Afbeelding met tekst, illustratie

Automatisch gegenereerde beschrijving **Campus Geel**

**DevOps & Security**

Case study

**Student: Ruben Heurckmans**

**R-number: R0831546**

**Academic year 2022-2023**

# Inhoud

[Inhoud 2](#_Toc121917784)

[Inleiding 3](#_Toc121917785)

[Link naar github 3](#_Toc121917786)

[Demo 3](#_Toc121917787)

[Video 3](#_Toc121917788)

[Uitleg 3](#_Toc121917789)

[Oplossing 5](#_Toc121917790)

[Aanmaken project 5](#_Toc121917791)

[Selenium toevoegen aan je project 6](#_Toc121917792)

[Code 7](#_Toc121917793)

[YouTube scraper 7](#_Toc121917794)

[Full code 7](#_Toc121917795)

[Code in detail 8](#_Toc121917796)

[IctJob scraper 10](#_Toc121917797)

[Full code 10](#_Toc121917798)

[Code in detail 11](#_Toc121917799)

[Twitch scraper 13](#_Toc121917800)

[Full code 13](#_Toc121917801)

[Code in detail 14](#_Toc121917802)

[Bronnen 16](#_Toc121917803)

[De basis 16](#_Toc121917804)

[Extra 16](#_Toc121917805)

# Inleiding

In deze case study geef ik weer hoe ik een webscraper heb gemaakt voor YouTube, Ictjob en Twitch.

De YouTube scraper geeft de 5 meest recent geüploade video’s op basis van een ingegeven zoekterm. De waarde die je krijgt zijn: de titel, het kanaal van wie de video is, het aantal weergaven en een link naar de video.

De Ictjob scraper geeft de 5 meest recent aangemaakte jobs op basis van keywords die je ingeeft.  
De waarde die je krijgt zijn: de titel, het bedrijf, de locatie, alle keywords van de job en een link naar de detailpagina.

De Twitch scraper geeft de 5 streams met de meeste kijkers van een ingegeven categorie.  
De waarde die je krijgt zijn: de titel, de taal, het kanaal en het aantal live kijkers.

# Link naar github

<https://github.com/Ruben-Heurckmans/DevopsCaseStudy>

# Demo

## Video

<https://youtu.be/X-gtbScq-Zk>

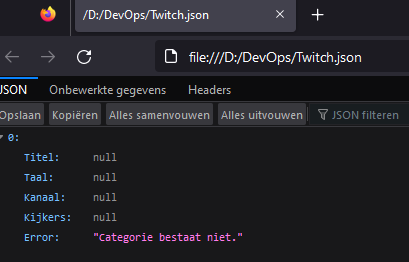
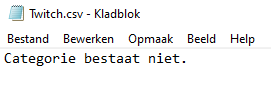
## Uitleg

Alle 3 de scrapers werken hetzelfde. Je start het programma. Deze gaat vragen voor een zoekterm in te vullen. Hierin vul je je zoekterm in. Vervolgens gaat de scraper naar de gepaste link aan de hand van de ingegeven zoekterm. Onderstaande voorbeelden zijn resultaten van de Twitch scraper.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Hierna gaan de programma’s na of er data te vinden is met de ingegeven zoekterm. Is dit er niet komt er in de console, csv en json te staan dat er niets te vinden is met de ingegeven zoekterm.

Als er wel data is komen er 5 video’s, vacatures of streams in je console, csv en json te staan.

Afbeelding met tekst

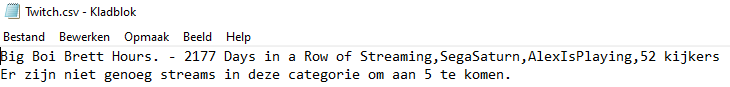
Automatisch gegenereerde beschrijving Afbeelding met tekst

Automatisch gegenereerde beschrijvingAfbeelding met tekst

Automatisch gegenereerde beschrijving

Afbeelding met tekst

Automatisch gegenereerde beschrijvingMoesten er minder dan 5 zoekresultaten zijn, dan geeft hij dit ook weer in de console, csv en json.

 Afbeelding met tekst, schermafbeelding, monitor, zwart

Automatisch gegenereerde beschrijving

# Oplossing

## Aanmaken project

Om te beginnen open je visual studio en maak je een Console App aan.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Daarna geef je je project een naam. Mijn projecten noemen: WebscraperYt, WebscraperTwitch en WebscraperIctjob.

Afbeelding met tekst

Automatisch gegenereerde beschrijvingVoor framework heb ik “.NET 5.0” genomen.

Daarna druk je op create.

## Selenium toevoegen aan je project

Als het project is aangemaakt moeten we onze scraping tool nog toevoegen aan ons project. Ik gebruik hier selenium voor. Om dit toe te voegen klik je rechts op “Solution (naam van je project)” en kies je voor “Manage NuGet Packages for Solution”.

