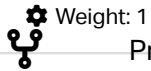🏠 Home(/)
0x03. ES6 data manipulation

JavaScript ES6 My Planning(/planning/me)

⚙ Weight: 1

⑂ Projects(/projects/current)

📅 Project over - took place from Jul 1, 2024 6:00 AM to Jul 3, 2024 6:00 AM

☑ An auto QA Review(/correctionst/to_review) will be launched at the deadline

? Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

In a nutshell…

- **Auto QA review:** 37.0/39 mandatory & 3.0/4 optional
- **Altogether: 166.02%**
  - Mandatory: 94.87%
  - Optional: 75.0%
  - Calculation: 94.87% + (94.87% * 75.0%) == **166.02%**

🎓 Curriculums(/dashboards/my_curriculums)

📄 Concepts(/concepts)

💬 Conference rooms(/dashboards/video_rooms)

🗄 Servers(/servers)

>_ Sandboxes(/user_containers/current)

🔧 Tools(/dashboards/my_tools)

🎞 Video on demand(/dashboards/videos)

👥 Peers(/users/peers)

👾 Discord(https://discord.com/app)

Resources

Read or watch:

- Array (/rltoken/bcXqK1IalHtrZ45sv0RxsQ)
- Typed Array (/rltoken/YZ5RtzAPTaWtF00MYbXuVw)
- Set Data Structure (/rltoken/Ch8vq39y9QnlTMr8CymgEg)

👤 My Profile(/users/my_profile)
- Map Data Structure (/rltoken/W29MV3f8li4HmeJSALNlpw)

🔍

💬

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone (rltoken/vFyWo9TJ_4ypOC6uPi2Iow), **without the help of Google**:

- How to use map, filter and reduce on arrays
- Typed Arrays
- The Set, Map, and Weak link data structures

Requirements

- All your files will be executed on Ubuntu 18.04 LTS using NodeJS 12.11.x

- Allowed editors: `vi`, `vim`, `emacs`, `Visual Studio Code`
- All your files should end with a new line
- A `README.md` file at the root of the folder of the project is mandatory

- Your code should use the `js` extension
- Your code will be tested using `Jest` and the command `npm run test`
- Your code will be verified against lint using ESLint
- Your code needs to pass all the tests and lint. You can verify the entire project running `npm run full-test`

- All of your functions must be exported

Setup

Install NodeJS 12.11.x

In your home directory):

```
curl -sL https://deb.nodesource.com/setup_12.x -o nodesource_setup.sh
sudo bash nodesource_setup.sh
sudo apt install nodejs -y
```

```
$ nodejs -v
v12.11.1
$ npm -v
6.11.3
```

Install Jest, Babel, and ESLint

In your project directory, install Jest, Babel and ESList by using the supplied `package.json` and run `npm install.`

Configuration files

Add the following files to your project directory

package.json

Click to show/hide file contents

babel.config.js

Click to show/hide file contents

.eslintrc.js

Click to show/hide file contents

and...

Don't forget to run $ `npm install` when you have the `package.json`

Tasks

0. Basic list of objects   mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Create a function named `getListStudents` that returns an array of objects.

Each object should have three attributes: `id` (Number), `firstName` (String), and `location` (String).

The array contains the following students in order:

- `Guillaume`, id: `1`, in `San Francisco`
- `James`, id: `2`, in `Columbia`
- `Serena`, id: `5`, in `San Francisco`

```
bob@dylan:~$ cat 0-main.js
import getListStudents from "./0-get_list_students.js";

console.log(getListStudents());
```

```
bob@dylan:~$
bob@dylan:~$ npm run dev 0-main.js
[
```
```
  { id: 1, firstName: 'Guillaume', location: 'San Francisco' },
  { id: 2, firstName: 'James', location: 'Columbia' },
  { id: 5, firstName: 'Serena', location: 'San Francisco' }
]
```
```
bob@dylan:~$
```

Repo:

- GitHub repository: `alx-backend-javascript`

- Directory: `0x03-ES6_data_manipulation`
- File: `0-get_list_students.js`

Check submission >_ Get a sandbox View results

1. More mapping

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Create a function `getListStudentIds` that returns an array of ids from a list of object.

