



DevOps & Security

Case study

Student: Ruben Heurckmans

R-number: R0831546

Academic year 2022-2023

Inleiding.....	3
Link naar github.....	3
Demo	3
Video.....	3
Uitleg	3
Oplossing.....	6
Aanmaken project	6
Selenium toevoegen aan je project.....	7
Code.....	8
Full code	8
Code in detail.....	13
Main function	13
GetUrl fuction.....	14
StartScraper function	14
YtScraper function.....	15
lctJobScraper function.....	17
TwitchScraper function	19
WriteFiles function	20
Bronnen.....	21
De basis	21
Extra.....	21

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://www.youtube.com/watch?v=UutNLhN8moM>

Uitleg

Je start het programma. Deze gaat je de keuze geven tussen drie verschillende webscrapers, deze kan je kiezen door een 1, 2 of een 3 te typen. Daarna vraagt hij om een zoekterm in te geven.

```
D:\Projecten\VisualStudio\DevOpsTaak\DevopsCaseStudy\WebScraper\bin\Debug\net5.0\WebScraper.exe
welke scraper wilt u gebruiken (1 = Youtube, 2 = IctJob, 3 = Twitch): 1
Geef een zoekterm in:
```

Hierin vul je je zoekterm in en dan gaat de scraper naar de gepaste link aan de hand van de gekozen scraper en de ingegeven zoekterm.

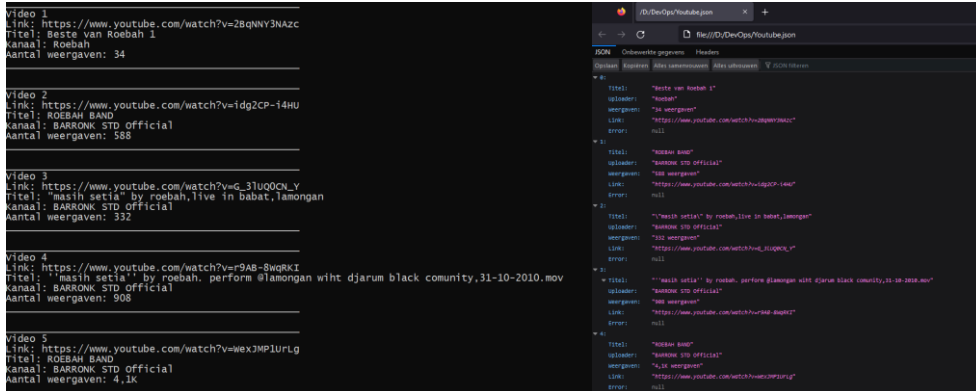
Onderstaande voorbeelden zijn resultaten van de YouTube scraper.

```
D:\Projecten\VisualStudio\DevOpsTaak\DevopsCaseStudy\WebScraper\bin\Debug\net5.0\WebScraper.exe
welke scraper wilt u gebruiken (1 = Youtube, 2 = IctJob, 3 = Twitch): 1
Geef een zoekterm in: roebah
```

Youtube.csv - Kladblok

Bestand Bewerken Opmaak Beeld Help

Beste van Roebah 1,Roebah,34 weergaven,https://www.youtube.com/watch?v=2BqNMY3NAzc
 ROEBAH BAND,BARRONK STD Official,588 weergaven,https://www.youtube.com/watch?v=idg2CP-14HU
 "masih setia" by roebah,live in babat,lamongan,BARRONK STD Official,332 weergaven,https://www.youtube.com/watch?v=G_31UQ0CN_Y
 "masih setia" by roebah. perform @lamongan wint djarum black community,31-10-2010.mov,BARRONK STD Official,908 weergaven,https://www.youtube.com/watch?v=r9AB-8WqRKI
 ROEBAH BAND,BARRONK STD Official,4,1K weergaven,https://www.youtube.com/watch?v=WexJMP1UrLg



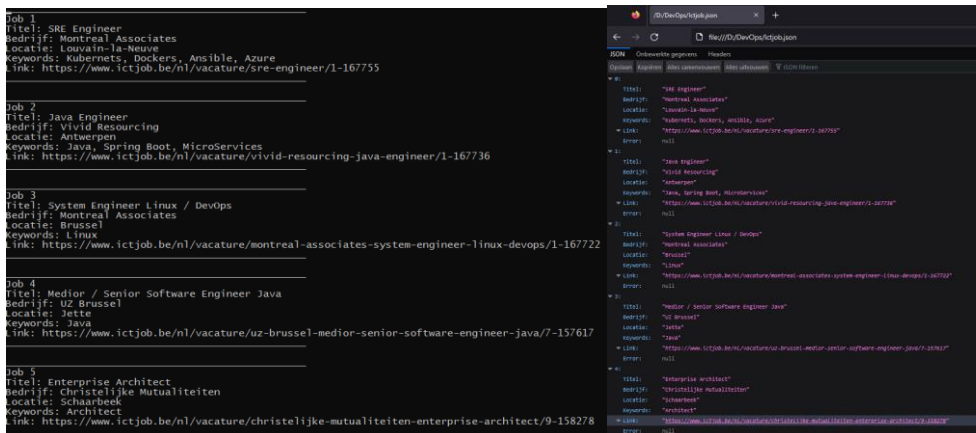
Hier enkele voorbeelden van de IctJob scraper

Welke scraper wilt u gebruiken (1 = Youtube, 2 = IctJob, 3 = Twitch): 2
 Geef een zoekterm in: java

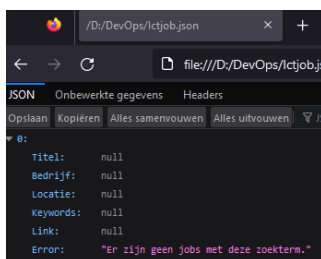
IctJob.csv - Kladblok

Bestand Bewerken Opmaak Beeld Help

SRE Engineer,Montreal Associates,Louvain-la-Neuve,Kubernetes, Docker, Ansible, Azure,https://www.ictjob.be/nl/vacature/sre-engineer/1-167755
 Java Engineer,Vivid Resourcing,Antwerpen,Java, Spring Boot, MicroServices,https://www.ictjob.be/nl/vacature/vivid-resourcing-java-engineer/1-167736
 System Engineer Linux / DevOps,Montreal Associates,Brussel,Linux,https://www.ictjob.be/nl/vacature/montreal-associates-system-engineer-linux-devops/1-167722
 Medior / Senior Software Engineer Java,UZ Brussel,Jette,Java,https://www.ictjob.be/nl/vacature/uz-brussel-medior-senior-software-engineer-java/7-157617
 Enterprise Architect,Christelijke Mutualiteiten,Schaarbeek,Architect,https://www.ictjob.be/nl/vacature/christelijke-mutualiteiten-enterprise-architect/9-158278



Welke scraper wilt u gebruiken (1 = Youtube, 2 = IctJob, 3 = Twitch): 2
 Geef een zoekterm in: webdesign
 Er zijn geen jobs met deze zoekterm.

