrome WebDriver를 이용해 Chrome을 실행합니다.
    opt = webdriver.ChromeOptions()
    opt.add_extension('D:/AutoTest/cmijipmellgcnlaoigimdoncdjidacklc.crx')
    driver = webdriver.Chrome('D:\WAUTOtest\chromedriver.exe', options=opt)

    # www.google.com으로 이동합니다.
    print('Link CKC Test WebSite')
    driver.get("http://127.0.0.1:8003")
    time.sleep(2)

    LinkToExportPage(driver)
    CaptureTestCase(driver)
    ExportTestCase(driver)

    driver.quit()

Capture_Export_TestSuit_Main()

```

API

https://selenium-python.readthedocs.io/api.html#selenium.webdriver.chrome.options.Options.add_experimental_option

문서 : https://www.seleniumhq.org/docs/03_webdriver.jsp#introducing-webdriver

주요 업무

제품 기획

제품 검증

형상관리

제품 인증(CC인증)

상세 업무

제품 기획

- 요구사항 수집
 - ✓ 타 부서(수행, 영업 등)로부터 요구사항 수집
 - ✓ 경쟁사 제품 사용 및 분석을 통한 요구사항 수집
- 요구 사항 명세
 - ✓ 수집된 요구사항을 통한 요구사항 명세서 작성 및 내부 리뷰
- 제품 용어 정의
 - ✓ 제품에서 사용되는 용어 (메뉴, 기술적인 용어등)에 대한 정의 및 문서 작성
- 정책정의서
 - ✓ 사용자 권한 및 세션 유지등과 같은 제품 정책에 대한 정의 및 문서 작성
- 메시지 정의서
 - ✓ 제품에 사용되는 로그, 안내 메시지 등에 대한 정의 및 문서 작성
- 메뉴 구성도
 - ✓ 각 제품 구성별 메뉴 트리 설계 및 문서 작성
- UI/UX 기획서 작성

제품 검증

검증 정책 정의

릴리즈 지표 및 조건 설계

검증 및 결과 보고

검증 결과 보고서 문서 형식 정의 및 보고서 작성

: 각 검증 회차 별 결함 추이 및 검증 결과 작성

릴리즈 검토 보고서 형식 정의 및 보고서 작성

: 최종 릴리즈 검증 결과 보고 및 릴리즈 가능 여부에 대한 QA의견 작성

버전 정책 정의 및 버전 정책 문서 작성

검증

기능 검증

TestCase 설계

Test Scenario 설계

상세 Test Case 설계

API 검증

Rest API Test Case 설계

Rest API 검증

형상관리

소스 및 산출물들에 대한 버전 별 형상관리 정책정의

제품 인증

Windows Server2012 에서 제공되는 Hyper-V를 활용한 가상데스크톱(VDI)솔루션에 대한 국내 CC인증
인증기관 담당자와 인증 진행사항 협의

: 인증에 필요한 산출물 작성

: CC 인증 요건 사항에 대한 검증 및 인증 기관과 검증 진행

주요 업무

개발 프로세스 정의

제품 설치 프로그램 생성 및 배포

검증

형상관리

국가 R&D

문서 작성

상세 업무

개발 프로세스 정의

DevOps 개발방법론에 따른 개발 업무 프로세스 정의

Jira를 통한 개발 프로세스 적용

제품 빌드 및 배포 자동화

빌드 자동화

Jenkins + MSBuild를 통한 C#, C++ 프로그램 자동 빌드

Jenkins + Mono를 통한 C# 프로그램 자동 빌드

패키지 자동화

Jenkins + Innosetup을 통한 Windows 설치 패키지 자동 생성

Jenkins + DockerFile를 통한 DockerImage 자동 생성

배포 자동화

Jira Webhook + Jenkin를 통한 Stable 환경 자동 배포 및 설치

제품 검증

검증 정책 정의

릴리즈 지표 및 조건 설계

검증 및 결과 보고

검증 결과 보고서 문서 형식 정의 및 보고서 작성

: 각 검증 회 차 별 결함 추이 및 검증 결과 작성

릴리즈 검토 보고서 형식 정의 및 보고서 작성

: 최종 릴리즈 검증 결과 보고 및 릴리즈 가능 여부에 대한 QA의견 작성

버전 정책 정의 및 버전 정책 문서 작성

검증

검증 계획 및 설계

검증 계획

Test Scenario 설계

상세 Test Case 설계

Rest API Test Case 설계

API Test Case 설계

성능 검증 지표 및 Case 설계

검증

수동 검증

Postman을 통한 RestAPI 검증

Client를 통한 기능 검증

기능 검증 자동화

Selenium(오브젝트 기반) + Guitar(이미지 기반)을 통한 일부 기능 자동화

성능 검증

JMeter를 통한 성능 검증

정적 분석

JSHint을 통한 Web 정적 분석 및 결과 리포팅

형상관리

Git을 통한 소스 및 산출물들에 대한 버전 별 형상관리

GoogleDrive를 통한 산출물(TC, 매뉴얼 등) 형상관리

국가 R&D

ATC (우수 기술연구 센터) 연구 과제 수행 (5 년 과제)

검증

정량적 목표에 대한 자체 검증

정량적 목표에 대한 공인기관(TTA) 인증 수행

요구 사항에 대한 기능 검증

산출물 작성

개발 산출물 작성

요구사항 명세서

요구사항 추적표

시스템 아키텍처

SW 상세 설계서
인터페이스 정의서
DB 설계서
검증 계획서
Test Case 설계서
검증 결과 보고서

문서 작성

릴리즈 노트 작성

설치 가이드 작성

사용자 가이드 작성

기술 가이드 및 참고 문서 작성

FAQ 작성

제품별 기능 리스트 및 기능 비교 문서 작성

- Jira 운영(사용자 관리, 이슈 처리 Workflow 설계 및 정의, 이슈 스키마 정의 등)

- Confluence 운영 (Confluence 공간 관리, 품질 관리를 위한 자료 관리, 제품 및 버전 별 검증 결과 보고서 템플릿 작성 및 운영등)

5. 제품 사용 방법 교육

- 고객 제품 사용 교육 진행

- 사내 제품 사용 교육 진행 (영업, 기술 수행, 마케팅 등 부서 대상)

- 제품 교육을 위한 샘플 코드 및 샘플 데이터 생성

- 기타 개발 및 시스템 유지 보수 경력을 활용한 팀원 및 동료 기술 교육 지원 등