



1506
UNIVERSITÀ
DEGLI STUDI
DI URBINO
CARLO BO

PROGETTO RETI DI CALCOLATORI

GIAMMARCO GAUDINI

SESSIONE NOVEMBRE 2021

Sviluppo di una WebApp per lo scraping di Film dal sito “www.imdb.com”, con l’utilizzo del tool Cheerio e ed eseguito deploy su Heroku. Inoltre, sviluppo di tre software per lo scraping di Film dai siti “www.imdb.com”, “www.tmdb.com”, “www.rottentomatoes.com” attraverso l’utilizzo del tool Puppeteer.

1. INTRODUZIONE

Il progetto è diviso in due parti principali. La prima riguarda lo sviluppo di una WebApp attraverso il tool Cheerio e la seconda riguarda lo sviluppo di tre script per lo scraping di tre siti differenti attraverso un tool differente, ovvero Puppeteer.

Per lo sviluppo del codice è stato utilizzato il runtime Node.js che permette l'esecuzione del linguaggio javascript.

Il linguaggio utilizzato è il Javascript, che, insieme al Python è il linguaggio di programmazione più utilizzato per lo scraping, in quanto è molto versatile, ben documentato, ma soprattutto sono disponibili innumerevoli librerie.

2. SCRAPING CON CHEERIO

La prima parte risulta essere diviso in due script “index.js” e “imdbCheerioScraper.js”.

- **“index.js”:** In questo script viene utilizzato il framework Express.js, che offre gli strumenti per creare l'applicazione in node.js. Vengono generati i vari percorsi URL che dovranno gestire le richieste che arrivano all'applicazione. Questa applicazione usa le funzioni di scraping dello script imdbCheerioScraper.js, accetta le richieste e restituisce un risultato in formato json.

Abbiamo tre percorsi:

1. (/): è il percorso di base della nostra applicazione, viene restituito un valore in json con il messaggio “Progetto reti di calcolatori”. Usato principalmente per verificare il funzionamento dell'app.
2. (/search/:title/): il quale usa lo scraper per prelevare informazioni dalla pagina di ricerca di IMDB. E avremo come valore di ritorno un array di film.
3. (/movie/:movieID): dove viene passato l'ID di un film per ricavare informazioni su un film specifico.

Viene usata anche la libreria “cors” che permette a qualsiasi server di accedere alle risorse dell'applicazione (img).

```
▼ Generali
Richiedi URL: chrome-extension://gbmdgpbipfallnflgajpalibnhdgobh/assets/viewer.css
Metodo di richiesta: GET
Codice di stato: ● 200 OK
Norme sui referrer: strict-origin-when-cross-origin

▼ Intestazioni della risposta
Access-Control-Allow-Origin: *
cache-control: no-cache
Content-Security-Policy: script-src 'self' blob: filesystem;; object-src 'self' blob: filesystem;;
Content-Type: text/css
Cross-Origin-Resource-Policy: cross-origin
```

- “imdbCheerioScraper.js”: Viene usata la libreria “node-fetch” ovvero uno specifico pacchetto di Node.js che ci permette di fare richieste http. Le richieste vengono inoltrate a un “url”+”searchTerm” cioè il nome del film che vogliamo cercare. Mentre se vogliamo analizzare un film specifico, viene mandata la richiesta a “movieUrl”+”movieID”.

Per grattare queste informazioni ci si avvale del tool Cheerio che è una libreria di Node.js che aiuta gli sviluppatori ad interpretare e analizzare pagine web usando la sintassi di JQuery.

Abbiamo due funzioni per lo scraping:

1. searchMovie:

Viene inoltrata la richiesta html alla pagina web, successivamente viene preso il suo codice html e viene “caricato” con il comando .load() su cheerio. Attraverso il selezionatore \$ posso prelevare i parametri di cui ho bisogno. In questo caso, la funzione preleva tutti [.each()] i risultati (.findResult) e successivamente estrae i campi di mio interesse (title, image, movieID).

```
//prelevo dalla classe findResult gli elementi che mi servono
$('.findResult').each(function(i, element){
  const $element = $(element);
  //prelevo il titolo dei film
  const $title = $element.find('td.result_text a')
```

In fine viene creato un oggetto di ritorno chiamato “movie” che contiene i campi appena prelevati e caricato all’interno di un array “movies”.