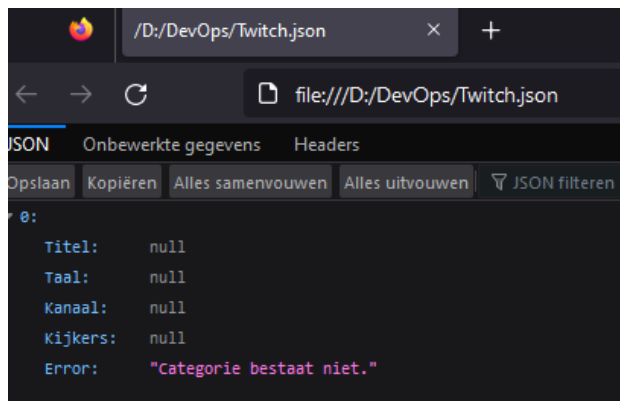


IctJob.csv - Kladblok


Bestand Bewerken Opmaak Beeld Help

Er zijn geen jobs met deze zoekterm.

En hier enkelen van twitch

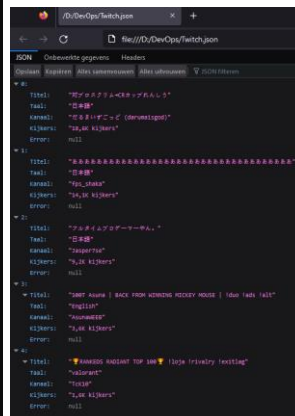
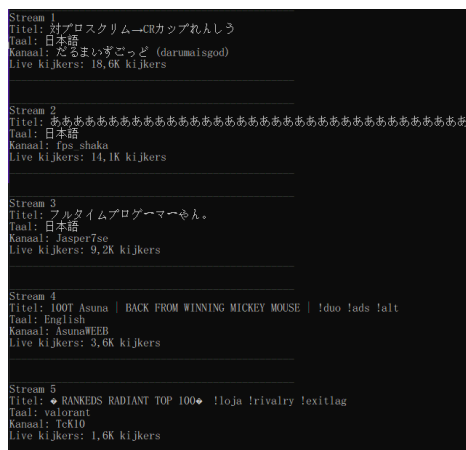


Categorie bestaat niet.

 Twitch.csv - Kladblok

Bestand Bewerken Opmaak Beeld Help

Categorie bestaat niet.



 Twitch.csv - Kladblok

Bestand Bewerken Opmaak Beeld Help

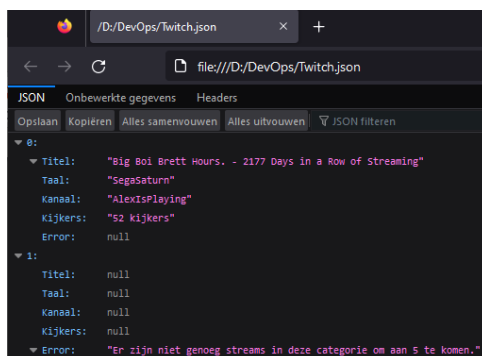
対プロスクリム→CRカップれんしゅう, 日本語, だるまいずごっど (darumaisgod), 18,6K kijkers

[illegible]

フルタイムプログラマーやん。 , 日本語, Jasper7se, 9, 2K kijkers

100T Asuna | BACK FROM WINNING MICKEY MOUSE | !duo !ads !alt,English,AsunaWEEB,3,6K kijkers

🏆RANKEDS RADIANT TOP 100🏆 !loja !rivalry !exitlag,valorant,TcK10,1,6K kijkers



Stream 1
Titel: Big Boi Brett Hours. - 2177 Days in a Row of Streaming
Taal: Sega Saturn
Kanaal: AlexIsPlaying
Live kijkers: 52 kijkers

Er zijn niet genoeg streams in deze categorie om aan 5 te komen.

Twitch.csv - Kladblok

Bestand Bewerken Opmaak Beeld Help

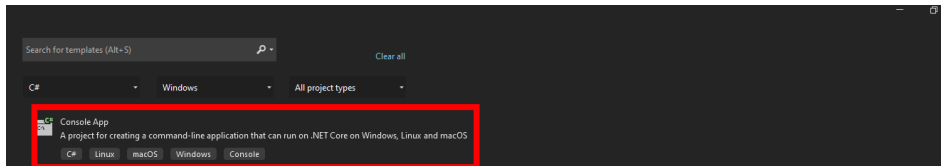
Big Boi Brett Hours. - 2177 Days in a Row of Streaming, Sega Saturn, AlexisPlaying, 52 kijkers
Er zijn niet genoeg streams in deze categorie om aan 5 te komen.

Er zijn niet genoeg streams in deze categorie om aan 5 te komen.

