

(/)



0x14. JavaScript - Web scraping

Scripting

API

JavaScript

👤 By: Guillaume, CTO at Holberton School

⚙️ Weight: 1

📅 Project over - took place from Apr 25, 2023 4:00 AM to Apr 26, 2023 4:00 AM

✅ An auto review will be launched at the deadline

In a nutshell...

- **Auto QA review:** 67.0/67 mandatory & 20.0/20 optional
- **Altogether: 200.0%**
 - Mandatory: 100.0%
 - Optional: 100.0%
 - Calculation: $100.0\% + (100.0\% * 100.0\%) == 200.0\%$

Resources

Read or watch:

- Working with JSON data (/rltoken/ONv-sSv-FA87Mc5rMZmO6A)
- The workflow of accessing the attributes of a simply-created JSON object by Jimmy Tran from Cohort 1 - San Francisco (/rltoken/zm0h7FqpQCZZpPZqxxwLxA)
- request module (/rltoken/goymbxGy-cTc5ZdKBTUcTQ)
- Modern JS (/rltoken/j2PStAUtVPdXKwrrFxpt0g)

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone (/rltoken/yZIL5HK-2hHAP-RJF6yInQ), **without the help of Google**:

General

- Why JavaScript programming is amazing
- How to manipulate JSON data
- How to use `request` and `fetch` API
- How to read and write a file using `fs` module



Copyright - Plagiarism

- You are tasked to come up with solutions for the tasks below yourself to meet with the above learning objectives.
- You will not be able to meet the objectives of this or any following project by copying and pasting someone else's work.
- You are not allowed to publish any content of this project.
- Any form of plagiarism is strictly forbidden and will result in removal from the program.

Requirements

General

- Allowed editors: `vi`, `vim`, `emacs`
- All your files will be interpreted on Ubuntu 14.04 LTS using `node` (version 10.14.x)
- All your files should end with a new line
- The first line of all your files should be exactly `#!/usr/bin/node`
- A `README.md` file, at the root of the folder of the project, is mandatory
- Your code should be `semistandard` compliant. Rules of Standard (/rltoken/W9rASrTqkF-xXjcwomrMLw) + semicolons on top (/rltoken/GXh9DyGGivUB7pdq9Oqmzg). Also as reference: AirBnB style (/rltoken/NZR55f9vk1dZXj5q7UI5mQ)
- All your files must be executable
- The length of your files will be tested using `wc`
- You are not allowed to use `var`

More Info

Install Node 10

```
$ curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -  
$ sudo apt-get install -y nodejs
```

Install semi-standard

Documentation (/rltoken/GXh9DyGGivUB7pdq9Oqmzg)

```
$ sudo npm install semistandard --global
```

Install request module and use it

Documentation (/rltoken/goymbxGy-cTc5ZdKBTUcTQ)

```
$ sudo npm install request --global  
$ export NODE_PATH=/usr/lib/node_modules
```

Notes: Request module has been deprecated since February 2020 - the team is considering alternative to replace this module - however, it's a really simple and powerful module for practicing web-scraping in JavaScript (and still used a lot in the industry).



Tasks

0. Readme

mandatory

Score: 100.0% (Checks completed: 100.0%)

Write a script that reads and prints the content of a file.

- The first argument is the file path
- The content of the file must be read in `utf-8`
- If an error occurred during the reading, print the error object

```
guillaume@ubuntu:~/0x14$ cat cisfun
C is super fun!
guillaume@ubuntu:~/0x14$ ./0-readme.js cisfun
C is super fun!

guillaume@ubuntu:~/0x14$ ./0-readme.js doesntexist
{ Error: ENOENT: no such file or directory, open 'doesntexist'
  at Error (native)
  errno: -2,
  code: 'ENOENT',
  syscall: 'open',
  path: 'doesntexist' }
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x14-javascript-web_scraping`
- File: `0-readme.js`

☒ Done!

Help

Check your code

>_ Get a sandbox

QA Review

1. Write me

mandatory

Score: 100.0% (Checks completed: 100.0%)

Write a script that writes a string to a file.

- The first argument is the file path
- The second argument is the string to write
- The content of the file must be written in `utf-8`
- If an error occurred during while writing, print the error object



