|  |  |
| --- | --- |
|  | |
| DevOps case study  Case 5: Web scraper | |
| **APP/AI 01**  Grevendonk Tim | 2de fase IT Factory |
|  |
| Academiejaar 2021-2022  Campus Geel, Kleinhoefstraat 4, BE-2440 Geel |



table of contents

[table of contents 2](#_Toc88159927)

[1 Concept 3](#_Toc88159928)

[2 Idea & introduction 4](#_Toc88159929)

[3 Report 5](#_Toc88159930)

[3.1 Installation steps 5](#_Toc88159931)

[4 Conclusion 8](#_Toc88159932)

[5 Citation 9](#_Toc88159933)

# Concept

Through a short intro to scraping of dynamic webpages with Selenium [[2]](#_bronvermelding).  
I came to the conclusion that this would be an interesting project for my case study.

I play a tabletop game called Warhammer 40K that often goes through rule updates, changes, and tweaks. This often happens through FAQ’s, blog posts or general updates.

Due to not having a lot of free time left anymore it is hard to stay up to date with these rules (for me and my opponents their army, and the core rules), Scraping the web for any rule updates, new tactics or combos without having to do forensic research myself would be an delightful situation.

As I understand Selenium to this point it can be done via keywords, search terms and videos. This might a nice (but hard) opportunity to learn C#, create a nice project and have a project that could benefit me after completion!

# Idea & introduction

## Assignment description

Final solution must be uploaded **before** Sunday 19 dec 24:00

Use Selenium together with C# to scrape web data with automated and emulated user interactions. Build a console-based scraping tool.

-Maka a minimum of 3 scraping options:

1. Scrape a database (links, title, uploader, views) of 5 most recent YouTube uploads taken from a search term inputted by a user of the scraping tool.
2. Scrape data from the jobsite “be.indeed.com” that will return the title, company, location, and link of jobs of the last 3 days. This also from a search term.
3. 1 own chosen site & data to scrape with a search term.

-Data must be written with Micro ORM or to a CSV file.  
-the app is in a GitHub Actions CI/CD pipeline with an downloadable .exe file.  
-highlight unique and creative solution or uses of the project!

-in PDF format description (this file).  
-link to public GitHub repository (with final code).  
-YouTube link to a video of the demo, along with a demonstration   
 video = 10+ min long and must be **accessible** to other users.

# Report

## Installation steps

Installation of visual studio (2019 community edition) [[1]](#_bronvermelding)

Selenium Browser drivers for chrome from ChromeDriver [[3]](#_bronvermelding)   
Taken use from Chromedriver 96.0.4664.45, Chromedriver\_Win32

C# Packages/libraries: Selenium.WebDriver, NUnit, NUnit3TestAdapter, Microsoft.NET.Test.Sdk. [[4]](#_bronvermelding)

1. In IDE tools => nuGet package Manager => Package Manager Console.
2. In Package Manager Console run:
   * Install-Package Selenium.WebDriver
   * Install-Package NUnit
   * Install-Package NUnit3TestAdapter
   * Install-Package Microsoft.NET.Test.Sdk
   * Install-Package Selenium.WebDriver.ChromeDriver
   * Get-package (to confirm all installations)

## Test build 1:

Open a basic empty webpage with the code

In solution-explorer right click > add > new solution folder  
in the folder copy the downloaded “chromedriver.exe” [[3]](#_Citation)

Richt click the chromedriver.exe > copy full path. And paste in code [[5]](#_bronvermelding)

|  |
| --- |
| using OpenQA.Selenium; |
| using OpenQA.Selenium.Chrome; |
| using OpenQA.Selenium.Support.UI; |
| //using SeleniumExtras.WaitHelpers; |
| using System; |
| using System.IO; |
| using System.Reflection; |
|  |
| using Microsoft.VisualStudio.TestTools.UnitTesting; |
| using System; |
|  |
| namespace webscraper |
| { |
| [TestClass] |
| public class webscraper |
| { |
| [TestMethod] |
| public static void Main(string[] args) |
| { |
| ChromeDriver driver = new ChromeDriver("C:\\paste\\path\\here"); |
| } |
| } |
| } |