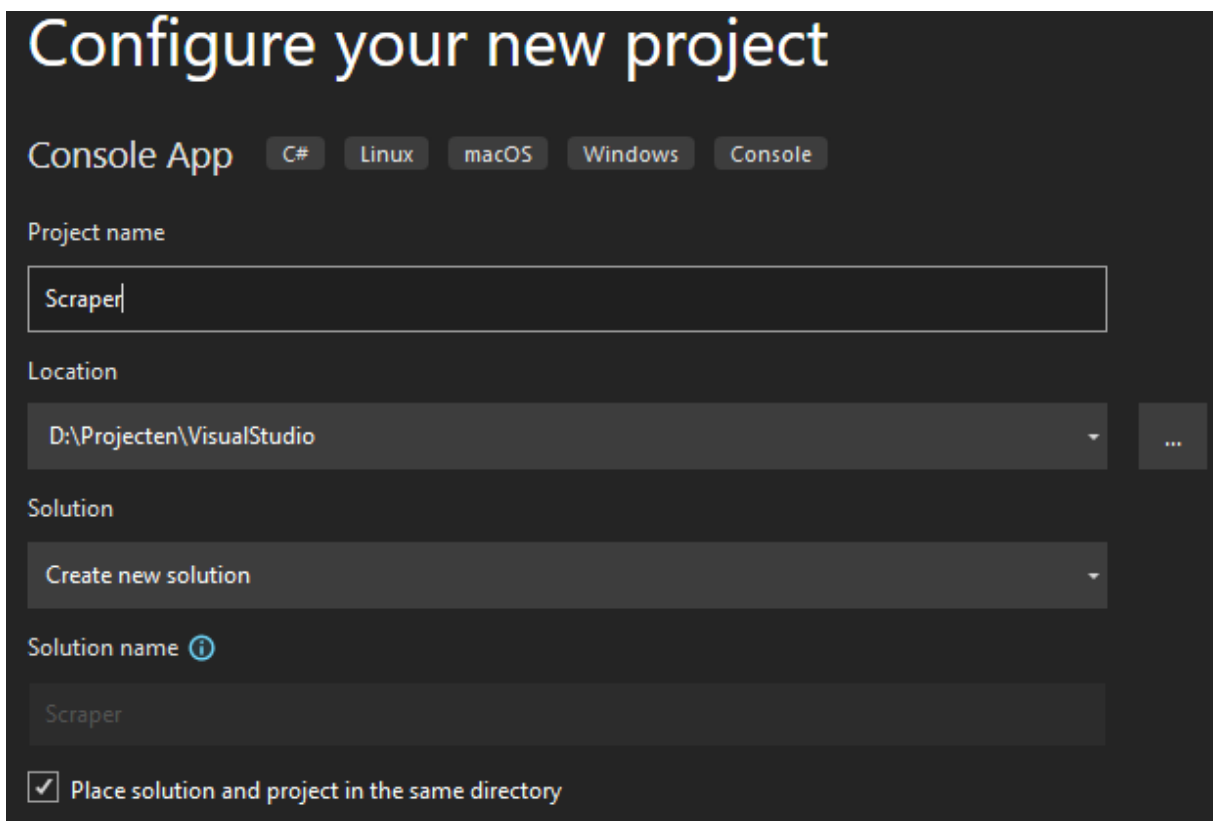
Oplossing

Aanmaken project

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



Daarna geef je je project een naam. Mijn project noemt "Scraper".

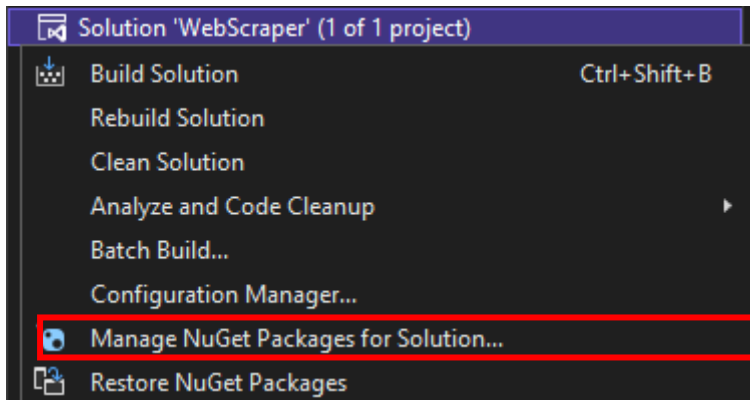


Voor framework heb ik ".NET 5.0" genomen.

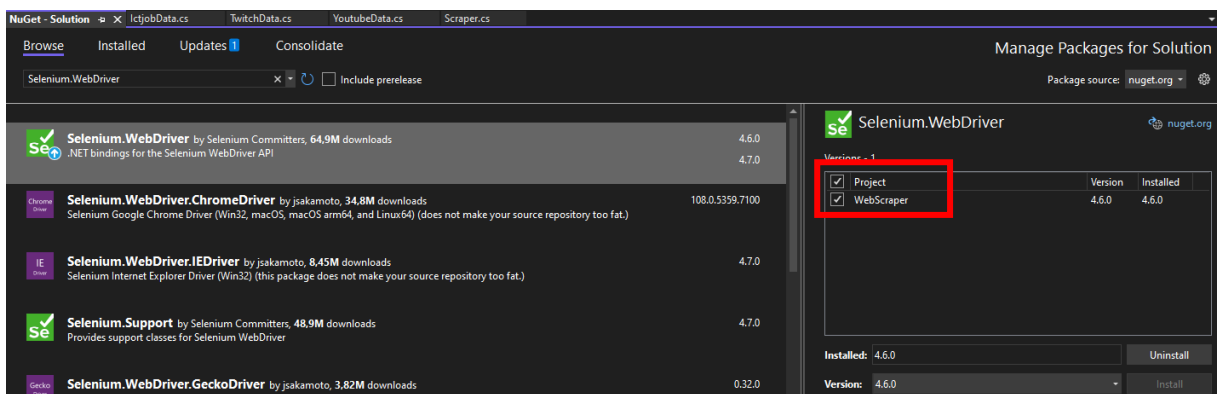
Daarna druk je op create.

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”.



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.



