

Resources

- Video on demand (/dashboards/videos)
- Promise (/rltoken/8IEJDdrFqrfsXUV9trNmKA)
- JavaScript Promise: An introduction (/rltoken/EnBUklulIIIrOZ3dRJV4LQ)
- Await (/rltoken/SAL0Z-GAD5GVCTnK1iTCdA)
- Async (/rltoken/QZMWLFR29PO2bVOS4_8j5Q)
- Throw (/rltoken/PTXqH5zA1NSVCwCoyr1cNxg)

Learning Objectives

- Promises (how, why, and what)
- How to use the then, resolve, catch methods
- My Profile(/users/my_profile)
- How to use every method of the Promise object





- Throw / Try
- The await operator
- How to use an async function

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
- 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 files below to your project directory



package.json

Peers(/users/peers)

Click to show/hide file contents



babel.config.js

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

Click to show/hide file contents

utils.js

My Profile(/users/my_profile)

Use when you get to tasks requiring uploadPhoto and createUser.



Click to show/hide file contents



.eslintrc.js

Click to show/hide file contents



Home(/)

and...



Don't forget to plan your planning time when you have the package.json

Response Data Format

uploadPhoto returns a response with the format



QA Reviews I can make(/corrections/to_review)



```
{
  status: 200,
  body: 'photo-profile-1'
}
```

Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

createUser returns a response with the format



Curriculums(/dashboards/my_curriculums)



```
{
  firstName: 'Guillaume',
  lastName: 'Salva',
}
```

Concepts(/concepts)



Conference rooms(/dashboards/video_rooms)

Tasks

Servers(/servers)



0. Keep every promise you make and only make promises you can keep

mandatory

Sandboxes(/user_containers/current)



Return a Promise using this prototype function `getResponseFromAPI()`

Tools(/dashboards/my_tools)



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

const response = getResponseFromAPI();
console.log(response instanceof Promise);
```

Video on demand(/dashboards/videos)



```
bob@dylan:~$
Peers(/users/peers)
bob@dylan:~$ npm run dev 0-main.js
true
```



```
bob@dylan:~$
Discord(https://discord.com/app)
```



Repo:

- GitHub repository: `alx-frontend-javascript`
- `My Profile (/users/my_profile)`

- File: 0-promise.js



(/)

Check submission

> Get a sandbox



Home(/)

1. Don't make a promise...if you know you can't keep it

mandatory



Using the prototype below, return a promise. The parameter is a boolean.

My Planning(/planning/me)

```
getFullResponseFromAPI(success)
```



Projects(/projects/current)

When the argument is:



- true
 - QA Reviews I can make(/corrections/to_review)
 - resolve the promise by passing an object with 2 attributes:
 - status: 200



Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

- false
 - reject the promise with an error object with the message The fake API is not working currently



Curriculums(/dashboards/my_curriculums)

Try testing it out for yourself



```
bob@dylan:~$ cat 1-main.js
import getFullResponseFromAPI from './1-promise';
```



```
console.log(getFullResponseFromAPI(true));
console.log(getFullResponseFromAPI(false));
```



```
bob@dylan:~$
bob@dylan:~$ npm run dev 1-main.js
Promise { { status: 200, body: 'Success' } }
Promise {
  <rejected> Error: The fake API is not working currently
  ...
}
```



```
bob@dylan:~$
Tools(/dashboards/my_tools)
```



Video on demand(/dashboards/videos)

Repo:

- GitHub repository: alx-frontend-javascript
- Directory: 0x01-ES6_promise
- Peers(/users/peers)
- File: 1-promise.js



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

Check submission

> Get a sandbox



2. Catch me if you can!

My Profile(/users/my_profile)

mandatory

unc
(/)



- When the Promise resolves, return an object with the following attributes
 - status: 200

- When the Promise rejects, return an empty `Error` object


```
bob@dylan:~$ cat 2-main.js
```



```
const promise = Promise.resolve();
```

```
handleResponseFromAPI(promise);
```

?

Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

bob@dylan:~\$

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

Got a response from the API


```
bob@dylan:~$ ls -la /dashboards/my_curriculums
```


lepo: Concepts(/concepts)

- GitHub repository: `alx-frontend-javascript`
- Directory: `001-556/01boards/video_rooms`
- File: `2-then.js`

Conference rooms (dashboards/video rooms)

- File: 2-then.js



Servers(/servers)

Check submission

> Get a sandbox

>

Sandboxes(/user containers/current)

mandatory


```
1 // Import the loadPhoto and createUser from utils.js
2 import {loadPhoto, createUser} from 'utils.js';
```

Knowing that the functions in `utils.js` return promises, use the prototype below to collectively

3. solve all promises and log `body.firstName`, `lastName` to the console.