Afbeelding met tekst, monitor, schermafbeelding, scherm

Automatisch gegenereerde beschrijving

In de package manager zoek je naar Selenium.WebDriver. De package manager scrolt zelf een beetje naar beneden dus om de juiste te zien moet je wat naar boven scrollen. Selecteer de package en check de checkboxes zoals in de afbeelding. De versie die ik gebruikte is 4.6.0. Druk daarna op install.

Afbeelding met tekst, monitor, schermafbeelding, binnen

Automatisch gegenereerde beschrijving

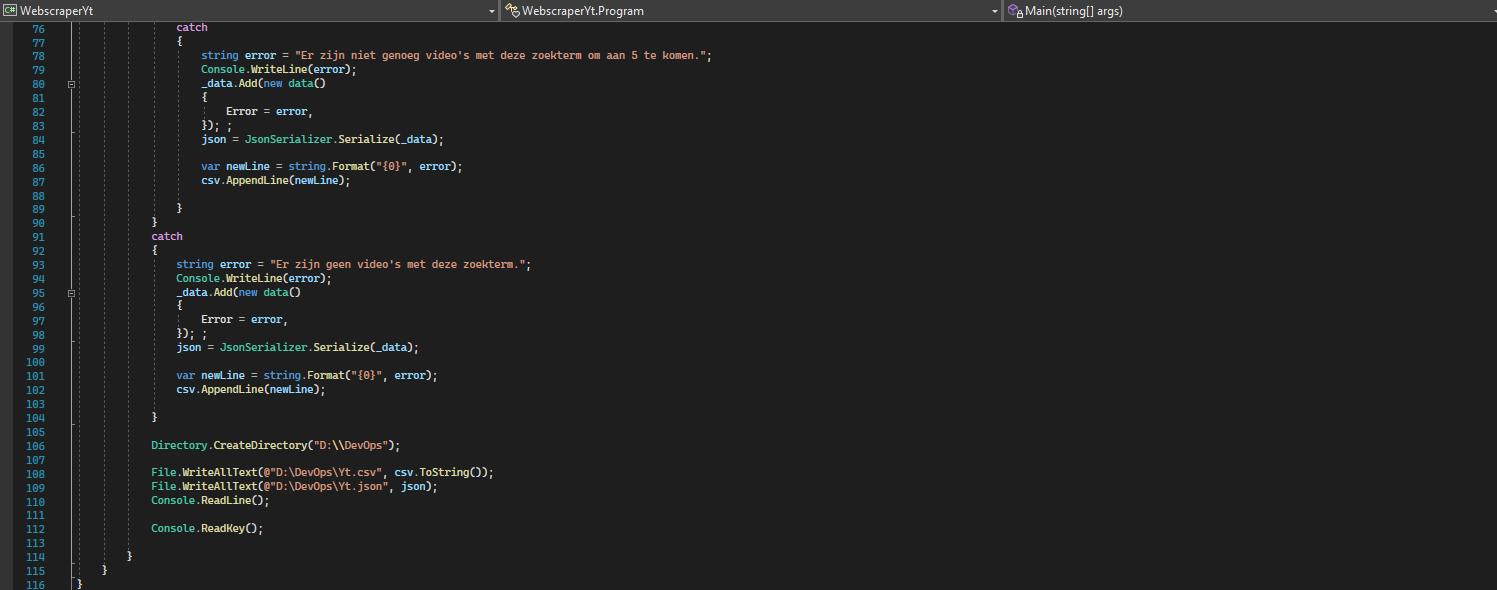
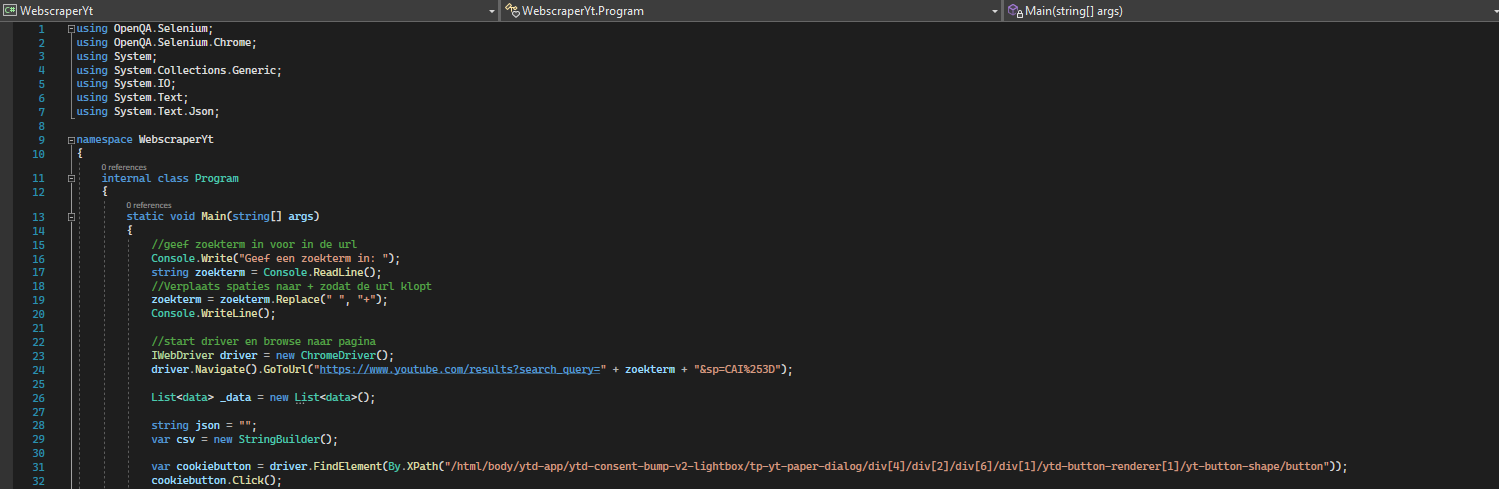
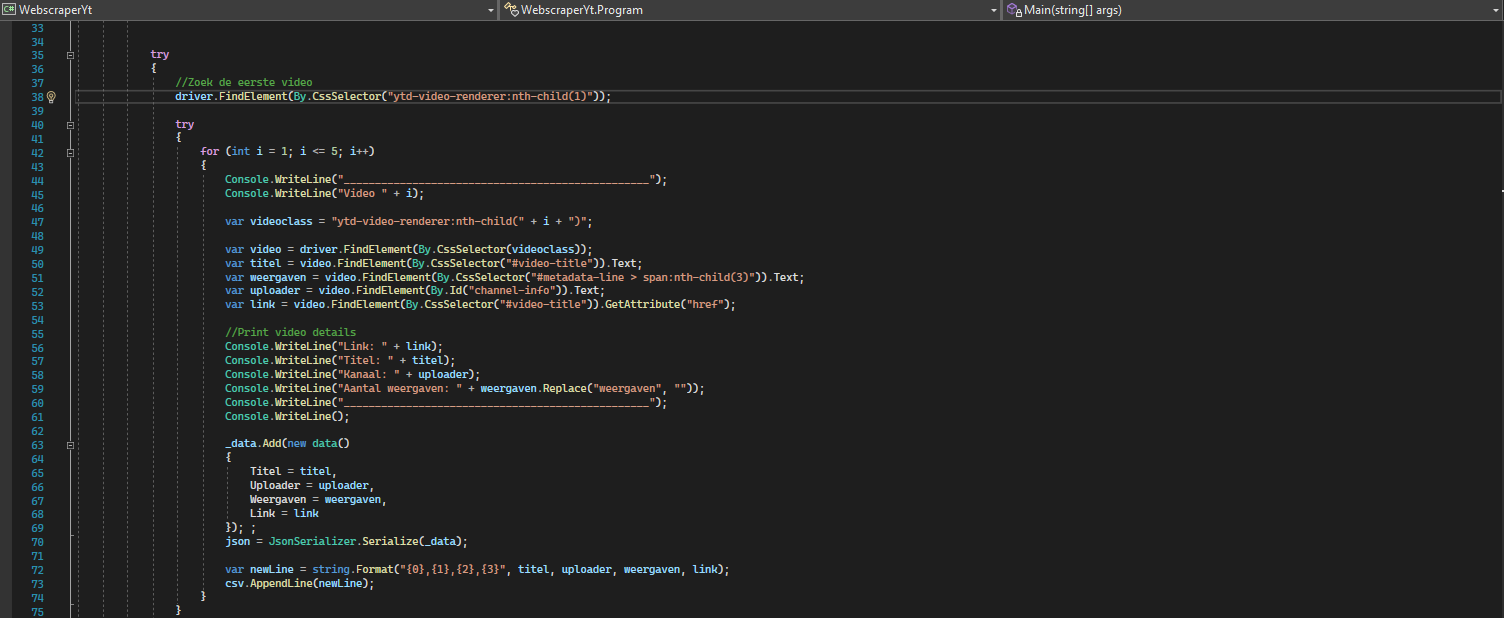
## Code

