Ask Company

크롤링 차근차근 시작하기 (2/E) - 중급편

Selenium IDE

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

Selenium IDE (버전 3.6.0 - 2019년 4월 기준)

2019년에 Selenium 4 릴리즈 예정

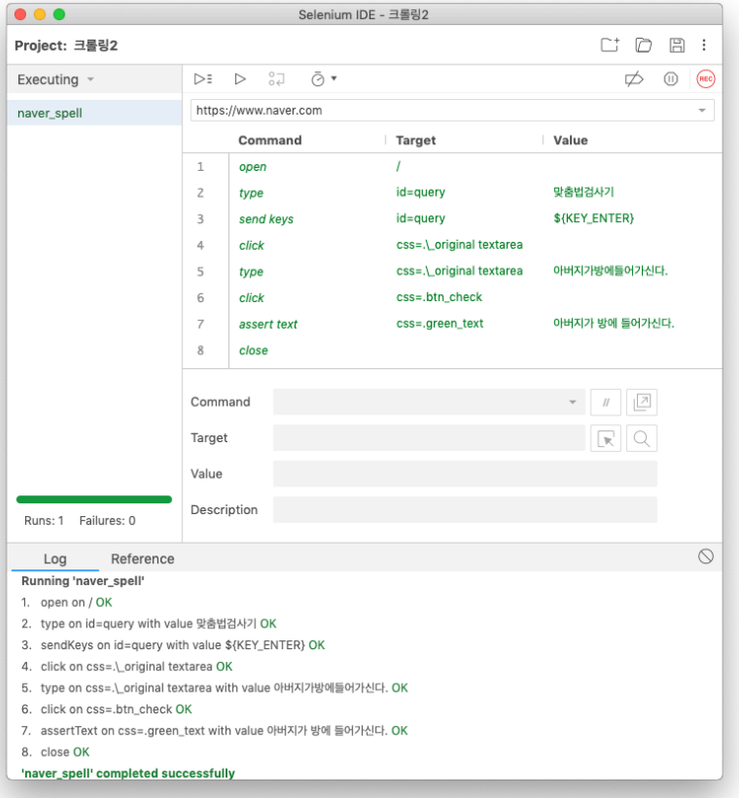
브라우저 UI를 통한 테스트 시나리오를 녹화/재실행

- 크롤링 목적이 아닌, 자동화 및 UI 테스트, QA에 포커스 → 웹서비스 CI에 유용

Continuous Integration

지원 기능

- 1 Project → N Suites → N Tests → N Commands
  - *.side 확장자로 저장 → JSON 포맷
  - 직접 명령을 작성하거나 녹화 기능을 활용
- breakpoint 지원, 예외 발생 시에 Pause 지원
- 부분적인 코드로의 Export (Java/JUnit 지원만 남겨두고, 다른 언어 지원 X)
  - 파이썬 지원도 없어졌습니다. ☹