This function is taking one argument which is an array of objects - and this array is the same format as

`getListStudents` from the previous task.

If the argument is not an array, the function is returning an empty array.

You must use the `map` function on the array.

```
bob@dylan:~$ cat 1-main.js
import getListStudentIds from "./1-get_list_student_ids.js";
import getListStudents from "./0-get_list_students.js";

console.log(getListStudentIds("hello"));
console.log(getListStudentIds(getListStudents()));

bob@dylan:~$
bob@dylan:~$ npm run dev 1-main.js
[]
[ 1, 2, 5 ]
bob@dylan:~$
```

Repo:

- GitHub repository: `alx-backend-javascript`
- Directory: `0x03-ES6_data_manipulation`

- File: `1-get_list_student_ids.js`

Check submission >_ Get a sandbox View results

2. Filter

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Create a function `getStudentsByLocation` that returns an array of objects who are located in a specific city.

It should accept a list of students (from `getListStudents`) and a `city` (string) as parameters.

You must use the `filter` function on the array.

```
bob@dylan:~$ cat 2-main.js
import getListStudents from "./0-get_list_students.js";
import getStudentsByLocation from "./2-get_students_by_loc.js";

const students = getListStudents();

console.log(getStudentsByLocation(students, 'San Francisco'));

bob@dylan:~$
bob@dylan:~$ npm run dev 2-main.js
[
  { id: 1, firstName: 'Guillaume', location: 'San Francisco' },
  { id: 5, firstName: 'Serena', location: 'San Francisco' }
]
bob@dylan:~$
```

Repo:

- GitHub repository: `alx-backend-javascript`
- Directory: `0x03-ES6_data_manipulation`
- File: `2-get_students_by_loc.js`

[Check submission] [>_ Get a sandbox] [View results]

3. Reduce

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Create a function `getStudentIdsSum` that returns the sum of all the student ids.

It should accept a list of students (from `getListStudents`) as a parameter.

You must use the `reduce` function on the array.

```
bob@dylan:~$ cat 3-main.js
import getListStudents from "./0-get_list_students.js";
import getStudentIdsSum from "./3-get_ids_sum.js";

const students = getListStudents();
const value = getStudentIdsSum(students);

console.log(value);

bob@dylan:~$
bob@dylan:~$ npm run dev 3-main.js
8
bob@dylan:~$
```

Repo:

Home(/)

Check submission   >_ Get a sandbox   View results

My Planning(/planning/me)

4. Combine

Projects(/projects/current)

Score: 100.0% (*Checks completed: 100.0%*)

QA Reviews I can make(/corrections/to_review)

Create a function `updateStudentGradeByCity` that returns an array of students for a specific city with their new grade

Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

It should accept a list of students (from `getListStudents`), a `city` (String), and `newGrades` (Array of "grade" objects) as parameters.

`newGrades` is an array of objects with this format:

Curriculums(/dashboards/my_curriculums)

```
{
  studentId: 31,
  grade: 78,
}
```

Concepts(/concepts)

If a student doesn't have grade in `newGrades`, the final grade should be `N/A`.

Conference recordings(/dashboards/videos_the_final)

You must use `filter` and `map` combined.

Servers(/servers)

>_ Sandboxes(/user_containers/current)

🔧 Tools(/dashboards/my_tools)

Video on demand(/dashboards/videos)

Peers(/users/peers)

Discord(https://discord.com/app)

