AUTOMATE THE PLANET

Most Complete Selenium WebDriver C# Cheat Sheet

Initialize

```
// NuGet: Selenium.WebDriver.ChromeDriver
using OpenQA.Selenium.Chrome;
IWebDriver driver = new ChromeDriver();
// NuGet: Selenium.Mozilla.Firefox.Webdriver
using OpenQA.Selenium.Firefox;
IWebDriver driver = new FirefoxDriver();
// NuGet: Selenium.WebDriver.IEDriver
using OpenQA.Selenium.IE;
IWebDriver driver = new InternetExplorerDriver();
// NuGet: Selenium.WebDriver.EdgeDriver
using OpenQA.Selenium.Edge;
IWebDriver driver = new EdgeDriver();
```

Locators

```
this.driver.FindElement(By.ClassName("className"));
this.driver.FindElement(By.CssSelector("css"));
this.driver.FindElement(By.Id("id"));
this.driver.FindElement(By.LinkText("text"));
this.driver.FindElement(By.Name("name"));
this.driver.FindElement(By.PartialLinkText("pText"));
this.driver.FindElement(By.TagName("input"));
this.driver.FindElement(By.XPath("//*[@id='editor']"));
// Find multiple elements
IReadOnlyCollection<IWebElement> anchors =
this.driver.FindElements(By.TagName("a"));
// Search for an element inside another
var div = this.driver.FindElement(By.TagName("div"))
.FindElement(By.TagName("a"));
```

Basic Browser Operations

```
// Navigate to a page
this.driver.Navigate().GoToUrl(@"http://google.com");
// Get the title of the page
string title = this.driver.Title;
// Get the current URL
string url = this.driver.Url;
// Get the current page HTML source
string html = this.driver.PageSource;
```

Basic Elements Operations

```
IWebElement element = driver.FindElement(By.Id("id"));
element.Click();
element.SendKeys("someText");
element.Clear();
element.Submit();
string innerText = element.Text;
bool isEnabled = element.Enabled;
bool isDisplayed = element.Displayed;
bool isSelected = element.Selected;
IWebElement element = driver.FindElement(By.Id("id"));
SelectElement select = new SelectElement(element);
select.SelectByIndex(1);
select.SelectByText("Ford");
select.SelectByValue("ford");
select.DeselectAll();
select.DeselectByIndex(1);
select.DeselectByText("Ford");
select.DeselectByValue("ford");
IWebElement selectedOption = select.SelectedOption;
IList<IWebElement> allSelected = select.AllSelectedOptions;
bool isMultipleSelect = select.IsMultiple;
```

Advanced Element Operations

```
// Drag and Drop
IWebElement element = driver.FindElement(
By.XPath("//*[@id='project']/p[1]/div/div[2]"));
Actions move = new Actions(driver);
move.DragAndDropToOffset(element, 30, 0).Perform();
// How to check if an element is visible
Assert.IsTrue(driver.FindElement(
By.XPath("//*[@id='tve_editor']/div")).Displayed);
// Upload a file
IWebElement element =
driver.FindElement(By.Id("RadUpload1file0"));
String filePath = @"D:\WebDriver.Series.Tests\\WebDriver.xml";
element.SendKeys(filePath);
// Scroll focus to control
IWebElement link =
driver.FindElement(By.PartialLinkText("Previous post"));
string js = string.Format("window.scroll(0, {0});",
link.Location.Y);
((IJavaScriptExecutor)driver).ExecuteScript(js);
link.Click();
// Taking an element screenshot
IWebElement element =
driver.FindElement(By.XPath("//*[@id='tve_editor']/div"));
var cropArea = new Rectangle(element.Location, element.Size);
var bitmap = bmpScreen.Clone(cropArea, bmpScreen.PixelFormat);
bitmap.Save(fileName);
// Focus on a control
IWebElement link =
driver.FindElement(By.PartialLinkText("Previous post"));
// Wait for visibility of an element
WebDriverWait wait = new WebDriverWait(driver,
TimeSpan.FromSeconds(30));
wait.Until(ExpectedConditions.VisibilityOfAllElementsLocatedBy(
By.XPath("//*[@id='tve_editor']/div[2]/div[2]/div/div")));
```

Advanced Browser Operations

```
// Handle JavaScript pop-ups
IAlert a = driver.SwitchTo().Alert();
a.Accept();
a.Dismiss();
// Switch between browser windows or tabs
ReadOnlyCollection<string> windowHandles = driver.WindowHandles;
string firstTab = windowHandles.First();
string lastTab = windowHandles.Last();
driver.SwitchTo().Window(lastTab);
// Navigation history
this.driver.Navigate().Back();
this.driver.Navigate().Refresh();
this.driver.Navigate().Forward();
// Option 1.
link.SendKeys(string.Empty);
// Option 2.
((IJavaScriptExecutor)driver).ExecuteScript("arguments[0].focus();",
link);
// Maximize window
this.driver.Manage().Window.Maximize();
// Add a new cookie
Cookie cookie = new OpenQA.Selenium.Cookie("key", "value");
this.driver.Manage().Cookies.AddCookie(cookie);
// Get all cookies
var cookies = this.driver.Manage().Cookies.AllCookies;
// Delete a cookie by name
this.driver.Manage().Cookies.DeleteCookieNamed("CookieName");
// Delete all cookies
this.driver.Manage().Cookies.DeleteAllCookies();
// Taking a full-screen screenshot
Screenshot screenshot = ((ITakesScreenshot)driver).GetScreenshot();
screenshot.SaveAsFile(@"pathToImage", ImageFormat.Png);
// Wait until a page is fully loaded via JavaScript
WebDriverWait wait = new WebDriverWait(this.driver,
TimeSpan.FromSeconds(30));
wait.Until((x) =>
{
    return ((IJavaScriptExecutor)this.driver).ExecuteScript(
    "return document.readyState").Equals("complete");
});
// Switch to frames
this.driver.SwitchTo().Frame(1);
this.driver.SwitchTo().Frame("frameName");
IWebElement element = this.driver.FindElement(By.Id("id"));
this.driver.SwitchTo().Frame(element);

// Switch to the default document
this.driver.SwitchTo().DefaultContent();
```

