Let's React

Speaker : Mathieu Ancelin (@SERLI)
Format : Conférence
Date : 21 avril 2016
Youtube : https://www.youtube.com/watch?v=IFM8krjbKmQ

Développeur Java, Scala et JS

Introduction à la librairie React

- Librairie JS pour créer des IHM
  - Rend des vues et répond à des évènements
  - Le V de MVC
  - Pas full stack comme Angular ou Ember
- Créée en 2011 par Facebook et rendue Open Source en 2013
  - Utilisée par Netflix, Courseran Yahoo, Reddit
  - La version tirée via npm est celle qu'utilise Facebook en prod
  - L'appli Instagram est écrite en React
  - Librairie mature

2 principes de base : déclaratif et composant.

1. Déclaratif

- Exprimer à quoi doit ressembler la vue
- Rendu de toute la vue à chaque F5

Exemple de code impératif à base de *if else* pour un bouton.

Virtual DOM

- React garde en mémoire la vue précédente

- Il va calculer un diff entre la nouvelle vue et celle actuelle puis appliquer les changements
- Cela permet de rester performant

2. Approche composant

- Avec React, on fabrique des composants, pas de template
- Tout est composant
- Même les slides de la présentation

Exemple de live coding de composant React avec la syntaxe ES6.
Balisage JSX : écriture de HTML à l'intérieur de JS.
Création d'un composant React.createClass() qui renvoie la vue.

Problématique : « Comment faire entrée des données dans les composants ? »

Trois façons de faire :

1. Utilisation de **Props** :
   - Définies à la création d'une instance de composant. Idéalement immuable
   - Possibilité de mettre une valeur par défaut
   - Il existe une propriété par défaut : children. Permet d'englober des composants dans un autre.

2. Utilisation de **State** :
   - Représente l'état interne d'une instance de composant
     - L'état est mutable
   - On initialise l'état du composant avec la fonction getInitialState
   - Puis on crée une fonction like() qui change l'état du composant

3. Utilisation du **Contexte**
   - Nouvelle feature expérimentale très prometteuse

Les composants ont un cycle de vie : componentWillMount, componentDidMount, shouldUpdateComponent (super pour les perfs) …

La validation de propriétés des composants est supportée par React.

React Test Utils permet de faire des tests.
Il existe d'autres frameworks de test : Enzyme (Rbnb)
Ecosystème : React Router, Redux, Immutable.JS, Radium …

Pour commencer : Awesome React

Mathieu code en live un client Reddit (sorte de forum de discussions) avec React

React Native

Nouveau projet qui permet de rendre les composants dans une application native.
Framework créé en 2015.
Les applications sont réalisées en JS et en React Core.
React Native n'est pas de la Web View.
On utilise des composants différents que le HTML : View, Text, NavigatorIOS, TabBarIOS …

Avantages :

- Hot reload
- Ecosystème JS disponible

Devise : Learn once, write anywhere

Les vues sont différentes entre les plateformes. 80% du code entre une même application iOS et Android peut être mutualisé.

Live coding d'un client Reddit pour iOS.