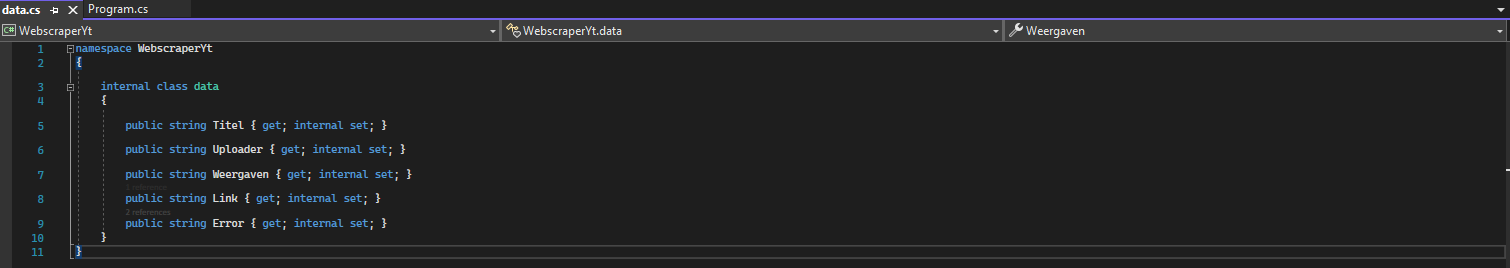
De drie projecten die gemaakt zijn voor deze taak lijken best hard op elkaar dus veel dingen gaan hetzelfde zijn bij de projecten.

## YouTube scraper

### Full code

Program.cs file



data.cs file

### Code in detail

Om te beginnen schrijven we de code om de zoekterm op te slaan als de variabele “zoekterm”.  
Dit doen we door een simpele “Console.Write()” met de vraag om een zoekterm in te vullen.  
Het lezen van de ingevulde variabele doen we met een “Console.ReadLine()” en dat slaan we op onder “string zoekterm”.

Aangezien spaties niet kunnen in een url vervangen websites dit door iets anders. Bij YouTube doen ze dit door de spaties te vervangen met een “+”, dus wij moeten dit ook doen. Dit doen we met de Replace() functie. Deze vervangt de waarde voor de komma met de waarde na de komma.

De Console.WriteLine() is voor een witruimte te laten tussen dit stuk en het volgende stuk in het resultaat.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Als volgende starten we de driver en gaan we naar de link. Zoals je ziet steken we de zoekterm in de url om naar de juiste pagina te gaan. De “&sp=CAI%253D” die ik op het einde toevoeg geeft aan dat we filteren op uploaddatum.

Verder maken we een list om later ons resultaat in json om te zetten en we maken de var csv aan.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Als we naar YouTube gaan met de url krijgen we een cookie pop-up op ons scherm. Deze gaan we eerst weg moeten klikken voor we de data kunnen opvragen.  
Afbeelding met tekst

Automatisch gegenereerde beschrijving

In onderstaande screenshot is de tweede “try” collapsed om het duidelijker te maken.  
In de code die je nu ziet gaan we kijken of er een video te vinden is. Als dit niet zo is zouden we normaal een error krijgen maar deze vangen we op met de “catch”. De variabele error wordt in de console geschreven en wordt ook opgeslagen in het Json en Csv bestand.

Als er wel een video gevonden is gaan we door naar de tweede “try”.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Deze “try” checkt of de for loop 5 keer kan worden gedaan. Als er een zoekresultaat is waar er bijvoorbeeld maar 3 video’s van te vinden zijn gaat hij bij de vierde loop een error willen geven. Dit vangen we weer op met de catch. In deze catch wordt de variabele “error” weer in de Json en Csv file gestoken met de tekst “Er zijn niet genoeg video’s met deze zoekterm om aan 5 te komen.”.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

De for loop zelf is vrij simpel. We gaan zoeken naar de video waarvan de CssSelector de waarde heeft van videoclass. En deze slaan we op onder de variabele “video”.  
Wat we dan zoeken is hetgeen dat weergegeven is in onderstaande screenshot.  
Afbeelding met tekst

Automatisch gegenereerde beschrijving

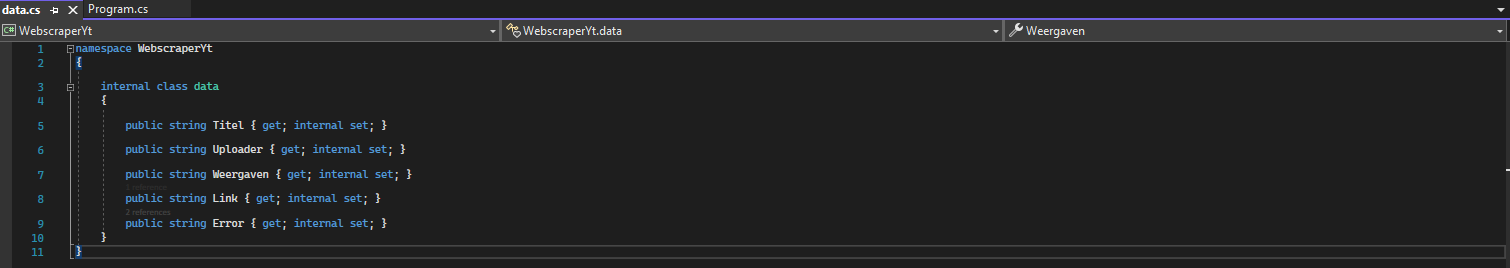
Vanuit hier gaan we zoeken naar de details van de video. En steken we deze in gepaste variabelen om ze daarna allemaal te printen in de console en in de Json en Csv files te steken.