A screenshot of a computer

Description automatically generated with medium confidence

Right click solution explorer > Properties > in popup select “application” (sidebar)  
in box “Output type:” > Console application [[6]](#_bronvermelding)

## Basic youtube setup:

using OpenQA.Selenium;

using OpenQA.Selenium.Chrome;

using OpenQA.Selenium.Support.UI;

using System;

using System.IO;

using System.Reflection;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

namespace webscraper

{

public class webscraper

{

public static void Main(string[] args)

{

// Use the chromedriver.exe in the Drivers folder.

ChromeDriver driver = new ChromeDriver(@"../../" + "/Drivers/");

driver.Manage().Window.Maximize();

// Go to the website (YouTube in this case)

driver.Navigate().GoToUrl("https://www.youtube.com/");

// Find and Click "accept Terms" button.

driver.FindElement(By.XPath(  
 "//\*[@id='content']/div[2]/div[5]/div[2]/ytd-button-renderer[2]"))  
 .Click();

// sleep/wait for 1 second to let the items/page load,   
 // Or else the input field is not fillable.

System.Threading.Thread.Sleep(1000);

// Find the Youtube input field.

IWebElement searchBarInput = driver.FindElement(By.Name("search\_query"));

// Enter value in the field and submit it.

searchBarInput.SendKeys("rick roll");

searchBarInput.Submit();

}

}

}

## Setting up VMC:

Graphical user interface, text, chat or text message

Description automatically generated

I work similar to the way the VMC setup has been done in our course[[7]](#_bronvermelding).

Make a folder with “Actions” that will perform actions like browsing to a site.

Make a folder with “views” that will print the prompt info per page.



Import them at the top of every class.

  
Text

Description automatically generated

In the main program page (webscraper.cs) call other classes to perform actions or print view content to the Console terminal.

# Conclusion

# Citation

[1] Thomas More. (2021, 25 October). Intro to C#. Project Mercury. . Referenced on 18 November 2021,   
From <https://mystifying-khorana-6156c2.netlify.app/programming/csharp/>

[2] Sheth, H. (2021, 17 November). Scraping Dynamic Web Pages Using Selenium and C#. LambdaTest. Referenced on November 2021,   
From <https://www.lambdatest.com/blog/scraping-dynamic-web-pages/>

[3] Chromium. (z.d.). ChromeDriver - WebDriver for Chrome - Downloads. Chromium.Org. . Referenced on 18 November 2021,   
From <https://chromedriver.chromium.org/downloads>

[4] Sheth, H. (2021, november 17). Scraping Dynamic Web Pages Using Selenium and C#. LambdaTest. Referenced on 18 November 2021,   
From <https://www.lambdatest.com/blog/scraping-dynamic-web-pages/>

[5] Repala, N. (2020, 2 april). Selenium C# – Adding Chrome driver to execute scripts on Chrome Browser - QAFox. QAFox | The Easiest Tutorial Site on Software Testing. Referenced on 25 November 2021,   
From <https://www.qafox.com/selenium-c-sharp-adding-chrome-driver/>

[6] H.H. (2010, 29 juli). “A project with an Output type of Class Library cannot be started directly”. Stack Overflow. Referenced on 25 November 2021,   
From <https://stackoverflow.com/questions/3363106/a-project-with-an-output-type-of-class-library-cannot-be-started-directly>

[7] Verboven, M. V. (2021, 9 november). Video - Dutch - From StockLister-Base to Stocklister-SemiComplete. thomasmore instructure. Referenced on 15 December 2021, From <https://thomasmore.instructure.com/courses/19824/pages/video-dutch-from-stocklister-base-to-stocklister-semicomplete?module_item_id=1198976>