2. getMovie:

È una funzione utilizzata per “grattare” le informazioni di un film specifico. Gli viene passata la variabile “movieID” il quale inoltra la richiesta alla pagina del film individuale. Viene restituito l’html di quella pagina e caricato su cheerio. Gli elementi vengono estratti attraverso l’uso del selezionatore \$ e la conoscenza dei percorsi per arrivare a quelle risorse.

```
//prelevo il titolo del film
const $title = $("#__next > main > div > section.ipc-page-background.ipc-page-background--base.TitlePage__StyledPageBackground-wzlr49";
const title = $title.text();

//prelevo la durata del film
const $runTime = $("#__next > main > div > section.ipc-page-background.ipc-page-background--base.TitlePage__StyledPageBackground-wzlr49;
const durata = $runTime.text();

//prelevo il o i generi
//creo un array che contiene i generi
const $genre = $("#__next > main > div > section.ipc-page-background.ipc-page-background--base.TitlePage__StyledPageBackground-wzlr49;
const generi = $genre.text();

//prelevo la data di pubblicazione
const $releaseDate = $("#__next > main > div > section.ipc-page-background.ipc-page-background--base.TitlePage__StyledPageBackground-wzlr49;
const dataRilascio = $releaseDate.text();

//prelevo la valutazione/rating
const $rating = $("#__next > main > div > section.ipc-page-background.ipc-page-background--base.TitlePage__StyledPageBackground-wzlr49;
const valutazione = $rating.text();

//prelevo l'immagine di copertina
const $poster = $("#__next > main > div > section.ipc-page-background.ipc-page-background--base.TitlePage__StyledPageBackground-wzlr49;
const copertina = $poster.attr('src');

//prelevo la trama
const $summary = $("#__next > main > div > section.ipc-page-background.ipc-page-background--base.TitlePage__StyledPageBackground-wzlr49;
const trama = $summary.text();

//prelevo lo scrittore
const $writer = $("#__next > main > div > section.ipc-page-background.ipc-page-background--base.TitlePage__StyledPageBackground-wzlr49;
const scrittore = $writer.text();

//prelevo il o i registi
const $director = $("#__next > main > div > section.ipc-page-background.ipc-page-background--base.TitlePage__StyledPageBackground-wzlr49;
const regista = $director.text();
```

In fine viene creato un oggetto di ritorno che contiene tutte le proprietà che sono state estratte e ritornate al richiedente.

Sono stati aggiunti due oggetti; “cache” e “movieCache” per restituire all’utente delle risposte in maniera fulminea. Quando viene fatta una richiesta, il “searchTerm” o il “movieID” vengono messi all’interno di questi oggetti così la prossima volta che si avrà la stessa richiesta, al posto di fare lo scrape di tutta la pagina un’altra volta, viene restituito il valore immagazzinato nella cache.

```
giammrcogaudini@AirdiGiammarco server-scraper % npm run dev
> progettoreticalcolatori@1.0.0 dev
> nodemon index.js

[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): ***!
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Server in ascolto sulla porta 3000
Dati presi dalla cahe avengers
□
```

Test e usabilità

- Test della funzione “searchMovie”.

Pagina su cui si vuole eseguire lo scraping: “searchTerm” = Avengers

The screenshot shows the IMDb search results page for the term "avengers". The main search bar at the top contains "Search IMDb" and the search term "avengers". Below the search bar, there are two sections: "Category Search" and "Advanced Search". The "Category Search" section lists various categories such as All, Name, Title, Movie, TV, TV Episode, Video Game, Company, Keyword, Plot Summaries, Biographies, and Quotes. The "Advanced Search" section allows creating an extremely specific search with options like Advanced Title Search, Advanced Name Search, Collaborations, and Topics. The main content area displays 200 results for "avengers", categorized under "Titles". Each result includes a thumbnail image, the title, and the year it was released. Some titles are followed by a brief description or alternative names in parentheses.