Advanced Browser Confgurations

```
// Use a specific Firefox profile
FirefoxProfileManager profileManager = new FirefoxProfileManager();
FirefoxProfile profile = profileManager.GetProfile("HARDDISKUSER");
IWebDriver driver = new FirefoxDriver(profile);
// Set a HTTP proxy Firefox
FirefoxProfile firefoxProfile = new FirefoxProfile();
firefoxProfile.SetPreference("network.proxy.type", 1);
firefoxProfile.SetPreference("network.proxy.http", "myproxy.com");
firefoxProfile.SetPreference("network.proxy.http_port", 3239);
IWebDriver driver = new FirefoxDriver(firefoxProfile);
// Set a HTTP proxy Chrome
ChromeOptions options = new ChromeOptions();
var proxy = new Proxy();
proxy.Kind = ProxyKind.Manual;
proxy.IsAutoDetect = false;
proxy.HttpProxy = "127.0.0.1:3239";
proxy.SslProxy = "127.0.0.1:3239";
options.Proxy = proxy;
options.AddArgument("ignore-certificate-errors");
IWebDriver driver = new ChromeDriver(options);
// Accept all certificates Firefox
FirefoxProfile firefoxProfile = new FirefoxProfile();
firefoxProfile.AcceptUntrustedCertificates = true;
firefoxProfile.AssumeUntrustedCertificateIssuer = false;
IWebDriver driver = new FirefoxDriver(firefoxProfile);
// Accept all certificates Chrome
DesiredCapabilities capability = DesiredCapabilities.Chrome();
Environment.SetEnvironmentVariable("webdriver.chrome.driver",
"C:\\PathToChromeDriver.exe");
capability.SetCapability(CapabilityType.AcceptSslCertificates, true);
IWebDriver driver = new RemoteWebDriver(capability);
// Set Chrome options.
ChromeOptions options = new ChromeOptions();
DesiredCapabilities dc = DesiredCapabilities.Chrome();
dc.SetCapability(ChromeOptions.Capability, options);
IWebDriver driver = new RemoteWebDriver(dc);
// Turn off the JavaScript Firefox
FirefoxProfileManager profileManager = new FirefoxProfileManager();
FirefoxProfile profile = profileManager.GetProfile("HARDDISKUSER");
profile.SetPreference("javascript.enabled", false);
IWebDriver driver = new FirefoxDriver(profile);
// Set the default page load timeout
driver.Manage().Timeouts().SetPageLoadTimeout(new TimeSpan(10));
// Start Firefox with plugins
FirefoxProfile profile = new FirefoxProfile();
profile.AddExtension(@"C:\extensionsLocation\extension.xpi");
IWebDriver driver = new FirefoxDriver(profile);
// Start Chrome with an unpacked extension
ChromeOptions options = new ChromeOptions();
options.AddArguments("load-extension=/pathTo/extension");
DesiredCapabilities capabilities = new DesiredCapabilities();
capabilities.SetCapability(ChromeOptions.Capability, options);
DesiredCapabilities dc = DesiredCapabilities.Chrome();
dc.SetCapability(ChromeOptions.Capability, options);
IWebDriver driver = new RemoteWebDriver(dc);
// Start Chrome with a packed extension
ChromeOptions options = new ChromeOptions();
options.AddExtension(Path.GetFullPath("localpathto/extension.crx"));
DesiredCapabilities capabilities = new DesiredCapabilities();
capabilities.SetCapability(ChromeOptions.Capability, options);
DesiredCapabilities dc = DesiredCapabilities.Chrome();
dc.SetCapability(ChromeOptions.Capability, options);
IWebDriver driver = new RemoteWebDriver(dc);
// Change the default files' save location
String downloadFolderPath = @"c:\temp";
FirefoxProfile profile = new FirefoxProfile();
profile.SetPreference("browser.download.folderList", 2);
profile.SetPreference("browser.download.dir", downloadFolderPath);
profile.SetPreference("browser.download.manager.alertOnEXEOpen",
false);
profile.SetPreference("browser.helperApps.neverAsk.saveToDisk",
"application/msword, application/binary, application/ris, text/csv,
image/png, application/pdf, text/html, text/plain, application/zip,
application/x-zip, application/x-zip-compressed,
application/download, application/octet-stream");
this.driver = new FirefoxDriver(profile);
```