Code

Full code

Scraper.cs (door kleine aanpassingen kloppen niet alle line numbers)

```
Scraper.cs
1 using OpenQA.Selenium;
2 using OpenQA.Selenium.Chrome;
3 using System;
4 using System.Collections.Generic;
5 using System.IO;
6 using System.Text;
7 using System.Text.Json;
8
9 namespace WebScraper
10 {
11     class Scraper
12     {
13         0 references
14         public static void Main(string[] args)
15         {
16             int keuzeScraper;
17             string zoekterm;
18             try{
19                 Console.WriteLine("Welke scraper wilt u gebruiken (1 = Youtube, 2 = IctJob, 3 = Twitch): ");
20                 keuzeScraper = Convert.ToInt32(Console.ReadLine());
21             }
22             catch { keuzeScraper = 4;}
23             while (keuzeScraper != 0)
24             {
25                 if (0 > keuzeScraper || keuzeScraper > 3){
26                     try{
27                         Console.WriteLine("Welke scraper wilt u gebruiken (1 = Youtube, 2 = IctJob, 3 = Twitch): ");
28                         keuzeScraper = Convert.ToInt32(Console.ReadLine());
29                     }
30                     catch { keuzeScraper = 4;}
31                 }
32                 else{
33                     Console.WriteLine("Geef een zoekterm in: ");
34                     zoekterm = Console.ReadLine();
35
36                     GetUrl(keuzeScraper, zoekterm);
37                     keuzeScraper = 4;
38                 }
39             }
40
41             1 reference
42             static string GetUrl(int keuzeScraper, string zoekterm)
43             {
44                 string url = "";
45                 switch (keuzeScraper)
46                 {
47                     case 1:
48                         zoekterm = zoekterm.Replace(" ", "+");
49                         url = "https://www.youtube.com/results?search_query=" + zoekterm + "&sp=CAI%253D";
50                         YtScraper(StartScraper(url));
51                         break;
52                     case 2:
53                         zoekterm = zoekterm.Replace(" ", "+");
54                         url = "https://www.ictjob.be/nl/it-vacatures-zoeken?keywords=" + zoekterm;
55                         IctJobScraper(StartScraper(url));
56                         break;
57                     case 3:
58                         zoekterm = zoekterm.Replace(" ", "%20");
59                         url = "https://www.twitch.tv/directory/game/" + zoekterm + "?sort=VIEWER_COUNT";
60                         TwitchScraper(StartScraper(url));
61                         break;
62                 }
63                 return url;
64             }
65
66             3 references
67             static IWebDriver StartScraper(string url)
68             {
69                 IWebDriver driver = new ChromeDriver();
70                 driver.Navigate().GoToUrl(url);
71                 return driver;
72             }
73
74             3 references
75             static void WriteFiles(StringBuilder csv, string json, string scraper)
76             {
77                 Directory.CreateDirectory("D:\\DevOps");
78                 File.WriteAllText(@"D:\DevOps\"+scraper+".csv", csv.ToString());
79                 File.WriteAllText(@"D:\DevOps\"+ scraper + ".json", json);
80                 Console.ReadLine();
81             }
82         }
83     }
84 }
```



```

1reference
static void YtScraper(IWebDriver driver)
{
    List<YoutubeData> _data = new List<YoutubeData>();

    string json = "";
    var csv = new StringBuilder();

    var cookiebutton = driver.FindElement(By.XPath("/html/body/ytd-app/ytd-consent-bump-v2-lightbox/tp-yt-paper-dialog/div[4]/div[2]/div[6]/div[1]/ytd-button-renderer[1]/yt-button-shape/button"));
    cookiebutton.Click();

    try
    {
        //Zoek de eerste video
        driver.FindElement(By.CssSelector("ytd-video-renderer:nth-child(1)"));
        try
        {
            for (int i = 1; i <= 5; i++)
            {
                var video = driver.FindElement(By.CssSelector("ytd-video-renderer:nth-child(" + i + ")"));
                string titel = video.FindElement(By.CssSelector("#video-title")).Text;
                string weergaven = video.FindElement(By.CssSelector("#metadata-line > span:nth-child(3)")).Text;
                string uploader = video.FindElement(By.Id("channel-info")).Text;
                string link = video.FindElement(By.CssSelector("#video-title")).GetAttribute("href");

                //Print video details
                Console.WriteLine("-----");
                Console.WriteLine("Video " + i);
                Console.WriteLine("Link: " + link);
                Console.WriteLine("Titel: " + titel);
                Console.WriteLine("Kanaal: " + uploader);
                Console.WriteLine("Aantal weergaven: " + weergaven.Replace("weergaven", ""));
                Console.WriteLine("-----");
                Console.WriteLine();

                _data.Add(new YoutubeData()
                {
                    Titel = titel,
                    Uploader = uploader,
                    Weergaven = weergaven,
                    Link = link
                }); ;
                json = JsonSerializer.Serialize(_data);

                var newLine = string.Format("{0},{1},{2},{3}", titel, uploader, weergaven, link);
                csv.AppendLine(newLine);
            }
        }
        catch
        {
            string error = "Er zijn niet genoeg video's met deze zoekterm om aan 5 te komen.";
            Console.WriteLine(error);

            _data.Add(new YoutubeData()
            {
                Error = error,
            }); ;
            json = JsonSerializer.Serialize(_data);

            var newLine = string.Format("{0}", error);
            csv.AppendLine(newLine);
        }
    }
    catch
    {
        string error = "Er zijn geen video's met deze zoekterm.";
        Console.WriteLine(error);

        _data.Add(new YoutubeData()
        {
            Error = error,
        }); ;
        json = JsonSerializer.Serialize(_data);

        var newLine = string.Format("{0}", error);
        csv.AppendLine(newLine);
    }
}

WriteFiles(csv, json, "Youtube");

```

```

156 //reference
157 static void IctJobScraper(IWebDriver driver)
158 {
159     driver.Manage().Timeouts().ImplicitWait = TimeSpan.FromSeconds(4);
160
161     var sortbutton = driver.FindElement(By.CssSelector("#sort-by-date"));
162     sortbutton.Click();
163
164     List<IctjobData> _data = new List<IctjobData>();
165
166     string json = "";
167     var csv = new StringBuilder();
168     try
169     {
170         //Vraag div op waar alle jobs instaan
171         var vacatures = driver.FindElement(By.CssSelector("#search-result-body"));
172         try
173         {
174             for (int i = 1; i <= 6; i++)
175             {
176                 if (i != 4)
177                 {
178                     var vacature = vacatures.FindElement(By.CssSelector("li:nth-child(" + i + ")"));
179
180                     var titel = vacature.FindElement(By.CssSelector(".job-title")).Text;
181                     var bedrijf = vacature.FindElement(By.CssSelector(".job-company")).Text;
182                     var locatie = vacature.FindElement(By.CssSelector(".job-location")).Text;
183                     var keywords = "";
184                     try {
185                         keywords = vacature.FindElement(By.CssSelector(".job-keywords")).Text; }
186                     catch { keywords = "Geen keywords"; }
187                     var link = vacature.FindElement(By.CssSelector(".job-title")).GetAttribute("href");
188
189                     //Print Job details
190                     Console.WriteLine("-----");
191                     if (i <= 3){
192                         Console.WriteLine("Job " + i); }
193                     else {
194                         Console.WriteLine("Job " + (i - 1)); }
195                     Console.WriteLine("Titel: " + titel);
196                     Console.WriteLine("Bedrijf: " + bedrijf);
197                     Console.WriteLine("Locatie: " + locatie);
198                     Console.WriteLine("Keywords: " + keywords);
199                     Console.WriteLine("Link: " + link);
200                     Console.WriteLine(" ");
201
202                     _data.Add(new IctjobData()
203                     {
204                         Titel = titel,
205                         Bedrijf = bedrijf,
206                         Locatie = locatie,
207                         Keywords = keywords,
208                         Link = link
209                     }); ;
210                     json = JsonSerializer.Serialize(_data);
211
212                     var newLine = string.Format("{0},{1},{2},{3},{4}", titel, bedrijf, locatie, keywords, link);
213                     csv.AppendLine(newLine);
214                 }
215             }
216         }
217         catch
218         {
219             string error = "Er zijn niet genoeg jobs met deze keyword(s) om aan 5 te komen.";
220             Console.WriteLine(error);
221             _data.Add(new IctjobData()
222             {
223                 Error = error,
224             }); ;
225             json = JsonSerializer.Serialize(_data);
226
227             var newLine = string.Format("{0}", error);
228             csv.AppendLine(newLine);
229         }
230     }
231     catch
232     {
233         string error = "Er zijn geen jobs met deze zoekterm.";
234         Console.WriteLine(error);
235
236         _data.Add(new IctjobData()
237         {
238             Error = error,
239         }); ;
240         json = JsonSerializer.Serialize(_data);
241
242         var newLine = string.Format("{0}", error);
243         csv.AppendLine(newLine);
244     }
245 }
246 WriteFiles(csv, json, "IctJob");
247
248