Rank	Title	Year
1	The Avengers (2012)	2012
2	Marvel's Avengers (2020) (Video Game)	2020
3	Avengers (1999) (TV Series)	1999
4	Avengers: Endgame (2019)	2019
5	Avengers: Infinity War (2018)	2018
6	Avengers: Age of Ultron (2015)	2015
7	Avengers Assemble (2012) (TV Series)	2012
8	Avengers - I più potenti eroi della Terra (2010) (TV Series)	2010
9	Agente speciale (1961) (TV Series) aka "The Avengers"	1961
10	The Avengers - Agenti speciali (1998)	1998
11	Tokyo Revengers (2021) (TV Series)	2021

Risultato:

```
1 // 20211114234337
2 // http://localhost:3000/search/avengers
3
4 [
5   {
6     "titolo": "The Avengers",
7     "poster": "https://m.media-
amazon.com/images/M/MVSBNDYxNjQyMjAtNTdiOS00ONGYwLWFmNTAtNThmYjUSZGI2YTI1XkEyXkFqcGdeQXVyMTMx0Dk20TU@._V1_UX32_CR0,0,32,44_AL_.jpg",
8     "movieID": "tt0848228"
9   },
10  {
11    "titolo": "Novgorodtsy",
12    "poster": "https://m.media-
amazon.com/images/M/MVSBYTk1NGUxNjItODvh0S00MDkyLWI3YjgtZTdhYzgzMDc5YTJmXkEyXkFqcGdeQXVyNzU3NjUzNTc@._V1_UX32_CR0,0,32,44_AL_.jpg",
13    "movieID": "tt0036220"
14  },
15  {
16    "titolo": "Fu chou zhe",
17    "poster": "https://m.media-
amazon.com/images/M/MVSBY2jYmJiYjYt2MwMS00Y2IwLWE3ZjAtNGM2MDUzMzM4NDZmXkEyXkFqcGdeQXVyNTM3MDMyMDQ@._V1_UY44_CR0,0,32,44_AL_.jpg",
18    "movieID": "tt0164450"
19  },
20  {
21    "titolo": "Avengers: Endgame",
22    "poster": "https://m.media-amazon.com/images/M/MV5BMTc5MDE20DcwNV5BMl5BanBnXkFtZTgwMzI2NzQ2NzM@._V1_UX32_CR0,0,32,44_AL_.jpg",
23    "movieID": "tt4154796"
24  },
25  {
26    "titolo": "Avengers: Infinity War"
```

- Test della funzione “getMovie”.
- Pagina su cui si vuole eseguire lo scraping: “movieID” = tt0848228

Risultato:

```

1 // 20211115000209
2 // http://localhost:3000/movie/tt0848228
3
4 {
5   "movieID": "tt0848228",
6   "titolo": "The Avengers",
7   "durata": "2h 23m",
8   "generi": "ActionAdventureSci-Fi",
9   "dataRilascio": "2012",
10  "valutazione": "8.0",
11  "copertina": "https://m.media-amazon.com/images/M/MV5BMjc1NWQ4ZDEtNGU2Ni00MWIwLWExDgtNzA4YmUwNGQ1ZDFiXkEyXkFqcGdeQXVyMTYzMMDM0NTU@.\_V1\_QL75\_UX190\_CR0,0,190,281\_.jpg",
12  "trama": "Earth's mightiest heroes must come together and learn to fight as a team if they are going to stop the mischievous Loki and his alien army from enslaving humanity.",
13  "scrittore": "Joss Whedon(screenplay) (story)Zak Penn(story)",
14  "regista": "Joss Whedon",
15  "attori": "Robert Downey Jr.Tony Stark...Chris EvansSteve Rogers...Scarlett JohanssonNatasha Romanoff...Natasha Romanoff...Jeremy RennerClint Barton...Mark RuffaloBruce Banner...Chris HemsworthThor...HiddlestonLokias LokiClark GreggAgent Phil Coulson...Agent Phil CoulsonCobie SmuldersAgent Maria Hill...Agent Maria HillStellan SkarsgårdSelvigas SelvigSamuel L. JacksonNick Furyas Nick FuryGwyneth PaltrowPepper PottsPaul BettanyJarvisas Jarvis(voice)Alexis DenisofThe Otheras The OtherTina BenkoNASA ScientistJerzy SkolimowskiGeorgi Luchkovas Georgi LuchkovKirill NikiforovWeaselly Thugas Weaselly ThugJeff WolfeTall Thugas Tall Thug",
16  "budget": "$220,000,000 (estimated)",
17  "incassi": "$1,518,815,515",
18  "trailer": "https://www.imdb.com/video/vi1891149081?playlistId=tt0848228&ref\_=tt\_ov\_vi"
19 }

