

## 4.1.5 检测应对

因为用 Selenium 做爬虫可以绕过 Js 的加密校验，所以很多网站都通过 Js 对 webdriver 进行检测，防止用自动化来做爬虫。

曾经最常见的检测是根据 navigator.webdriver 属性是否为 True 来判断浏览器真假，如果是 Google 浏览器 79 版本以前的，可以启用开发者模式来规避这个检测，方法如下：

```
from selenium.webdriver import Chrome
from selenium.webdriver import ChromeOptions

option = ChromeOptions()
option.add_experimental_option('excludeSwitches', ['enable-automation'])
driver = Chrome(options=option, executable_path="")
```

但是在 79 版本之后，chrome 修复了开发者模式下 navigator.webdriver 未定义的问题，在 79 版本之后的应对检测需要覆盖掉 webdriver 为 True 的属性，之前笔者一直采用下面的方法。

```
from selenium import webdriver
import time
from selenium.webdriver import ChromeOptions

option = ChromeOptions()
driver = webdriver.Chrome(executable_path="", options=option)
driver.execute_cdp_cmd("Page.addScriptToEvaluateOnNewDocument", {
    "source": """
        Object.defineProperty(navigator, 'webdriver', {
            get: () => undefined
        })
    """ })
```

但是近期又发现，88 版本的 Google 用这个方法失效了，不过笔者找到了新的解决方法，通过配置让 Chrome 隐藏掉 webdriver 的一些特征。如图 4-3 所示。

```
chrome_options = webdriver.ChromeOptions()
chrome_options.add_experimental_option('useAutomationExtension', False)
chrome_options.add_argument("disable-blink-features")
chrome_options.add_argument("disable-blink-features=AutomationControlled")
driver = webdriver.Chrome(executable_path="", chrome_options = chrome_options)
driver.get("https://bot.sannysoft.com/")
```

除了 navigator.webdriver 以外，还可能检测其他属性，比如 languages、length、user-agent 等，有时候分析起来着实麻烦。那么当确定 chrome 驱动被检测到，上面的方法不能解决问题，就更换浏览器，比如 Firefox、Edge，实在不行就换 IE 浏览器。不过 Selenium 调用 IE 也需要下载对应的 IEDriverServer，打开下载链接后，根据 Selenium 版本选择对应的驱动目录即可。

IEDriverServer 下载地址：<http://selenium-release.storage.googleapis.com/inDex.html>

IE7 或者更高的版本需要关闭保护模式，方法如下：打开 Internet 选项，选择“安全”选项，关闭 4 个区域的保护模式。关闭高级选项中的增强保护（默认关闭），如图 4-4 所示。