Op het einde van de code vind je dit terug. Wat we hier doen is, dat we een directory maken op de D schijf met de naam “DevOps” en dat we daarna de Json en Csv file daarin gaan opslaan.

De Console.ReadLine op het einde plaatsen we zodat de console open blijft staan en niet direct sluit.

Afbeelding met tekst

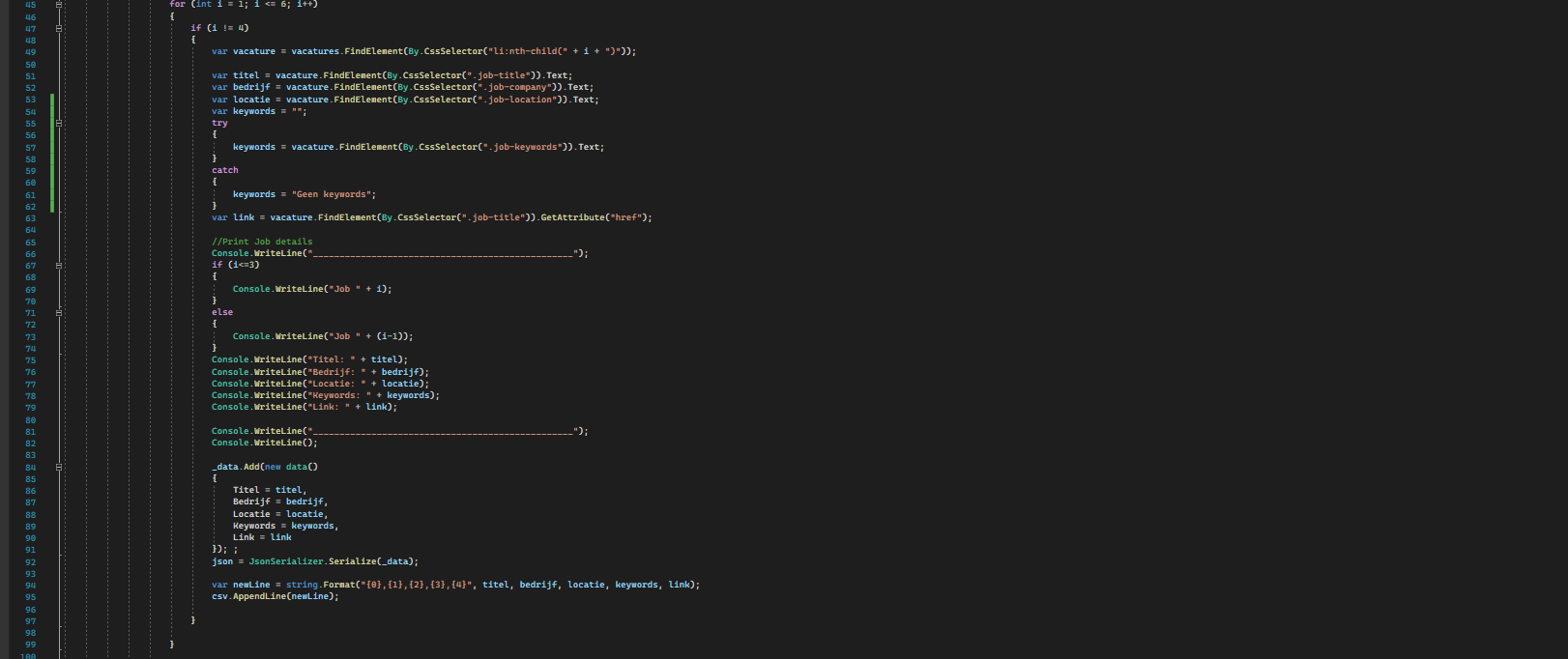
Automatisch gegenereerde beschrijving

De code van de “data.cs” is vrij vanzelfsprekend, het enige dat we daarin zetten is het aanmaken van enkele variabelen die nodig zijn voor in de Json te steken.