```

Il deployment dell'applicazione è stato svolto su heroku.

3. SCRAPING CON PUPPETEER

La seconda parte del progetto è relativa allo scraping attraverso il tool Puppeteer.

Puppeteer è una libreria di Node.js che fornisce un'API di alto livello per controllare Chrome e Chromium in modalità headless tramite il protocollo DevTools. La modalità headless è una modalità di esecuzione alternativa di Chrome priva di interfaccia grafica: stessa capacità di esecuzione del browser, ma maggiore velocità e stabilità data dall'assenza di UI. Puppeteer può essere configurato per usare Chrome e Chromium anche in modalità non-headless.

È composta da tre directory dedicate rispettivamente a tre siti diversi. Gli script al loro interno sono molto simili (ridondanti), variano per l'url della pagina web che dovrà essere aperta dal browser e da alcuni campi che devono essere estrapolati:

1. imdbPuppeteer:

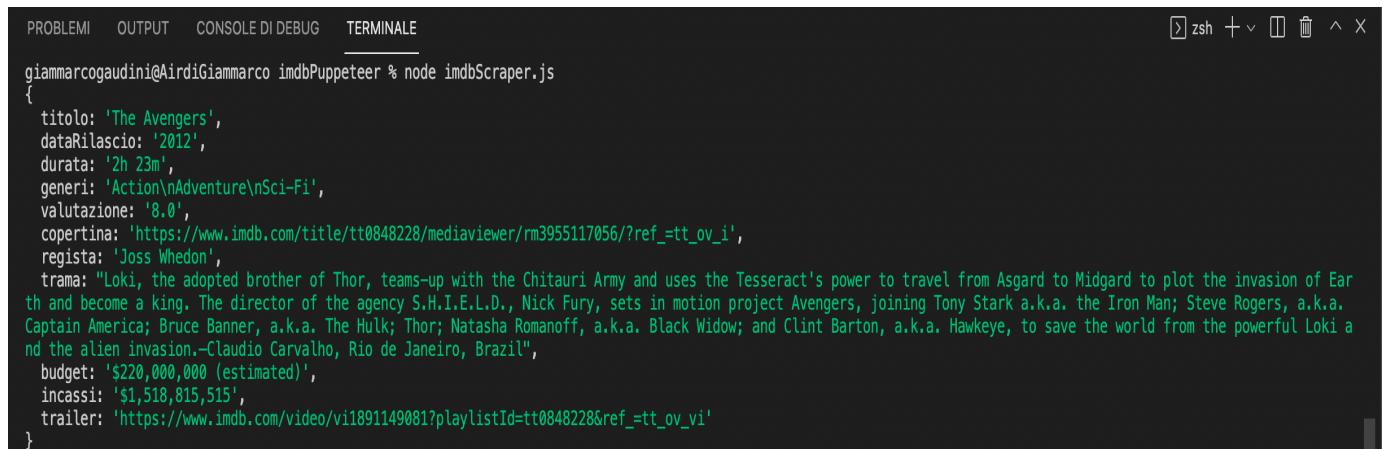
E' composto dallo script `imdbScraper.js`. In questo script, viene importata la libreria `Puppeteer`, e dichiarata una variabile "url" che contiene l'indirizzo della pagina web da raschiare.