```

```

249 static void TwitchScraper(IWebDriver driver)
250 {
251     List<TwitchData> _data = new List<TwitchData>();
252
253     string json = "";
254     var csv = new StringBuilder();
255
256     try
257     {
258         //Vraag div op waar alle streams instaan
259         var streams = driver.FindElement(By.CssSelector("#directory-game-main-content > div:nth-child(4) > div > div.Layout-sc-1xcs6mc-0.bYReVr > div.ScTower-sc-1sjzzes-0.czz
260
261         try
262         {
263             for (int i = 1; i <= 5; i++)
264             {
265                 //Haal alle data op van de stream
266                 var stream = streams.FindElement(By.CssSelector("#directory-game-main-content > div:nth-child(4) > div > div.Layout-sc-1xcs6mc-0.bYReVr > div.ScTower-sc-1sjzz
267                 //Zoek in alle data de details
268                 string titel = stream.FindElement(By.CssSelector("h3")).Text;
269                 string kanaal = stream.FindElement(By.CssSelector("p")).Text;
270                 string kijkers = stream.FindElement(By.CssSelector("div.tw-media-card-stat")).Text;
271                 string taal = "";
272                 try
273                 {
274                     taal = stream.FindElement(By.CssSelector("div.bPzjwR")).Text;
275                 }
276                 catch
277                 {
278                     taal = "Geen taal";
279                 }
280
281                 //Fix encoding voor speciale tekens (bv. aziatische letters)
282                 Console.OutputEncoding = Encoding.UTF8;
283                 //Print stream details
284                 Console.WriteLine("-----");
285                 Console.WriteLine("Stream " + i);
286                 Console.WriteLine("Titel: " + titel);
287                 Console.WriteLine("Taal: " + taal);
288                 Console.WriteLine("Kanaal: " + kanaal);
289                 Console.WriteLine("Live kijkers: " + kijkers);
290                 Console.WriteLine("-----");
291                 Console.WriteLine();
292
293                 _data.Add(new TwitchData()
294                 {
295                     Titel = titel,
296                     Taal = taal,
297                     Kanaal = kanaal,
298                     Kijkers = kijkers,
299                 });
300                 json = JsonSerializer.Serialize(_data);
301
302                 var newLine = string.Format("{0},{1},{2},{3}", titel, taal, kanaal, kijkers);
303                 csv.AppendLine(newLine);
304             }
305         }
306         catch
307         {
308             string error = "Er zijn niet genoeg streams in deze categorie om aan 5 te komen.";
309             Console.WriteLine(error);
310
311             _data.Add(new TwitchData()
312             {
313                 Error = error,
314             });
315             json = JsonSerializer.Serialize(_data);
316
317             var newLine = string.Format("{0}", error);
318             csv.AppendLine(newLine);
319         }
320     }
321     catch
322     {
323         string error = "Categorie bestaat niet.";
324         Console.WriteLine(error);
325
326         _data.Add(new TwitchData()
327         {
328             Error = error,
329         });
330         json = JsonSerializer.Serialize(_data);
331
332         var newLine = string.Format("{0}", error);
333         csv.AppendLine(newLine);
334     }
335     WriteFiles(csv, json, "Twitch");
336 }
337
338

```

YoutubeData.cs

```
1 namespace WebScraper
2 {
3     internal class YoutubeData
4     {
5         public string Titel { get; internal set; }
6         public string Uploader { get; internal set; }
7         public string Weergaven { get; internal set; }
8         public string Link { get; internal set; }
9         public string Error { get; internal set; }
10    }
11 }
```

IctjobData.cs

```
1 namespace WebScraper
2 {
3     internal class IctjobData
4     {
5         public string Titel { get; internal set; }
6         public string Bedrijf { get; internal set; }
7         public string Locatie { get; internal set; }
8         public string Keywords { get; internal set; }
9         public string Link { get; internal set; }
10        public string Error { get; internal set; }
11    }
12 }
```

TwitchData.cs

```
1 namespace WebScraper
2 {
3     internal class TwitchData
4     {
5         public string Titel { get; internal set; }
6         public string Taal { get; internal set; }
7         public string Kanaal { get; internal set; }
8         public string Kijkers { get; internal set; }
9         public string Error { get; internal set; }
10    }
11 }
```

Code in detail

Main function

Als eerste declareren we de variabelen "keuzeScraper" en "zoekterm".

Daarna gaan we de gebruiker vragen welke scraper hij/zij wilt gaan gebruiken. De waarde die wordt ingelezen converteren we naar een Int32 en slaan we op in "KeuzeScraper". Dit zetten we in een try zodat de applicatie geen error geeft als de gebruiker iets anders dan een geheel getal invult. Als de gebruiker bijvoorbeeld een woord zou ingeven wordt dit opgevangen door de catch waarbij "keuzeScraper" altijd 4 wordt.