## IctJob scraper

### Full code

Program.cs file



Afbeelding met tekst, monitor, schermafbeelding, scherm

Automatisch gegenereerde beschrijving

data.cs file

### Code in detail

Het begin van de code is perfect hetzelfde als de YouTube scraper.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Als volgende starten we de driver en gaan we naar de link. Zoals je ziet steken we de zoekterm in de url om naar de juiste pagina te gaan.

Daarna zoeken we het element met de CssSelector “#sort-by-date” en klikken we erop om te filteren op basis van datus.

Verder maken we weer de list aan om later ons resultaat in json om te zetten en we maken de var csv aan.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

In onderstaande screenshot is de tweede “try” collapsed om het duidelijker te maken.  
In de code die je nu ziet gaan we kijken of er jobs te vinden zijn. Als dit niet zo is zouden we normaal een error krijgen maar deze vangen we op met de “catch”. De variabele error wordt in de console geschreven en wordt ook opgeslagen in het Json en Csv bestand.

Als er wel een job gevonden is gaan we door naar de tweede “try”.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Deze “try” checkt of de for loop 6 keer kan worden gedaan. Als er een zoekresultaat is waar er bijvoorbeeld maar 3 jobs van te vinden zijn gaat hij bij de vierde loop een error willen geven. Dit vangen we weer op met de catch. In deze catch wordt de variabele “error” weer in de Json en Csv file gestoken met de tekst “Er zijn niet genoeg jobs met deze keyword(s) om aan 5 te komen.”.

De rede dat we nu de loop 6 keer herhalen is omdat het vierde object een call to action is om zelf een job alert aan te maken. Deze slaan we over en voor in de console doen we vanaf “i>3”, -1 voor de i.

Afbeelding met tekst

Automatisch gegenereerde beschrijvingAfbeelding met tekst, monitor, schermafbeelding, scherm

Automatisch gegenereerde beschrijvingAfbeelding met tekst

Automatisch gegenereerde beschrijving

De for loop zelf is vrij simpel. We zoeken naar een vacature en die slaan we in zijn geheel op als “vacature. Wat we dan zoeken is hetgeen dat weergegeven is in onderstaande screenshot.  
Afbeelding met tekst

Automatisch gegenereerde beschrijving

Vanuit hier gaan we zoeken naar de details van de vacature. En steken we deze in gepaste variabelen om ze daarna allemaal te printen in de console en in de Json en Csv files te steken.

Verder hebben we een extra try en catch bij keywords. Dit doen we omdat niet elke joblisting keywords heeft. Als we dit niet zouden doen krijgen we een error en zouden we de melding krijgen dat er niet genoeg jobs zijn.

Op het einde van de code vind je dat ik op dezelfde manier terug de data naar de json en Csv file schrijf in de map DevOps. Alleen dan nu onder de namen “Yt.csv” en “Yt.json”

De Console.ReadLine op het einde plaatsen we zodat de console open blijft staan en niet direct sluit.