Viene inizializzato il browser, cioè attraverso la funzione ".lounch()" viene lanciato Chromium. A seguire viene aperta una nuova pagina con il comando "browser.newPage()" e di conseguenza con l'istruzione "page.goto(movieUrl)" indichiamo che la nuova pagina che si aprirà deve aprire il contenuto della variabile `movieUrl`.

Una volta aperta la pagina web con la funzione "page.evaluate()" viene analizzato tutto il contenuto di essa tramite i `querySelector`.

Alla fine dopo avere ritornato l'oggetto da stampare mediante la funzione "browser.close()", il browser viene chiuso.

Esempio del funzionamento del file `imdbScraper.js` con il film `Avengers`.



```
PROBLEMI OUTPUT CONSOLE DI DEBUG TERMINALE zsh + ×
giammarcogaudini@AirdiGiammarco ~ % node imdbScraper.js
{
  titolo: 'The Avengers',
  dataRilascio: '2012',
  durata: '2h 23m',
  generi: 'Action\\nAdventure\\nSci-Fi',
  valutazione: '8,0',
  copertina: 'https://www.imdb.com/title/tt0848228/mediaviewer/rm3955117056/?ref_=tt_ov_i',
  regista: 'Joss Whedon',
  trama: "Loki, the adopted brother of Thor, teams-up with the Chitauri Army and uses the Tesseract's power to travel from Asgard to Midgard to plot the invasion of Earth and become a king. The director of the agency S.H.I.E.L.D., Nick Fury, sets in motion project Avengers, joining Tony Stark a.k.a. the Iron Man; Steve Rogers, a.k.a. Captain America; Bruce Banner, a.k.a. The Hulk; Thor; Natasha Romanoff, a.k.a. Black Widow; and Clint Barton, a.k.a. Hawkeye, to save the world from the powerful Loki and the alien invasion.-Claudio Carvalho, Rio de Janeiro, Brazil",
  budget: '$220,000,000 (estimated)',
  incassi: '$1,518,815,515',
  trailer: 'https://www.imdb.com/video/vi1891149081?playlistId=tt0848228&ref_=tt_ov_vi'
}
```

2. tmdbPuppeteer:

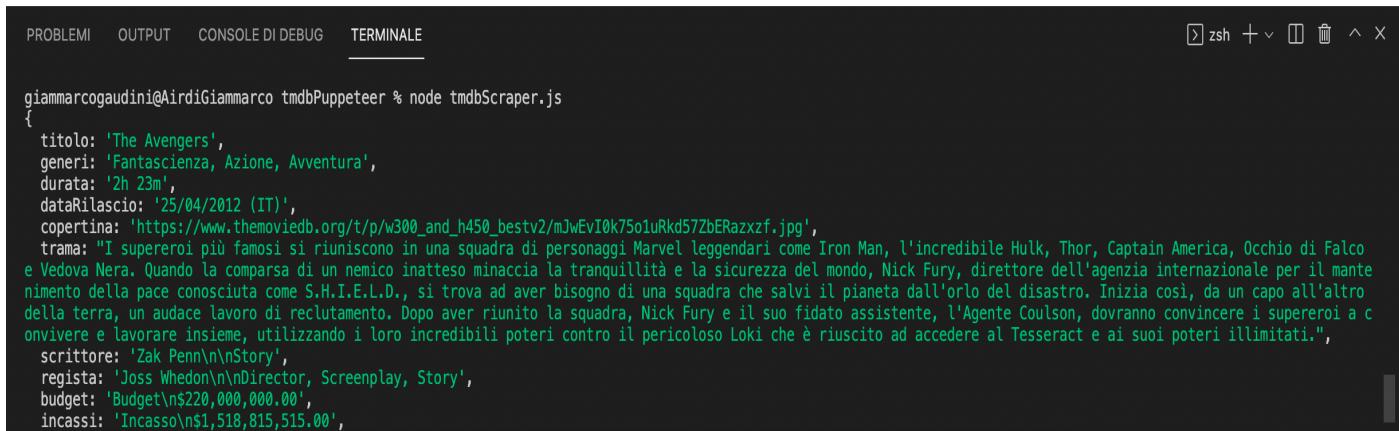
E' composto dallo script tmdbScraper.js., In questo script, viene importata la libreria Puppeteer, e dichiarata una variabile "url" che contiene l'indirizzo della pagina web da raschiare.