Dan start de while loop. Zolang "keuzeScraper" niet 0 is blijft deze loop gaande. Als je 0 invult dan stopt de applicatie. Zolang je niet 1, 2 of 3 invult zal de vraag zich blijven herhalen. Ook hier staat dezelfde try, catch rond zodat de applicatie geen error krijgt.

Als je 1, 2 of 3 ingeeft gaat de console vragen om een zoekterm in te vullen. Deze wordt dan opgeslagen onder "zoekterm".

Daarna wordt de functie "GetUrl()" opgeroepen met "keuzeScraper" en "zoekterm" als parameter. Daarna maken we "keuzeScraper" terug 4 zodat hij niet automatisch weer gaat vragen voor een zoekterm.

```
13 public static void Main(string[] args)
14 {
15     int keuzeScraper;
16     string zoekterm;
17     try{
18         Console.Write("Welke scraper wilt u gebruiken (1 = Youtube, 2 = IctJob, 3 = Twitch): ");
19         keuzeScraper = Convert.ToInt32(Console.ReadLine());
20     }
21     catch { keuzeScraper = 4;}
22     while (keuzeScraper != 0)
23     {
24         if (0 > keuzeScraper || keuzeScraper > 3){
25             try{
26                 Console.Write("Welke scraper wilt u gebruiken (1 = Youtube, 2 = IctJob, 3 = Twitch): ");
27                 keuzeScraper = Convert.ToInt32(Console.ReadLine());
28             }
29             catch { keuzeScraper = 4;}
30         }
31         else{
32             Console.Write("Geef een zoekterm in: ");
33             zoekterm = Console.ReadLine();
34
35             GetUrl(keuzeScraper, zoekterm);
36             keuzeScraper = 4;
37         }
38     }
39 }
```

GetUrl fuction

We declareren als eerste de variabele string. Deze gaat later als parameter gebruikt worden in de fuctie "StartScrapper()".

Om te bepalen welke url we nodig hebben gebruiken we een switch met 3 cases. Eentje voor als de waarde 1 is, eentje voor 2 en eentje voor 3. Welke case het wordt hangt af van de waarde van "keuzeScrapper".

In de cases worden de spaties in "zoekterm" verandert met de waarde dat de site gebruikt om spaties in de url te verwerken. Bij youtube en de ictjob site is dit een "+" en bij twitch "%20".

Daarna roepen we de functie op van de juiste scraper met als parameter de waarde die we krijgen uit de functie StartScrapper met als parameter de url die zonet gemaakt is.

```
1 reference
41 static string GetUrl(int keuzeScrapper, string zoekterm)
42 {
43     string url = "";
44     switch (keuzeScrapper)
45     {
46         case 1:
47             zoekterm = zoekterm.Replace(" ", "+");
48             url = "https://www.youtube.com/results?search_query=" + zoekterm + "&sp=CAI%253D";
49             YtScrapper(StartScrapper(url));
50             break;
51         case 2:
52             zoekterm = zoekterm.Replace(" ", "+");
53             url = "https://www.ictjob.be/nl/it-vacatures-zoeken?keywords=" + zoekterm;
54             IctJobScrapper(StartScrapper(url));
55             break;
56         case 3:
57             zoekterm = zoekterm.Replace(" ", "%20");
58             url = "https://www.twitch.tv/directory/game/" + zoekterm + "?sort=VIEWER_COUNT";
59             TwitchScrapper(StartScrapper(url));
60             break;
61     }
62     return url;
63 }
```

StartScrapper function

Deze functie start de driver, navigeert naar de url die doorgegeven is als parameter en geeft daarna de waarde driver terug. Deze gaat dan gebruikt worden in de scraper functies.

```
3 references
65 static IWebDriver StartScrapper(string url)
66 {
67     IWebDriver driver = new ChromeDriver();
68     driver.Navigate().GoToUrl(url);
69     return driver;
70 }
```

YtScraper function

Om te beginnen maken we een List aan met als type YoutubeData. Dit is een class in de file YoutubeData.cs, die zichtbaar is in

onderstaande screenshot. Deze list gaan we

gebruiken om onze json file op te bouwen. Daarna maken we nog een variabele json en csv aan waar we de data naartoe gaan schrijven.

```
80 1 reference
81 static void YtScraper(IWebDriver driver)
82 {
83     List<YoutubeData> _data = new List<YoutubeData>();
84     string json = "";
85     var csv = new StringBuilder();
```

YoutubeData.cs

```
1 namespace WebScraper
2 {
3     5 references
4     internal class YoutubeData
5     {
6         1 reference
7         public string Titel { get; internal set; }
8         1 reference
9         public string Uploader { get; internal set; }
10        1 reference
11        public string Weergaven { get; internal set; }
12        1 reference
13        public string Link { get; internal set; }
14        2 references
15        public string Error { get; internal set; }
16    }
17 }
```

Youtube gaat vragen om akkoord te gaan met de cookies. Als we deze pop-up niet wegdoen kunnen we de data niet lezen dus we laten de driver klikken op de knop "akkoord".

```
87 var cookiebutton = driver.FindElement(By.XPath("/html/body/ytd-app/ytd-consent-bump-v2-lightbox/tp-yt-paper-dialog/div[4]/div[2]/div[6]/div[1]/ytd-button-renderer[1]/yt-button-shape/button"));
88 cookiebutton.Click();
89
```

In onderstaande screenshot is de 2^{de} try even ingevouwen zodat het duidelijker wordt.

We laten de driver zoeken naar een video. Als deze er niet zou zijn krijgen we normaal een error en stopt onze applicatie. Dit vangen we op met een try, catch waarbij er door de catch een errormelding in de console komt te staan en deze melding wordt ook in de Json file en de Csv file gezet.

```
90
91
92 //Zoek de eerste video
93 driver.FindElement(By.CssSelector("ytd-video-renderer:nth-child(1)"));
94 try...
141
142 catch
143 {
144     string error = "Er zijn geen video's met deze zoekterm.";
145     Console.WriteLine(error);
146
147     _data.Add(new YoutubeData()
148     {
149         Error = error,
150     }); ;
151     json = JsonSerializer.Serialize(_data);
152
153     var newLine = string.Format("{0}", error);
154     csv.AppendLine(newLine);
155 }
```

De 2^{de} try en catch dient voor als er niet genoeg video's zijn voor een bepaalde zoekterm. De catch daarvan is precies dezelfde als die van de eerste "try" alleen heeft die een andere error message.