```
function handleProfileSignup()
```



```
Peers(/users/peers)
    the event of an error, log Signup system offline to the console
```

```
bob@dvlan:~$ cat 3-main.js
```



```
import handleProfileSignup from "../3-all";
Discord(https://discord.com/app)
```

```
handleProfileSignup();
```

bob@dylan:~\$

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

photo-profile-1 Guillaume Salva

```
bob@dylan:~$
```

My Profile
bob@dy lan:~\$



Repo:

- GitHub repository: alx-frontend-javascript
- Directory: 0x01-ES6_promise
- File: 3-4-1.js



Check submission

My Planning(/planning/me)
Get a sandbox



4. Simple promise

Projects(/projects/current)

mandatory

Using the following prototype



QA Reviews I can make(/corrections/to_review)

```
function signUpUser(firstName, lastName) {  
  }  
}
```



Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

That returns a resolved promise with this object:

```
{  
  firstName: value,  
  Curriculum(/dashboards/my_curriculums)  
  lastName: value,  
}
```



Concepts(/concepts)

```
bob@dylan:~$ cat 4-main.js  
import signUpUser from "../4-user-promise";  
Conference rooms(/dashboards/video_rooms)  
console.log(signUpUser("Bob", "Dylan"));
```



Servers(/servers)

```
bob@dylan:~$ npm run dev 4-main.js  
Promise { { firstName: 'Bob', lastName: 'Dylan' } }  
> bob@dylan:~/sandboxes(/user_containers/current)
```



Repo:

Tools(/dashboards/my_tools)

- GitHub repository: alx-frontend-javascript
- Directory: 0x01-ES6_promise
- File: 4-user-promise.js



Video on demand(/dashboards/videos)



Check submission

Discord(/users/me) Get a sandbox



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

5. Reject the promises

mandatory

Write and export a function named `uploadPhoto`. It should accept one argument `fileName` (string).

My Profile(/users/my_profile)

The function should return a Promise rejecting with an Error and the string \$fileName cannot be processed

```
(/)  
export default function uploadPhoto(filename) {
```

Home(/)

```
bob@dylan:~$ cat 5-main.js  
import uploadPhoto from './5-photo-reject';
```

```
console.log(uploadPhoto('guillaume.jpg'));
```

bob@dylan:~\$

```
bob@dylan:~$ npm run dev 5-main.js  
QA Reviews I can make(/corrections/to_review)  
Promise {
```

<rejected> Error: guillaume.jpg cannot be processed

```
? .. Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)
```

bob@dylan:~\$

Curriculumums(/dashboards/my_curriculumums)

- GitHub repository: alx-frontend-javascript
- Directory: 0x01-ES6_promise
- File: 5-photo-reject.js

Conference rooms(/dashboards/video_rooms)

Check submission

>_ Get a sandbox

Servers(/servers)

6. Handle multiple promises

mandatory

import signUpUser from '4-user-promise.js' and uploadPhoto from 5-photo-reject.js.

Write and export a function named handleProfileSignup . It should accept three arguments

firstName (string), lastName (string), and fileName (string). The function should call the two other functions. When the promises are all settled it should return an array with the following structure:

Video on demand(/dashboards/videos)

```
{  
  status: status_of_the_promise,  
  value: value or error returned by the Promise
```

Peers(/users/peers)

...

]


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





My Profile(/users/my_profile)

```
bob@dylan:~$ cat 6-main.js
import handleProfileSignup from './6-final-user';
(
  console.log(handleProfileSignup("Bob", "Dylan", "bob_dylan.jpg"));
);
```

```
bob@dylan:~$
bob@dylan:~$ npm run dev 6-main.js
Promise { <pending> }
bob@dylan:~$
```

 **Repo:** Projects(/projects/current)

✓  • GitHub repository: alx-frontend-javascript
• QA Reviews I can make(/corrections/to_review)
• Directory: 0x01-ES6_promise
• File: 6-final-user.js

?  Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)


Load balancer mandatory

Write and export a function named `loadBalancer`. It should accept two arguments `chinaDownload` and `usDownload` (both Promise).

The function should return the value returned by the promise that resolved the first.