Viene inizializzato il browser, cioè attraverso la funzione ".lounch()" viene lanciato Chromium. A seguire viene aperta una nuova pagina web con il comando "browser.newPage()" e di conseguenza con l'istruzione "page.goto(movieUrl)" indichiamo che la nuova pagina che si aprirà deve aprire il contenuto della variabile movieUrl.

Una volta aperta la pagina web con la funzione "page.evaluate()" viene analizzato tutto il contenuto di essa tramite i querySelector.

Alla fine dopo avere ritornato l'oggetto da stampare mediante la funzione "browser.close.close()", il browser viene chiuso.

Esempio del funzionamento del file tmdbScraper.js con il film Avengers.



```
PROBLEMI OUTPUT CONSOLE DI DEBUG TERMINALE zsh + ↻ ⌫ ^ X

giammarcogaudini@AirdiGiammarco tmdbPuppeteer % node tmdbScraper.js
{
  titolo: 'The Avengers',
  generi: 'Fantascienza, Azione, Avventura',
  durata: '2h 23m',
  dataRilascio: '25/04/2012 (IT)',
  copertina: 'https://www.themoviedb.org/t/p/w300_and_h450_bestv2/mJwEvI0k75o1uRkd57ZbERazxzf.jpg',
  trama: "I supereroi più famosi si riuniscono in una squadra di personaggi Marvel leggendari come Iron Man, L'incredibile Hulk, Thor, Captain America, Occhio di Falco e Vedova Nera. Quando la comparsa di un nemico inatteso minaccia la tranquillità e la sicurezza del mondo, Nick Fury, direttore dell'agenzia internazionale per il mantenimento della pace conosciuta come S.H.I.E.L.D., si trova ad aver bisogno di una squadra che salvi il pianeta dall'orlo del disastro. Inizia così, da un capo all'altro della terra, un audace lavoro di reclutamento. Dopo aver riunito la squadra, Nick Fury e il suo fidato assistente, l'Agente Coulson, dovranno convincere i supereroi a convivere e lavorare insieme, utilizzando i loro incredibili poteri contro il pericoloso Loki che è riuscito ad accedere al Tesseract e ai suoi poteri illimitati.",
  scrittore: 'Zak Penn\n\nStory',
  regista: 'Joss Whedon\n\nDirector, Screenplay, Story',
  budget: 'Budget\n$220,000,000.00',
  incassi: 'Incasso\n$1,518,815,515.00',
```

3. rottenPuppeteer:

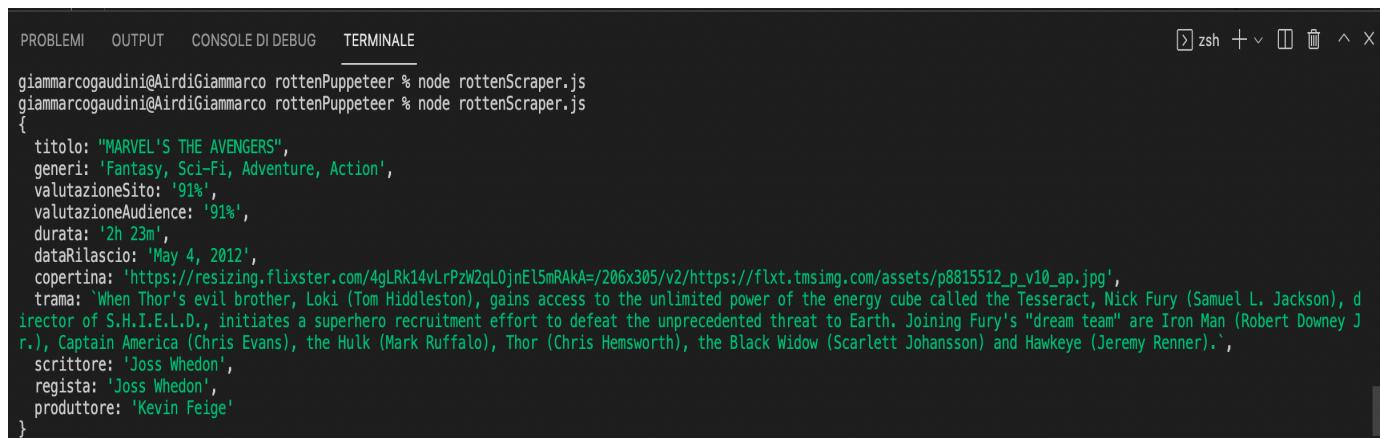
E' composto dallo script rottenScraper.js., In questo script, viene importata la libreria Puppeteer, e dichiarata una variabile "url" che contiene l'indirizzo della pagina web da raschiare.