In de "try" wordt een for loop 5 keer uitgevoerd. Deze zoekt 5 keer naar een video element en haalt daar de details zoals de titel, het aantal weergaven, ... uit. Een video element ziet er uit zoals in de screenshot.



Verder wordt de data in de console getoond en toegevoegd aan de list om dan in de json variabele gestoken te worden en de waarden worden ook in de csv variabele gezet.

```
194     try
195     {
196         for (int i = 1; i <= 5; i++)
197         {
198             var video = driver.FindElement(By.CssSelector("ytd-video-renderer:nth-child(" + i + ")"));
199             string titel = video.FindElement(By.CssSelector("#video-title")).Text;
200             string weergaven = video.FindElement(By.CssSelector("#metadata-line > span:nth-child(3)")).Text;
201             string uploader = video.FindElement(By.Id("channel-info")).Text;
202             string link = video.FindElement(By.CssSelector("#video-title")).GetAttribute("href");
203
204             //Print video details
205             Console.WriteLine("-----");
206             Console.WriteLine("Video " + i);
207             Console.WriteLine("Link: " + link);
208             Console.WriteLine("Titel: " + titel);
209             Console.WriteLine("Kanaal: " + uploader);
210             Console.WriteLine("Aantal weergaven: " + weergaven.Replace("weergaven", ""));
211             Console.WriteLine("-----");
212             Console.WriteLine();
213
214             _data.Add(new YoutubeData()
215             {
216                 Titel = titel,
217                 Uploader = uploader,
218                 Weergaven = weergaven,
219                 Link = link
220             });
221             json = JsonSerializer.Serialize(_data);
222
223             var newLine = string.Format("{0},{1},{2},{3}", titel, uploader, weergaven, link);
224             csv.AppendLine(newLine);
225         }
226     }
```

```
127     catch
128     {
129         string error = "Er zijn niet genoeg video's met deze zoekterm om aan 5 te komen.";
130         Console.WriteLine(error);
131
132         _data.Add(new YoutubeData()
133         {
134             Error = error,
135         });
136         json = JsonSerializer.Serialize(_data);
137
138         var newLine = string.Format("{0}", error);
139         csv.AppendLine(newLine);
140     }
```

En op het einde van de functie wordt de "WriteFiles" functie gebruikt met als parameters csv, json en de string "youtube". Deze functie wordt op het einde uitgelegd.

```
157     WriteFiles(csv, json, "Youtube");
158 }
```


IctJobScraper function

Aangezien er niet veel verschillen zijn tussen de scraperfuncties ga ik vooral focus leggen op de verschillen.

Als eerste laten we de driver 5 seconden wachten. Dit doen we omdat ictjob een protectie heeft voor scrapers waarbij je even moet wachten voor de pagina laadt. In de screenshot is dit te zien.

Uw verzoek wordt verwerkt. Dien uw verzoek opnieuw in als de pagina niet automatisch wordt vernieuwd.



Daarna klikken we op de sort-by-date knop zodat we de juiste sorteermethode gebruiken.

En natuurlijk maken we weer de list aan alleen deze keer met het type IctjobData uit de file "Ictjobdata.cs"

```
159 static void IctJobScraper(IWebDriver driver)
160 {
161     driver.Manage().Timeouts().ImplicitWait = TimeSpan.FromSeconds(4);
162
163     var sortbutton = driver.FindElement(By.CssSelector("#sort-by-date"));
164     sortbutton.Click();
165
166     List<IctjobData> _data = new List<IctjobData>();
167
168     string json = "";
169     var csv = new StringBuilder();
```

```
1 namespace WebScraper
2 {
3     3 references
4     internal class IctjobData
5     {
6         1 reference
7         public string Titel { get; internal set; }
8         1 reference
9         public string Bedrijf { get; internal set; }
10        1 reference
11        public string Locatie { get; internal set; }
12        1 reference
13        public string Keywords { get; internal set; }
14        1 reference
15        public string Link { get; internal set; }
16        2 references
17        public string Error { get; internal set; }
18    }
19 }
```

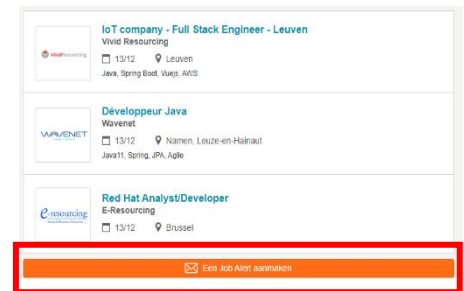
De 2^{de} try is weer collapsed. We zien hier dat de error message aangepast is zodat het past bij de site die we scrapen.

```
170 try
171 {
172     //Vraag div op waar alle jobs instaan
173     var vacatures = driver.FindElement(By.CssSelector("#search-result-body"));
174     try...
234 }
235
236 catch
237 {
238     string error = "Er zijn geen jobs met deze zoekterm.";
239     Console.WriteLine(error);
240
241     _data.Add(new IctjobData()
242     {
243         Error = error,
244     }); ;
245     json = JsonSerializer.Serialize(_data);
246
247     var newLine = string.Format("{0}", error);
248     csv.AppendLine(newLine);
249 }
```

De 2^{de} try is ook grotendeels hetzelfde, natuurlijk met andere CssSelectors maar de basis blijft hetzelfde.

De grote verschillen zijn dat we nu de loop 6 keer gaan doen ipv 5 keer. Dit komt omdat er een call to action op de site staat om zelf een job listing aan te maken en dit zit in een div met dezelfde CssSelector als de joblistings zelf.

Hierdoor moeten we natuurlijk de gepaste if, else schrijven zodat hij de vierde loop geen data opvraagt en zodat in de 5^{de} en 6^{de} loop het juiste getal in de console typt.