My Profile(/users/my_profile)

```
bob@dylan:~$ cat 4-main.js
import getListStudents from "./0-get_list_students.js";
import updateStudentGradeByCity from "./4-update_grade_by_city.js";

console.log(updateStudentGradeByCity(getListStudents(), "San Francisco", [{ stude
ntId: 5, grade: 97 }, { studentId: 1, grade: 86 }]));

console.log(updateStudentGradeByCity(getListStudents(), "San Francisco", [{ stude
ntId: 6, grade: 97 }]));

bob@dylan:~$
bob@dylan:~$ npm run dev 4-main.js
[
  {
    id: 1,
    firstName: 'Guillaume',
    location: 'San Francisco',
    grade: 86
  },
  { id: 5, firstName: 'Serena', location: 'San Francisco', grade: 97 }
]
[
  {
    id: 1,
    firstName: 'Guillaume',
    location: 'San Francisco',
    grade: 'N/A'
  },
  { id: 5, firstName: 'Serena', location: 'San Francisco', grade: 97 }
]
bob@dylan:~$
```

Repo:

- GitHub repository: `alx-backend-javascript`
- Directory: `0x03-ES6_data_manipulation`
- File: `4-update_grade_by_city.js`

Check submission | >_ Get a sandbox | View results

5. Typed Arrays

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Create a function named `createInt8TypedArray` that returns a new `ArrayBuffer` with an `Int8` value
at a specific position.

It should accept three arguments: `length` (Number), `position` (Number), and `value` (Number).

If adding the value is not possible the error `Position outside range` should be thrown.

```
bob@dylan:~$ cat 5-main.js
import createInt8TypedArray from "./5-typed_arrays.js";

console.log(createInt8TypedArray(10, 2, 89));

bob@dylan:~$
bob@dylan:~$ npm run dev 5-main.js
DataView {
  byteLength: 10,
  byteOffset: 0,
  buffer: ArrayBuffer {
    [Uint8Contents]: <00 00 59 00 00 00 00 00 00 00>,
    byteLength: 10
  }
}
bob@dylan:~$
```

Repo:

- GitHub repository: `alx-backend-javascript`
- Directory: `0x03-ES6_data_manipulation`

- File: `5-typed_arrays.js`

Check submission   >_ Get a sandbox   View results

6. Set data structure

`mandatory`

Score: 100.0% (Checks completed: 100.0%)

Create a function named `setFromArray` that returns a `Set` from an array.

It accepts an argument (Array, of any kind of element).

```
bob@dylan:~$ cat 6-main.js
import setFromArray from "./6-set.js";

console.log(setFromArray([12, 32, 15, 78, 98, 15]));
```

```
bob@dylan:~$
bob@dylan:~$ npm run dev 6-main.js
Set { 12, 32, 15, 78, 98 }
bob@dylan:~$
```

Repo:

- GitHub repository: `alx-backend-javascript`
- Directory: `0x03-ES6_data_manipulation`
- File: `6-set.js`

7. More set data structure

Score: 100.0% (*Checks completed: 100.0%*)

Create a function named `hasValuesFromArray` that returns a boolean if all the elements in the array exist within the set.

It accepts two arguments: a `set` (Set) and an `array` (Array).

```
bob@dylan:~$ cat 7-main.js
import hasValuesFromArray from "./7-has_array_values.js";

console.log(hasValuesFromArray(new Set([1, 2, 3, 4, 5]), [1]));
console.log(hasValuesFromArray(new Set([1, 2, 3, 4, 5]), [10]));
console.log(hasValuesFromArray(new Set([1, 2, 3, 4, 5]), [1, 10]));

bob@dylan:~$
bob@dylan:~$ npm run dev 7-main.js
true
false
false
bob@dylan:~$
```

Repo:

- GitHub repository: `alx-backend-javascript`
- Directory: `0x03-ES6_data_manipulation`
- File: `7-has_array_values.js`

8. Clean set

mandatory

Score: 60.0% (*Checks completed: 60.0%*)

Create a function named `cleanSet` that returns a string of all the set values that start with a specific string (`startString`).

It accepts two arguments: a `set` (Set) and a `startString` (String).