Afbeelding met tekst, monitor, schermafbeelding, scherm

Automatisch gegenereerde beschrijvingAfbeelding met tekst

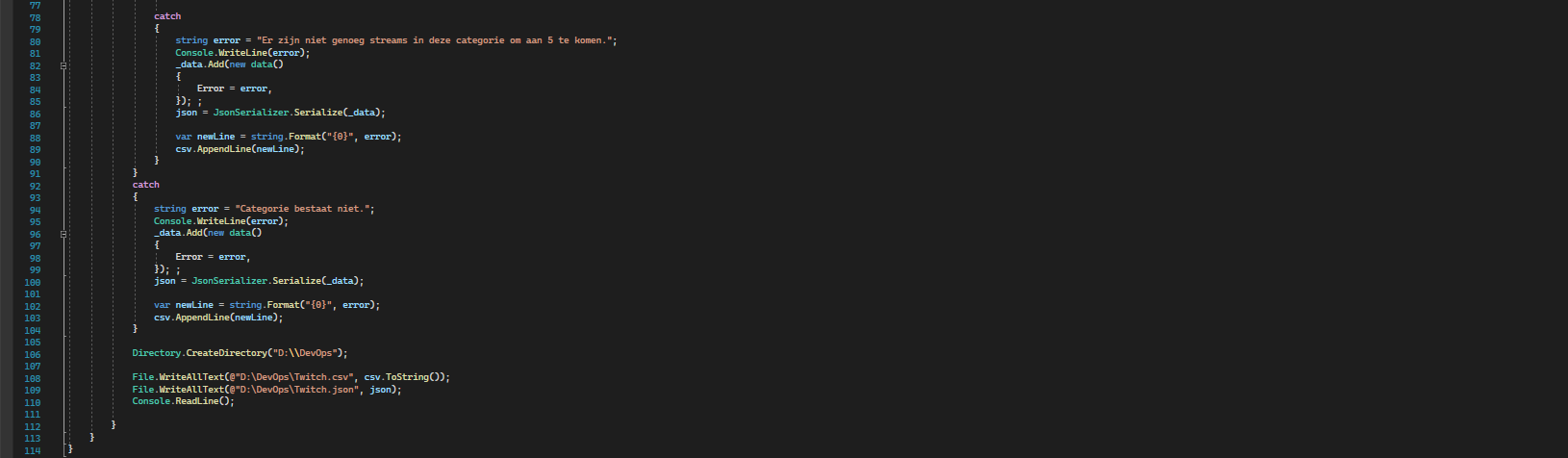
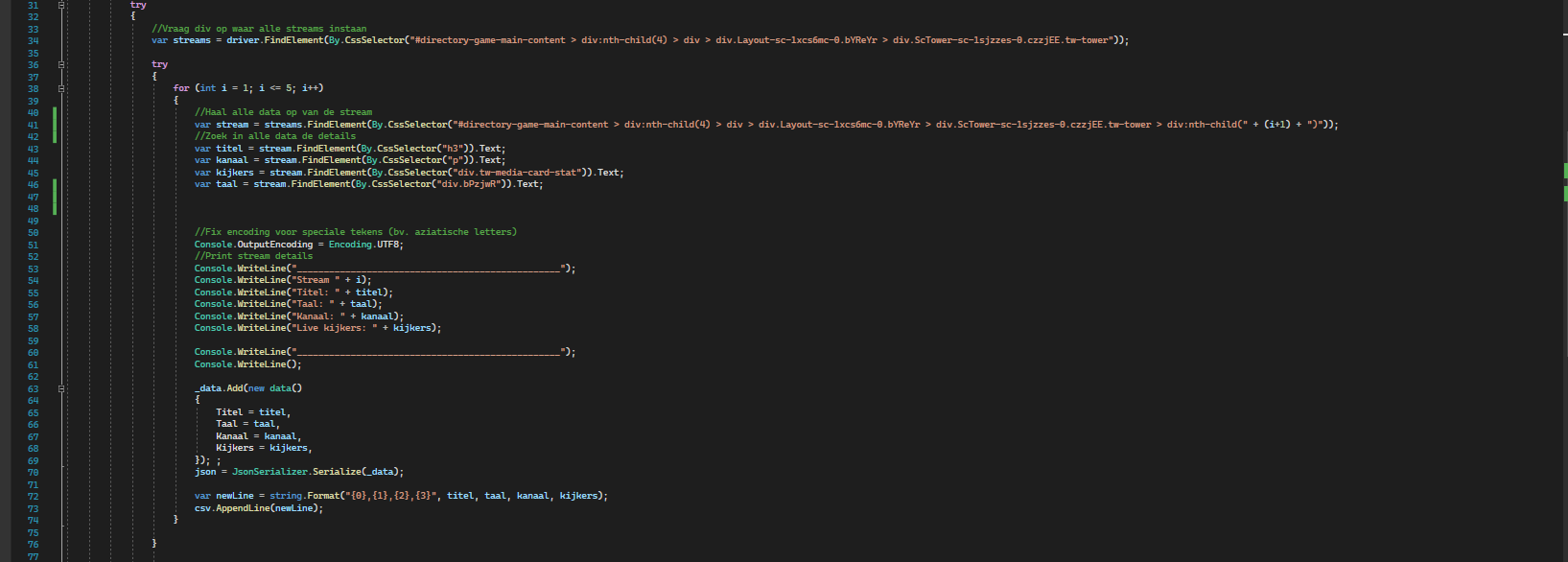
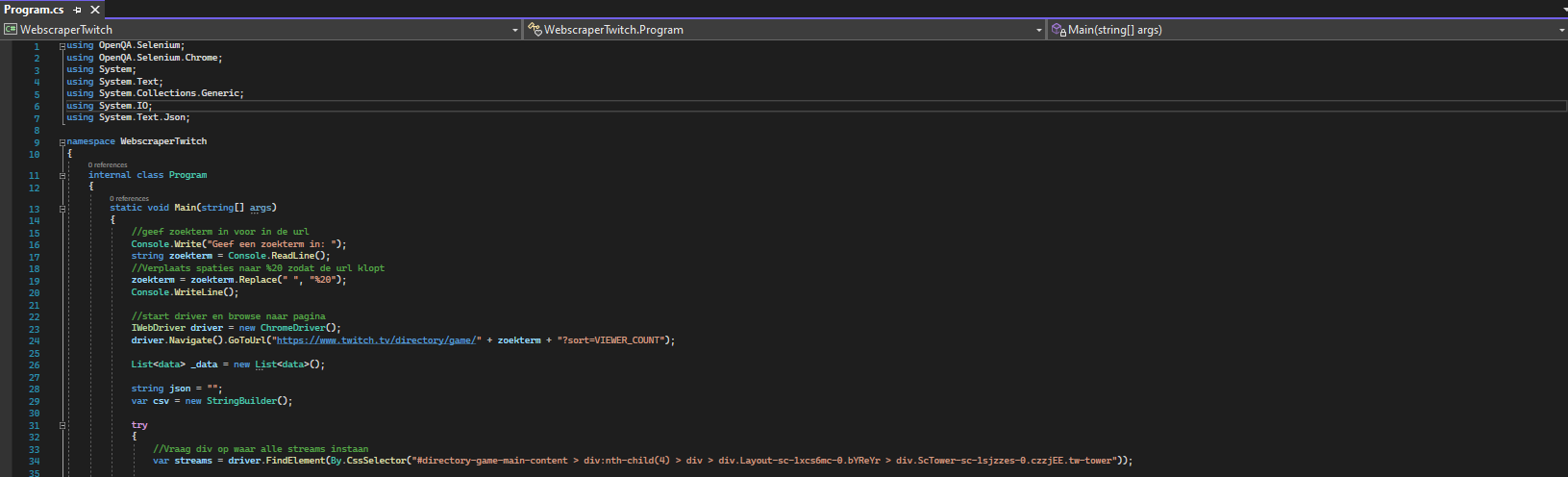
Automatisch gegenereerde beschrijving

