

# Asynchronismes en JavaScript

---

Florent Le Gall

@flornt

Xebia



# L'asynchronisme en JS

- Les callbacks : À éviter!
  - “Pyramid of doom”
  - Gestion des erreurs difficile

```
1 var client = new Client();
2
3 var cleanup = function (error) {
4     handleError(error);
5     client.close();
6 };
7
8 client.get('/book/bookId', function (book) {
9     displayCover(book.coverUrl);
10
11     client.get(book.contentURL, function (content) {
12         displayChapter(chapter);
13         client.close();
14     }, cleanup);
15 }, cleanup);
```

# L'asynchronisme en JS

- Les promesses:
  - Permet de chaîner les appels en renvoyant une autre promesse
  - Et de gérer les erreurs simplement
  - Pur JS : Disponible partout

```
1 var client = new Client();
2
3 client.get('/book/bookId')
4   .then(function (book) {
5     displayCover(book.coverUrl);
6     return client.get(book.contentURL);
7   }).then(function (content) {
8     displayContent(content);
9   }).catch(function (error) {
10    handleError(error);
11  }).finally(function() {
12    client.close();
13  }).done();
14 }
15
```



# L'asynchronisme en JS

- Yield
  - Détourne la sémantique des générateurs pour gérer l'asynchronisme
- ECMAScript 6 Only
- Pas de traitement en parallèle !

```
1 Q.spawn(function* () {  
2   var client = new Client();  
3  
4   try {  
5     var book = yield client.get('/book/bookId');  
6     displayCover(book.coverUrl);  
7     displayContent(yield client.get(book.contentUrl));  
8   } catch (error) {  
9     handleError(error);  
10  } finally {  
11    client.close();  
12  }  
13 });  
14
```

# Gérer les promesses

---

- Q.js
  - Implémente la spécification Promises A+
  - Très populaire en 2014
  - Angular introduit une version simplifiée \$q



# Gérer les promesses

---

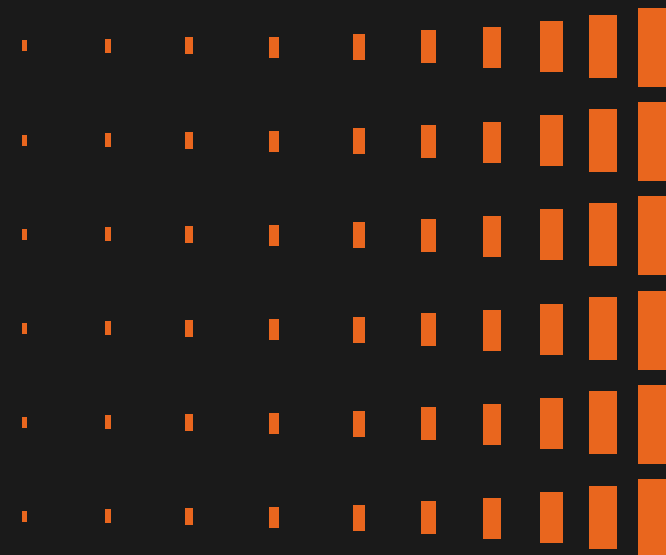
- BlueBird
  - Implémente la spécification Promises A+
  - Propose une API similaire à celles des promesses ECMA 6
  - De meilleurs performances par rapport à Q
  - Innovations liées à la programmation fonctionnelle



# Live coding !

---

<https://github.com/flegall/asynchronismes>



# Conclusion

---



# Conclusion

---

- BlueBird est la librairie de gestion de promesses la plus aboutie à ce jour
- Les apports fonctionnels se marient plutôt bien avec le JavaScript
- Le paramètre concurrency n'est pas un pool de threads

# Q & R

---