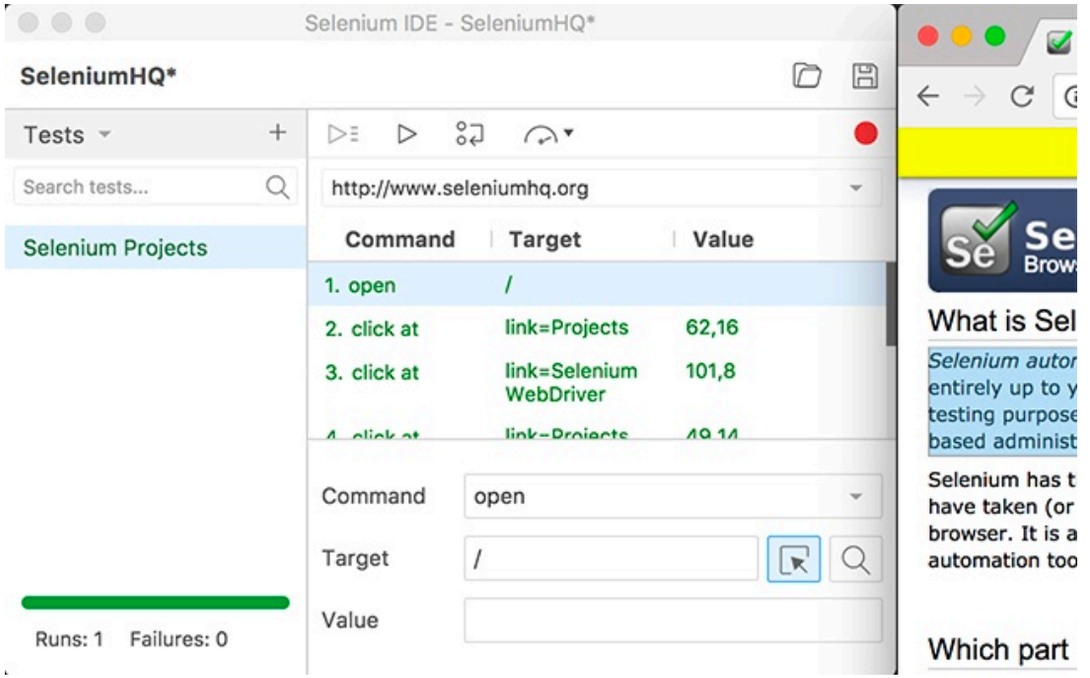


https://www.seleniumhq.org/selenium-ide/docs/en/introduction/getting-started/

브라우저 확장 기술로 만든 GUI 애플리케이션

Chrome/Firefox 확장 플러그인 형태로 애플리케이션 제공

- Chrome 확장 : https://chrome.google.com/webstore/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd
- Firefox 확장 : https://addons.mozilla.org/en-GB/firefox/addon/selenium-ide/



다음 코드를 옮겨본다면 ...

```python
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

with webdriver.Chrome() as driver:
    driver.get("https://www.naver.com")
    ele = driver.find_element_by_id("query")
    ele.send_keys("python" + Keys.ENTER)
```

side 포맷

원본에서 중요부분만 남기고 제거한 코드

```
{
    "id": "29a92b87-6ff0-430a-9aac-58195e48dbb0",
    "version": "2.0",
    "name": "naver",
    "url": "https://www.naver.com",
    "tests": [{
        "name": "naver",
        "commands": [
            { "command": "open", "target": "/"},
            { "command": "setWindowSize", "target": "1234x983"},
            { "command": "type", "target": "id=query",
                                 "value": "python" },
            { "command": "sendKeys", "target": "id=query",
                                     "value": "${KEY_ENTER}" },
            { "command": "close" }
        ]
    }],
    "suites": [{
        "id": "a16b7b27-41c5-468a-985b-5f8cdb271abb",
        "name": "Default Suite",
        "persistSession": false,
        "parallel": false,
        "timeout": 300,
        "tests": []
    }],
    "urls": ["https://www.naver.com/"],
    "plugins": []
}
```

지원 Commands

open : 새 윈도우에서 지정 URL을 열고 페이지 로딩 완료까지 대기

click : 지정 타겟 Element를 클릭, click at : 지정 타겟 Element내 상대좌표를 클릭

type : 입력 필드 (콤보박스, 체크박스 등)에 값 할당

send keys : key 입력 시뮬레이션

run script : 새 script 태그를 생성하여, 현재 테스트 윈도우의 body 태그에 주입하여 실행

set window size : 윈도우를 지정 크기로 변경

select window : 지정 윈도우를 활성화

repeat, repeat if : 반복 수행

...

https://www.seleniumhq.org/selenium-ide/docs/en/api/commands/

Code Samples

Step 1: Python Code

```
import time
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

with webdriver.Chrome() as driver:
    driver.implicitly_wait(10)

    driver.get("https://www.naver.com")
    driver.find_element_by_id("query").send_keys("맞춤법검사기" + Keys.ENTER)

    textarea = driver.find_element_by_css_selector('._original textarea')
    textarea.clear()
    textarea.send_keys("아버지가방에들어가신다.")

    btn = driver.find_element_by_css_selector('.btn_check')
    btn.click()

    time.sleep(1)

    ele = driver.find_element_by_css_selector('._result_text')
    assert ele.text == "아버지가 방에 들어가신다."
```

Step 2 : Pyton/unittest 활용 (1)

```
import time
import unittest
from selenium import webdriver
from selenium.webdriver.common.keys import Keys


class TestNaver(unittest.TestCase):
    def setUp(self):
        self.driver = webdriver.Chrome()
        self.driver.implicitly_wait(10)

    def get_by_id(self, *args):
        return self.driver.find_element_by_id(*args)

    def get_by_css(self, *args):
        return self.driver.find_element_by_css_selector(*args)

    def tearDown(self):
        self.driver.quit()
```