When a value starts with `startString` you only append the rest of the string. The string contains all the values of the set separated by `-`.

```
bob@dylan:~$ cat 8-main.js
import cleanSet from "./8-clean_set.js";

console.log(cleanSet(new Set(['bonjovi', 'bonaparte', 'bonappetit', 'banana']),
'bon'));
console.log(cleanSet(new Set(['bonjovi', 'bonaparte', 'bonappetit', 'banana']),
''));

bob@dylan:~$
bob@dylan:~$ npm run dev 8-main.js
jovi-aparte-appetit

bob@dylan:~$
```

Repo:

- GitHub repository: `alx-backend-javascript`
- Directory: `0x03-ES6_data_manipulation`
- File: `8-clean_set.js`

Check submission | Mark submission | >_ Get a sandbox | View results

9. Map data structure

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Create a function named `groceriesList` that returns a map of groceries with the following items (name, quantity):

```
Apples, 10
Tomatoes, 10
Pasta, 1
Rice, 1
Banana, 5
```

Result:

```
bob@dylan:~$ cat 9-main.js
import groceriesList from "./9-groceries_list.js";
```

```
console.log(groceriesList());
```

```
bob@dylan:~$
bob@dylan:~$ npm run dev 9-main.js
Map {
```

```
  'Apples' => 10,
  'Tomatoes' => 10,
  'Pasta' => 1,
```

```
  'Rice' => 1,
  'Banana' => 5
}
```

Repo:

- GitHub repository: `alx-backend-javascript`
- Directory: `0x03-ES6_data_manipulation`

- File: `9-groceries_list.js`

Check submission sandbox

Get a sandbox | View results

10. More map data structure

`mandatory`

Score: 100.0% (*Checks completed: 100.0%*)

Create a function named `updateUniqueItems` that returns an updated map for all items with initial quantity at 1.

It should accept a map as an argument. The map it accepts for argument is similar to the map you create in the previous task.

For each entry of the map where the quantity is 1, update the quantity to 100. If updating the quantity is not possible (argument is not a map) the error `Cannot process` should be thrown.

```
bob@dylan:~$ cat 10-main.js
import updateUniqueItems from "./10-update_uniq_items.js";
import groceriesList from "./9-groceries_list.js";

const map = groceriesList();
console.log(map);

updateUniqueItems(map)
console.log(map);

bob@dylan:~$
bob@dylan:~$ npm run dev 10-main.js
Map {
  'Apples' => 10,
  'Tomatoes' => 10,
  'Pasta' => 1,
  'Rice' => 1,
  'Banana' => 5
}
Map {
  'Apples' => 10,
  'Tomatoes' => 10,
  'Pasta' => 100,
  'Rice' => 100,
  'Banana' => 5
}
bob@dylan:~$
```

Repo:

- GitHub repository: `alx-backend-javascript`
- Directory: `0x03-ES6_data_manipulation`
- File: `10-update_uniq_items.js`

Check submission Get a sandbox View results

11. Weak link data structure   `#advanced`

Score: 75.0% (*Checks completed: 75.0%*)

Export a const instance of `WeakMap` and name it `weakMap`.

Export a new function named `queryAPI`. It should accept an endpoint argument like so:

```
{
  protocol: 'http',
  name: 'getUsers',
}
```

Track within the `weakMap`, the number of times `queryAPI` is called for each endpoint.

When the number of queries is >= 5 throw an error with the message `Endpoint load is high`.

```
bob@dylan:~$ cat 100-main.js
import { queryAPI, weakMap } from "./100-weak.js";

const endpoint = { protocol: 'http', name: 'getUsers' };
weakMap.get(endpoint);

queryAPI(endpoint);
console.log(weakMap.get(endpoint));

queryAPI(endpoint);
console.log(weakMap.get(endpoint));

queryAPI(endpoint);
queryAPI(endpoint);
queryAPI(endpoint);
queryAPI(endpoint);

bob@dylan:~$
bob@dylan:~$ npm run dev 100-main.js
1
2

.../100-weak.js:16
    throw new Error('Endpoint load is high');
    ..
bob@dylan:~$
```

Repo:

- GitHub repository: `alx-backend-javascript`
- Directory: `0x03-ES6_data_manipulation`
- File: `100-weak.js`

Check submission   Mark submission   >_ Get a sandbox   View results