Viene inizializzato il browser, cioè attraverso la funzione ".lounch()" viene lanciato Chromium. A seguire viene aperta una nuova pagina web con il comando "browser.newPage()" e di conseguenza con l'istruzione "page.goto(movieUrl)" indichiamo che la nuova pagina che si aprirà deve aprire il contenuto della variabile movieUrl.

Una volta aperta la pagina web con la funzione "page.evaluate()" viene analizzato tutto il contenuto di essa tramite i querySelector.

Alla fine dopo avere ritornato l'oggetto da stampare mediante la funzione "browser.close.close()", il browser viene chiuso.

Esempio del funzionamento del file rottenScraper.js con il film Avengers.



```
PROBLEMI OUTPUT CONSOLE DI DEBUG TERMINALE
giammarcogaudini@AirdiGiammarco rottenPuppeteer % node rottenScraper.js
giammarcogaudini@AirdiGiammarco rottenPuppeteer % node rottenScraper.js
{
  titolo: "MARVEL'S THE AVENGERS",
  generi: 'Fantasy, Sci-Fi, Adventure, Action',
  valutazioneSito: '91%',
  valutazioneAudience: '91%',
  durata: '2h 23m',
  dataRilascio: 'May 4, 2012',
  copertina: 'https://resizing.flixster.com/4gLRk14vLrPzW2qLojnEl5mRAkA=/206x305/v2/https://flxt.tmsimg.com/assets/p8815512_p_v10_ap.jpg',
  trama: 'When Thor\'s evil brother, Loki (Tom Hiddleston), gains access to the unlimited power of the energy cube called the Tesseract, Nick Fury (Samuel L. Jackson), director of S.H.I.E.L.D., initiates a superhero recruitment effort to defeat the unprecedented threat to Earth. Joining Fury\'s "dream team" are Iron Man (Robert Downey Jr.), Captain America (Chris Evans), the Hulk (Mark Ruffalo), Thor (Chris Hemsworth), the Black Widow (Scarlett Johansson) and Hawkeye (Jeremy Renner).',
  scrittore: 'Joss Whedon',
  regista: 'Joss Whedon',
  produttore: 'Kevin Feige'
```

4. CONCLUSIONI

In conclusione, nella prima parte si è voluta porre l'attenzione nello scraping attraverso l'utilizzo del tool Cheerio, mentre nella seconda, attraverso il tool Puppeteer e possiamo notare notevoli differenze tra questi tools.

Cheerio è un modulo di Node.js la cui implementazione si basa sul core Jquery, funziona con un modello DOM (document object model), letteralmente modello a oggetti del documento, è una forma di rappresentazione dei documenti strutturati come modello orientato agli oggetti. Rispetto a Puppeteer è abbastanza veloce in quanto è basato su Jquery. Cheerio inoltre è in grado di analizzare qualsiasi tipo di documento HTML e XML.

Puppeteer invece, è utilizzato per automatizzare le attività del browser e può funzionare solo con il browser headless di Google Chrome come Chromium. Oltre ad essere utilizzato per attività di scraping ha molte funzionalità aggiuntive che non sono presenti su Cheerio come ad esempio la possibilità di fare screenshot e salvare file pdf.