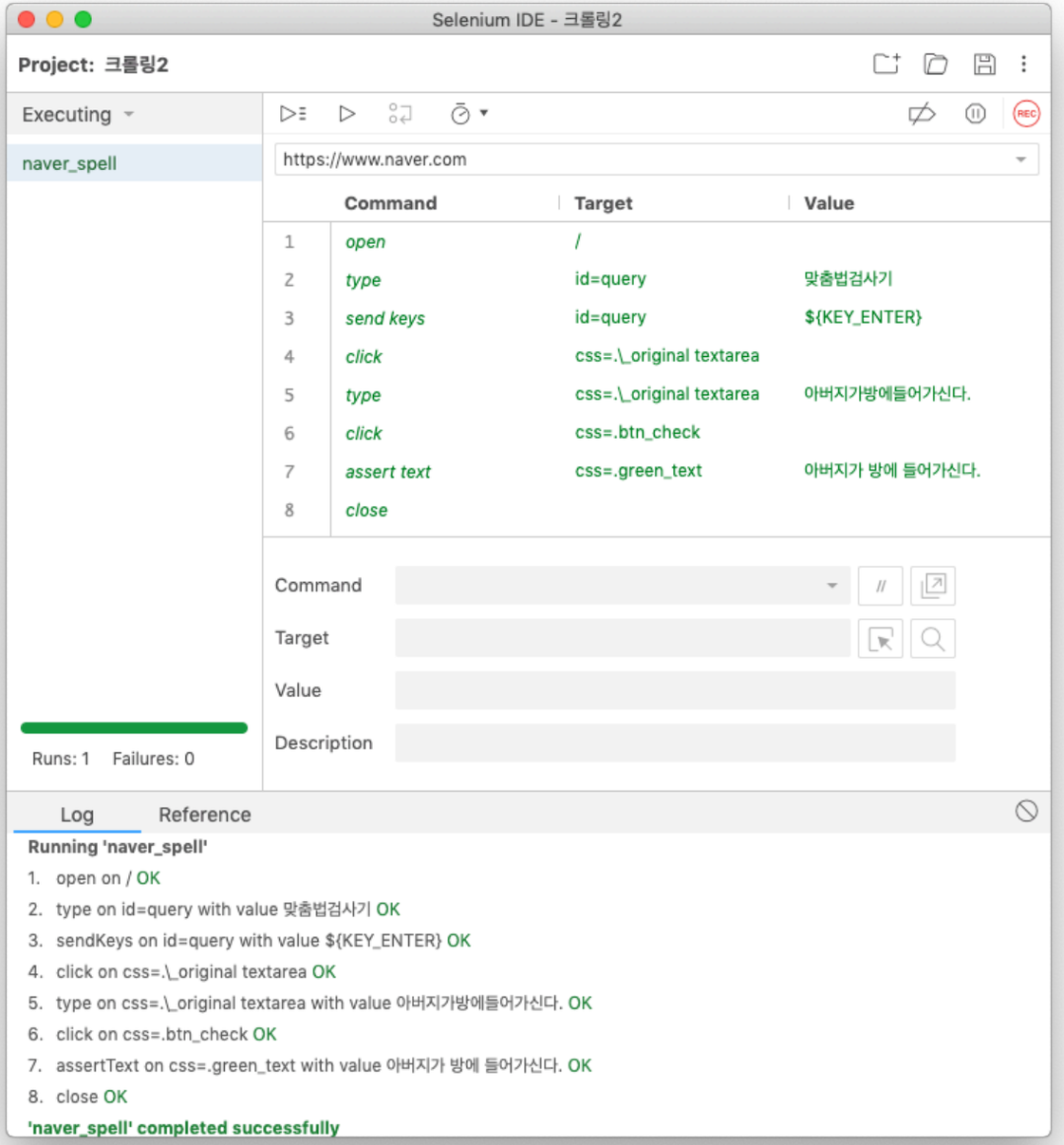
다음 페이지에 계속 ...