De code van de “data.cs” is ook bijna hetzelfde   
alleen met andere namen voor de variabelen.

## Twitch scraper

### Full code

program.cs file



Data.cs file

Afbeelding met tekst, monitor, schermafbeelding, scherm

Automatisch gegenereerde beschrijving

### Code in detail

Het begin van de code is weer perfect hetzelfde als de YouTube scraper buiten dat we de spaties naar “%20” veranderen ipv naar een “+”.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Als volgende starten we de driver en gaan we naar de link. Zoals je ziet steken we de zoekterm in de url om naar de juiste pagina te gaan. Op het einde van de url voegen we “?sort=VIEWER\_COUNT” toe om te filteren op meeste naar minste kijkers.

Verder maken we weer de list aan om later ons resultaat in json om te zetten en we maken de var csv aan.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

In onderstaande screenshot is de tweede “try” collapsed om het duidelijker te maken.  
In de code die je nu ziet gaan we kijken of de categorie te vinden is. Als dit niet zo is zouden we normaal een error krijgen maar deze vangen we op met de “catch”. De variabele error wordt in de console geschreven en wordt ook opgeslagen in het Json en Csv bestand.

Als er wel een categorie gevonden is gaan we door naar de tweede “try”.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Deze “try” checkt of de for loop 5 keer kan worden gedaan. Als er een categorie is waar er bijvoorbeeld maar 3 streams van te vinden zijn gaat hij bij de vierde loop een error willen geven. Dit vangen we weer op met de catch. In deze catch wordt de variabele “error” weer in de Json en Csv file gestoken met de tekst “Er zijn niet genoeg streams in deze categorie om aan 5 te komen.”.

Afbeelding met tekst

Automatisch gegenereerde beschrijvingAfbeelding met tekst

Automatisch gegenereerde beschrijving

De for loop zelf is vrij simpel. We zoeken naar een stream en die slaan we in zijn geheel op als “stream”. Vanuit hier gaan we zoeken naar de details van de stream. En steken we deze in gepaste variabelen om ze daarna allemaal te printen in de console en in de Json en Csv files te steken.

Op het einde van de code vind je dat ik op dezelfde manier terug de data naar de json en Csv file schrijf in de map DevOps. Alleen dan nu onder de namen “Twitch.csv” en “Twitch.json”

De Console.ReadLine op het einde plaatsen we zodat de console open blijft staan en niet direct sluit.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

De code van de “data.cs” is ook bijna hetzelfde   
alleen met andere namen voor de variabelen. Afbeelding met tekst, monitor, schermafbeelding, binnen

Automatisch gegenereerde beschrijving

# Bronnen

## De basis

Scrapax. (2019, 25 augustus). *How to scrape websites using Selenium in C#* [Video]. YouTube. <https://www.youtube.com/watch?v=CpugqTr2j60>

## Extra

Joshi, M. (2022, 29 september). *Wait Commands in Selenium C and C#*. BrowserStack. <https://www.browserstack.com/guide/wait-commands-in-selenium-c-and-c-sharp>

*How to write a JSON file in C#?* (2013, 4 juni). Stack Overflow. <https://stackoverflow.com/questions/16921652/how-to-write-a-json-file-in-c>

*Writing data into CSV file in C#*. (2013, 12 september). Stack Overflow. <https://stackoverflow.com/questions/18757097/writing-data-into-csv-file-in-c-sharp>