```
guillaume@ubuntu:~/0x14$ ./1-writeme.js my_file.txt "Python is cool"
Python is cool
guillaume@ubuntu:~/0x14$ cat my_file.txt ; echo ""
```

Repo:

- GitHub repository: alx-higher_level_programming
- Directory: 0x14-javascript-web_scraping
- File: 1-writeme.js

☒ Done!

Help

Check your code

>_ Get a sandbox

QA Review

2. Status code

mandatory

Score: 100.0% (Checks completed: 100.0%)

Write a script that display the status code of a GET request.

- The first argument is the URL to request (GET)
- The status code must be printed like this: code: <status code>
- You must use the module request

```
guillaume@ubuntu:~/0x14$ ./2-statuscode.js https://alx-intranet.hbtn.io/status
code: 200
guillaume@ubuntu:~/0x14$ ./2-statuscode.js https://alx-intranet.hbtn.io/doesnt_exist
code: 404
guillaume@ubuntu:~/0x14$
```

Repo:

- GitHub repository: alx-higher_level_programming
- Directory: 0x14-javascript-web_scraping
- File: 2-statuscode.js

☒ Done!

Help

Check your code

>_ Get a sandbox

QA Review

3. Star wars movie title

mandatory

Score: 100.0% (Checks completed: 100.0%)

Write a script that prints the title of a Star Wars movie where the episode number matches a given integer.

- The first argument is the movie ID



- You must use the Star wars API (/rltoken/HwLU2L7tJ4TEjzFTBc7zTA) with the endpoint (/) `https://swapi-api.alx-tools.com/api/films/:id`
- You must use the module `request`

```
guillaume@ubuntu:~/0x14$ ./3-starwars_title.js 1
A New Hope
guillaume@ubuntu:~/0x14$ ./3-starwars_title.js 5
Attack of the Clones
guillaume@ubuntu:~/0x14$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x14-javascript-web_scraping`
- File: `3-starwars_title.js`

☒ Done![Help](#)[Check your code](#)[>_ Get a sandbox](#)[QA Review](#)

4. Star wars Wedge Antilles

mandatory

Score: 100.0% (Checks completed: 100.0%)

Write a script that prints the number of movies where the character "Wedge Antilles" is present.

- The first argument is the API URL of the Star wars API (/rltoken/HwLU2L7tJ4TEjzFTBc7zTA): `https://swapi-api.alx-tools.com/api/films/`
- Wedge Antilles is character ID `18` - your script **must** use this ID for filtering the result of the API
- You must use the module `request`

```
guillaume@ubuntu:~/0x14$ ./4-starwars_count.js https://swapi-api.alx-tools.com/api/films
3
guillaume@ubuntu:~/0x14$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x14-javascript-web_scraping`
- File: `4-starwars_count.js`

☒ Done![Help](#)[Check your code](#)[>_ Get a sandbox](#)[QA Review](#)

5. Loripsum

mandatory

Score: 100.0% (Checks completed: 100.0%)



Write a script that gets the contents of a webpage and stores it in a file.

(/)

- The first argument is the URL to request
- The second argument the file path to store the body response
- The file must be UTF-8 encoded
- You must use the module `request`

```
guillaume@ubuntu:~/0x14$ ./5-request_store.js http://loripsum.net/api loripsum
```

```
guillaume@ubuntu:~/0x14$ cat loripsum
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Haec quo modo convenient, non sane intellego. Nam memini etiam quae nolo, oblivisci non possum quae volo. Te enim iudicem aequum puto, modo quae dicat ille bene noris. Terram, mihi crede, ea lanx et maria deprimet. Deinde prima illa, quae in congressu solemus: Quid tu, inquit, huc? Hoc etsi modis ultimis reprehendi potest, tamen accipio, quod dant. </p>
```

```
<p>Ad eos igitur converte te, quaeso. Pudebit te, inquam, illius tabulae, quam Cleanthes sane commode verbis depingere solebat. Sic enim censent, oportunitatis esse beate vivere. Quo studio Aristophanem putamus aetatem in litteris duxisse? Aequae enim contingit omnibus fidibus, ut incontentae sint. Ut aliquid scire se gaudeant? Qui enim existimabit posse se miserum esse beatus non erit. Putabam equidem satis, inquit, me dixisse. </p>
```