Ook zien we dat we de juiste data naar de json en csv variabele schrijven.

```
174         try
175         {
176             for (int i = 1; i <= 6; i++)
177             {
178                 if (i != 4)
179                 {
180                     var vacature = vacatures.FindElement(By.CssSelector("li:nth-child(" + i + ")"));
181
182                     var titel = vacature.FindElement(By.CssSelector(".job-title")).Text;
183                     var bedrijf = vacature.FindElement(By.CssSelector(".job-company")).Text;
184                     var locatie = vacature.FindElement(By.CssSelector(".job-location")).Text;
185                     var keywords = "";
186                     try {
187                         keywords = vacature.FindElement(By.CssSelector(".job-keywords")).Text;
188                     } catch { keywords = "Geen keywords"; }
189                     var link = vacature.FindElement(By.CssSelector(".job-title")).GetAttribute("href");
190
191                     //Print Job details
192                     Console.WriteLine("-----");
193                     if (i <= 3){
194                         Console.WriteLine("Job " + i);
195                     }
196                     else {
197                         Console.WriteLine("Job " + (i - 1));
198                     }
199                     Console.WriteLine("Titel: " + titel);
200                     Console.WriteLine("Bedrijf: " + bedrijf);
201                     Console.WriteLine("Locatie: " + locatie);
202                     Console.WriteLine("Keywords: " + keywords);
203                     Console.WriteLine("Link: " + link);
204                     Console.WriteLine("-----");
205                     Console.WriteLine();
206
207                     _data.Add(new IctJobData()
208                     {
209                         Titel = titel,
210                         Bedrijf = bedrijf,
211                         Locatie = locatie,
212                         Keywords = keywords,
213                         Link = link
214                     });
215                     json = JsonSerializer.Serialize(_data);
216
217                     var newLine = string.Format("{0},{1},{2},{3},{4}", titel, bedrijf, locatie, keywords, link);
218                     csv.AppendLine(newLine);
219                 }
220             }
221         }
```

Daarna roepen we weer de WriteFiles functie aan met de csv, json en de string "IctJob" als parameters.

```
250         WriteFiles(csv, json, "IctJob");
251     }
```

TwitchScraper function

We maken weer een list maar nu met TwitchData.

```
252 static void TwitchScraper(IWebDriver driver)
253 {
254     List<TwitchData> _data = new List<TwitchData>();
255
256     string json = "";
257     var csv = new StringBuilder();
258 }
```

```
1 namespace WebScraper
2 {
3     internal class TwitchData
4     {
5         public string Titel { get; internal set; }
6         public string Taal { get; internal set; }
7         public string Kanaal { get; internal set; }
8         public string Kijkers { get; internal set; }
9         public string Error { get; internal set; }
10    }
11 }
```

In de eerste try gaan we kijken of de categorie wel bestaat. Als deze niet bestaat is er de catch om de error op te vangen

```
259 try
260 {
261     //Vraag div op waar alle streams instaan
262     var streams = driver.FindElement(By.CssSelector("#directory-game-main-content > div:nth-child(4) > div > div.Layout-sc-lxcs6mc-0.bYReYr > div.ScTower-sc-lsJz2es-0.czzjEE.tw-tower"));
263     try
264     {
265         string error = "Categorie bestaat niet.";
266         Console.WriteLine(error);
267
268         _data.Add(new TwitchData()
269         {
270             Error = error,
271         });
272         json = JsonSerializer.Serialize(_data);
273         var newLine = string.Format("{0}", error);
274         csv.AppendLine(newLine);
275     }
276 }
```

In de 2^{de} try gaan we weer door de loop, bij de taal hebben we een extra try, catch omdat sommige mensen geen taal hebben aanstaan. Ook is het te merken dat er op twitch veel aziatische tekens in de titels, talen, ... staan. Om deze leesbaar te maken zetten we de OutputEncoding van de console op UTF8. Dit is vaak niet voldoende, in de demo video zie je hoe ik het font van de console aanpas zodat deze tekens zichtbaar worden.

```
267 try
268 {
269     for (int i = 1; i <= 5; i++)
270     {
271         //Haal alle data op van de stream
272         var stream = streams.FindElement(By.CssSelector("#directory-game-main-content > div:nth-child(4) > div > div.Layout-sc-lxcs6mc-0.bYReYr > div.ScTower-sc-lsJz2es-0.czzjEE.tw-tower > div:nth-child(" + (i + 1) + ")"));
273         //Ziekt de alle data de details
274         string titel = stream.FindElement(By.CssSelector("h3")).Text;
275         string kanaal = stream.FindElement(By.CssSelector("p")).Text;
276         string kijkers = stream.FindElement(By.CssSelector("div.tw-media-card-stat")).Text;
277         string taal = "";
278         try
279         {
280             taal = stream.FindElement(By.CssSelector("div.bPzjdB")).Text;
281         }
282         catch
283         {
284             taal = "Geen taal";
285         }
286         //Via encoding voor speciale tekens (bv. aziatische letters)
287         Console.OutputEncoding = Encoding.UTF8;
288         //Print stream details
289         Console.WriteLine("-----");
290         Console.WriteLine("Stream: " + i);
291         Console.WriteLine("Titel: " + titel);
292         Console.WriteLine("Taal: " + taal);
293         Console.WriteLine("Kanaal: " + kanaal);
294         Console.WriteLine("Live kijkers: " + kijkers);
295         Console.WriteLine("-----");
296     }
297 }
```

En dan heb je nog de data die naar de json en csv variabele worden geschreven.

De catch die een aangepaste error message heeft

En op het einde weer de Writefiles functie met de juiste parameters.

```
295 _data.Add(new TwitchData()
296 {
297     Titel = titel,
298     Taal = taal,
299     Kanaal = kanaal,
300     Kijkers = kijkers,
301 });
302 json = JsonSerializer.Serialize(_data);
303
304 var newLine = string.Format("{0},{1},{2},{3}", titel, taal, kanaal, kijkers);
305 csv.AppendLine(newLine);
306 }
307 catch
308 {
309     string error = "Er zijn niet genoeg streams in deze categorie om aan 5 te komen.";
310     Console.WriteLine(error);
311
312     _data.Add(new TwitchData()
313     {
314         Error = error,
315     });
316     json = JsonSerializer.Serialize(_data);
317     var newLine = string.Format("{0}", error);
318     csv.AppendLine(newLine);
319 }
```

```
338 WriteFiles(csv, json, "Twitch");
339 }
```

WriteFiles function

De WriteFiles functie maakt een directory aan in de D schijf als deze er nog niet is en plaatst daar de juiste json en csv file in aan de hand van de string die meegegeven is onder "scraper".

```
72 3 references  
73 static void WriteFiles(StringBuilder csv, string json, string scraper)  
74 {  
75     Directory.CreateDirectory("D:\\DevOps");  
76     File.WriteAllText(@"D:\\DevOps\\"+scraper+".csv", csv.ToString());  
77     File.WriteAllText(@"D:\\DevOps\\" + scraper + ".json", json);  
78     Console.ReadLine();  
}
```

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>