Step 2 : Pyton/unittest 활용 (2)

이전 페이지에 이어 ...

```
    def test_spelling(self):
        self.driver.get("https://www.naver.com")
        self.get_by_id("query").send_keys("맞춤법검사기" + Keys.ENTER)

        textarea = self.get_by_css('._original textarea')
        textarea.clear()
        textarea.send_keys("아버지가방에들어가신다.")

        btn = self.get_by_css('.btn_check')
        btn.click()

        # 안정적인 체크를 위해, 딜레이를 늘리거나 로직을 개선할 필요가 있다.
        time.sleep(1)

        ele = self.get_by_css('._result_text')

        self.assertEqual(ele.text, '아버지가 방에 들어가신다.')


if __name__ == '__main__':
    unittest.main()
```

Step 3: Selenium IDE 활용



selenium-side-runnder

(명령행을 통한 *.side 파일 수행)

selenium-side-runner

*.side, 커맨드 실행 유틸리티 (nodejs)

- nodejs Active LTS 버전 (2019.04 기준 10.x) 설치 → npm install --global selenium-side-runner

실행 예시

- selenium-side-runner proj01.side proj02.side *.side
- selenium-side-runner --server http://localhost:4444/wd/hub
- selenium-side-runner --base-url https://www.seleniumhq.org
- selenium-side-runner --capabilities "browserName=chrome platform=MAC"
- selenium-side-runner --capabilities "chromeOptions.args=[disable-infobars, headless]"

https://www.npmjs.com/package/selenium-side-runner

https://www.seleniumhq.org/selenium-ide/docs/en/introduction/command-line-runner/

브라우저 Drivers

디폴트 파일명으로 환경변수 PATH 상에 두거나

아래 npm 명령으로 설치도 가능.

- npm install --global chromedriver
  - iedriver, edgedriver, geckodriver

앞선 샘플 코드를 실행할 경우

```
$ selenium-side-runner naver_spell.side
info:     Running naver_spell.side
 PASS  ./DefaultSuite.test.js (6.492s)
  Default Suite
    ✓ naver_spell (4179ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        6.739s
Ran all test suites.
```

실행 옵션 (--help를 통해 확인)

- -s, --server : WebDriver Remote Server 주소
  - -s http://localhost:4444/wd/hub
- -c, --capabilities : WebDriver Capabilities 설정
  - -c "browserName='internet explorer' version='11.0' platform='Windows 8.1'"
  - -c "chromeOptions.args=[disable-infobars, headless]"
- -p, --params : 일반 옵션
- -f, --filter : 매칭되는 이름의 Suites만 실행 (정규표현식 지원)
- -w, --max-workers : 최대 수행 Workers 수 (디폴트: CPU 코어 개수)
- --base-url : 지정된 base URL을 재지정 (로컬개발환경, 테스트환경, 스테이징환경, 프로덕션 환경별로 수행)
- --timeout : Element 찾기에서의 타임아웃. 밀리세컨드 단위. 디폴트 15000 (15초)
- --config, --config-file, --configuration-file [파일경로] : YAML 포맷의 설정파일 경로, 디폴트 .side.yml
- --output-directory [디렉토리경로] : 테스트 수행 결과 파일을 생성할 디렉토리 경로 (JSON 포맷)
- --force : 프로젝트 버전을 무시하고, 프로젝트를 수행
- --debug : 디버그 로그 출력
- --proxy-type [type], --proxy-options [list] : 프록시 지원

.side.yml

```
capabilities:
    browserName: "internet explorer"
    version: "11.0"
    platform: "Windows"
baseUrl: "https://www.seleniumhq.org"
server: "http://localhost:4444/wd/hub"
```

다양한 Selenium IDE 제품군들 (2018년 기준)

	Selenium IDE	Kantu	Katalon	Sideex	iMacros
명령 녹화	O	O	O	O	-
Chrome 지원	O	O	O	O	O
Firefox 지원	O	O	O	O	O
오픈소스 여부	O	O	-	O	-
제어구조 지원 (조건/반복)	O	O	O	-	-
코드로 내보내기	△	-	O	-	-
웹 스크래핑	△	O	△	△	O
전체 스크린샷 지원	-	O	-	-	-
AI기반 시각적 테스팅	-	O	-	-	-
데스크탑 자동화	-	O	-	-	-
파일 다운로드 지원	△	O	△	△	O
파일 업로드 지원	-	O	-	-	O
	Link	Link	Link	Link	Link

https://a9t9.com/blog/selenium-ide-2018/

인생은 짧습니다.
파이썬/장고를 쓰세요.

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

- Ask Company