```
<p>Duo Reges: constructio interrete. Quid ei reliquisti, nisi te, quoquo modo loqueretur, intellegere, quid diceret? Quis animo aequo videt eum, quem impure ac flagitiose putat vivere? Illud non continuo, ut aeque incontentae. Illa videamus, quae a te de amicitia dicta sunt. At ille pellit, qui permulcet sensum voluptate. Tamen aberramus a proposito, et, ne longius, prorsus, inquam, Piso, si ista mala sunt, placet. </p>
```

```
<p>Non enim, si omnia non sequebatur, idcirco non erat ortus illinc. Nos cum te, M. Quem si tenueris, non modo meum Ciceronem, sed etiam me ipsum abducas licebit. Apparet statim, quae sint officia, quae actiones. Ergo instituto veterum, quo etiam Stoici utuntur, hinc capiamus exordium. Eadem nunc mea adversum te oratio est. Quid, si etiam iucunda memoria est praeteritorum malorum? Hoc enim constituto in philosophia constituta sunt omnia. </p>
```

```
guillaume@ubuntu:~/0x14$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x14-javascript-web_scraping`
- File: `5-request_store.js`

☒ Done!

Help

Check your code

>_ Get a sandbox

QA Review

6. How many completed?

mandatory

Score: 100.0% (Checks completed: 100.0%)

Write a script that computes the number of tasks completed by user id.

- The first argument is the API URL: `https://jsonplaceholder.typicode.com/todos`



- Only print users with completed task
- (/)
- You must use the module `request`

```
guillaume@ubuntu:~/0x14$ ./6-completed_tasks.js https://jsonplaceholder.typicode.com/todos
{ '1': 11,
  '2': 8,
  '3': 7,
  '4': 6,
  '5': 12,
  '6': 6,
  '7': 9,
  '8': 11,
  '9': 8,
  '10': 12 }
```

guillaume@ubuntu:~/0x14\$

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x14-javascript-web_scraping`
- File: `6-completed_tasks.js`

☒ Done![Help](#)[Check your code](#)[>_ Get a sandbox](#)[QA Review](#)

7. Who was playing in this movie?

#advanced

Score: 100.0% (Checks completed: 100.0%)

Write a script that prints all characters of a Star Wars movie:

- The first argument is the Movie ID - example: `3` = "Return of the Jedi"
- Display one character name by line
- You must use the Star wars API (`/rltoken/HwLU2L7tJ4TEjzfTBc7zTA`)
- You must use the module `request`



```
guillaume@ubuntu:~/0x14$ ./100-starwars_characters.js 3
Darth Vader
R2-D2
Luke Skywalker
Han Solo
Leia Organa
Chewbacca
Palpatine
Obi-Wan Kenobi
Jabba Desilijic Tiure
Wedge Antilles
Yoda
Boba Fett
Ackbar
Arvel Crynyd
Mon Mothma
Nien Nunb
Wicket Systri Warrick
Bib Fortuna
C-3PO
Lando Calrissian
guillaume@ubuntu:~/0x14$
```

Repo:

- GitHub repository: alx-higher_level_programming
- Directory: 0x14-javascript-web_scraping
- File: 100-starwars_characters.js

☒ Done![Help](#)[Check your code](#)[>_ Get a sandbox](#)[QA Review](#)**8. Right order**

#advanced

Score: 100.0% (Checks completed: 100.0%)

Write a script that prints all characters of a Star Wars movie:

- The first argument is the Movie ID - example: 3 = "Return of the Jedi"
- Display one character name by line **in the same order of the list "characters" in the /films/ response**
- You must use the Star wars API (/rltoken/HwLU2L7tJ4TEjzFTBc7zTA)
- You must use the module request




```
guillaume@ubuntu:~/0x14$ ./101-starwars_characters.js 3
7)
Luke Skywalker
C-3PO
R2-D2
Darth Vader
Leia Organa
Obi-Wan Kenobi
Chewbacca
Han Solo
Jabba Desilijic Tiure
Wedge Antilles
Yoda
Palpatine
Boba Fett
Lando Calrissian
Ackbar
Mon Mothma
Arvel Crynyd
Wicket Systri Warrick
Nien Nunb
Bib Fortuna
guillaume@ubuntu:~/0x14$
```

Repo:

- GitHub repository: alx-higher_level_programming
- Directory: 0x14-javascript-web_scraping
- File: 101-starwars_characters.js

☒ Done!

Help

Check your code

>_ Get a sandbox

QA Review