```
export default function loadBalancer(chinaDownload, USDownload) {
```

 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 7-main.js
import loadBalancer from './7-load_balancer';

()
const ukSuccess = 'Downloading from UK is faster';
const frSuccess = 'Downloading from FR is faster';
Home()

const promiseUK = new Promise(function(resolve, reject) {
  setTimeout(resolve, 100, ukSuccess);
});
My Planning(/planning/me)

const promiseUKSlow = new Promise(function(resolve, reject) {
  setTimeout(resolve, 200, ukSuccess);
});
Projects(/projects/current)

const promiseFR = new Promise(function(resolve, reject) {
  setTimeout(resolve, 200, frSuccess);
});
QA Reviews (/qa-reviews/review)

const test = async () => {
  console.log(await loadBalancer(promiseUK, promiseFR));
  console.log(await loadBalancer(promiseUKSlow, promiseFR));
};
Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

Curriculums(/dashboards/my_curriculums)




test();

Concepts(/concepts)
bob@dylan:~$
bob@dylan:~$ npm run dev 7-main.js
Downloading from UK is faster
Downloading from FR is faster
bob@dylan:~$

```

 Servers(/servers)

Repo:


-  [GitHub repository: alx-frontend-javascript](#)
-  Directory: 0x01-ES6_promise
-  File: 7-load_balancer.js


 Check submission  Get a sandbox

Video on demand(/dashboards/videos)

8. Throw error / try catch

mandatory

 Write a function `divideFunction` that will accept two arguments: `numerator` (Number) and `denominator` (Number).

 When the `denominator` argument is equal to 0, the function should throw a new error with the message `cannot divide by 0`. Otherwise it should return the numerator divided by the denominator.

```

export default function divideFunction(numerator, denominator) {

}
My Profile(/users/my_profile)

```

```
bob@dylan:~$ cat 8-main.js
import divideFunction from './8-try';
()
console.log(divideFunction(10, 2));
console.log(divideFunction(10, 0));
Home()
bob@dylan:~$
bob@dylan:~$ npm run dev 8-main.js
5 My Planning(/planning/me)
.... /8-try.js:15
  throw Error('cannot divide by 0');
  ^   Projects(/projects/current)
  ....
✓ bob@dylan:~$
```

? Repo: Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

- GitHub repository: alx-frontend-javascript
- Directory: 0x01-ES6_promise
- File Curriculum(/dashboards/my_curriculums)

Check sandbox (/concepts/sandbox)

9. Throw an error Conference rooms(/dashboards/video_rooms)

mandatory

Write a function named `guardrail` that will accept one argument `mathFunction` (Function).

Servers(/servers)
This function should create and return an array named `queue`.

When the `mathFunction` function is executed, the value returned by the function should be appended to the queue. If this function throws an error, the error message should be appended to the queue. In every case, the message `Guardrail was processed` should be added to the queue.

ExampleTools(/dashboards/my_tools)

```
[
  1000, video on demand(/dashboards/videos)
  'Guardrail was processed',
]
```

Peers(/users/peers)

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



My Profile(/users/my_profile)

```
bob@dylan:~$ cat 9-main.js
import guardrail from './9-try';
import divideFunction from './8-try';

console.log(guardrail(() => { return divideFunction(10, 2)}));
console.log(guardrail(() => { return divideFunction(10, 0)}));

bob@dylan:~$
bob@dylan:~/Planning/planning/me$ cat main.js
[ 5, 'Guardrail was processed' ]
[ 'Error: cannot divide by 0', 'Guardrail was processed' ]
bob@dylan:~/Projects/projects/current$
```



QA Reviews I can make(/corrections/to_review)



- GitHub repository: alx-frontend-javascript
- Directory: 0x01-ES6_promise
- File: 9-try.js



Check submissions(/dashboards/my_curriculums)



Concepts(/concepts)



Conference rooms(/dashboards/video_rooms)

Done with the mandatory tasks? Unlock 1 advanced task now!



Servers(/servers)



Sandboxes(/user_containers/current)



Tools(/dashboards/my_tools)

Copyright © 2024 ALX, All rights reserved.



Video on demand(/dashboards/videos)



Peers(/users/peers)



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



My Profile(/users/my_profile)