Zenika **Academy**

Coder l'asynchrone en JavaScript

By Matthieu **Lux**

Coder l'asynchrone n'est pas facile

zenika

Synchrone

```
const oAuthResponse = syncRequest(oAuthOptions());
const token = JSON.parse(oAuthResponse.body).access_token;

const searchResponse = syncRequest(searchOptions(token));
const search = JSON.parse(searchResponse.body);

search.statuses.forEach(status => console.log(status.text));
```

zenika

Simple mais inutilisable et contre performant

zenika

On utilise alors les callbacks pour passer à l'asynchrone

zenika

Callbacks

```
callbackRequest(oAuthOptions(), (error, response, body) => {
  const token = JSON.parse(body).access_token;
  callbackRequest(searchOptions(token), (error, response, body) => {
    const search = JSON.parse(body);
    search.statuses.forEach(status => console.log(status.text));
  });
});
```

zenika

L'utilisation importante des callbacks crée un code complexe

zenika

Comment avoir du code asynchrone maintenable ?

zenika

Les promesses

zenika

Promesses

```
promiseRequest(oAuthOptions()).then(response => {
  const token = JSON.parse(response).access_token;
  return promiseRequest(searchOptions(token));
}).then(response => {
  const search = JSON.parse(response);
  search.statuses.forEach(status => console.log(status.text));
});
```

zenika

On peut aller encore plus loin

ES6 incoming

zenika

Generators

```
function *foo() {
  yield 1;
  yield 2;
  yield 3;
}

var it = foo();
console.log( it.next() ); // { value:1, done:false }
console.log( it.next() ); // { value:2, done:false }
console.log( it.next() ); // { value:3, done:false }
console.log( it.next() ); // { value: undefined, done: true }
```

zenika

Et si on yield une promesse ?

zenika

Generators

```
import co from 'co';

co(function *() {
  const oAuthResponse = yield promiseRequest(oAuthOptions());
  const token = JSON.parse(oAuthResponse).access_token;
  const searchResponse = yield promiseRequest(searchOptions(token));
  const search = JSON.parse(searchResponse);
  search.statuses.forEach(status => console.log(status.text));
});
```

zenika

Peut-on aller encore plus loin ?

ES7 incoming

zenika

Async

```
async function start() {
  const oAuthResponse = await promiseRequest(oAuthOptions());
  const token = JSON.parse(oAuthResponse).access_token;
  const searchResponse = await promiseRequest(searchOptions(token));
  const search = JSON.parse(searchResponse);
  search.statuses.forEach(status => console.log(status.text));
}

start();
```

zenika

Coder l'asynchrone en JavaScript

Slides
swiip.github.com/talk-async-js

Examples
github.com/Swiip/talk-async-js/tree/master/examples

Matthieu Lux (@Swiip)

zenika