React

ESGI - 3ème année IW 2020

Introduction

Objectifs

- Connaître quelques forces et faiblesses du framework
- Pouvoir démarrer un projet en React
- Comprendre le fonctionnement de React:
  - Ce qu'est un composant et son affichage sur la page
  - Passer de la donnée à un composant
  - Faire réagir un composant à un changement de donnée
  - Comprendre et se brancher sur le cycle de vie d'un composant
- Utiliser React en 2020: les hooks
- Découvrir un design pattern: les compound components

Modalités d’évaluations

- TP
- Examen écrit ?

Présentation du Framework

React en quelques points clefs

- Bibliothèque (framework) JS
- Pour créer des interfaces utilisateurs (UI)
- Privilégie l’expérience développeur
- Déclaratif plutôt que impératif
- Composants réutilisables
- Enrichissable avec des modules externes
- Intégrable dans une page HTML classique
- Maintenu par Facebook
- Open source
- Très grande communauté

Fonctionnement

- Abstraction du DOM de la page avec un DOM virtuel
- Re-affichage par composant:
  - Quand une donnée de l'état (state ou props) change
  - Quand le "parent" d'un composant s'actualise
- Événement du cycle de vie d'un composant
- JSX comme abstraction des éléments HTML

React avant 2019

- Composant déclaré grâce aux class
- Orienté objet
- Méthodes à surcharger: constructor, render, componentWillMount etc.

Problème: beaucoup de redondant, et surtout: Javascript n’est pas vraiment un langage orienté objet, mais plutôt fonctionnel

React moderne

- Déclaration des composants sous forme de fonction
- Introduction des hooks pour se “brancher” aux événements
- Code plus concis
- Compatible avec les classes et (pour le moment) ne les remplace pas

Ressources

- https://reactjs.org/docs/getting-started.html
- https://www.freecodecamp.org/learn/front-end-libraries/react/ (class)

Démarrage d’un projet

Les toolchains

- Create-react-app: uniquement React, pas d'outils supplémentaire (pas de routing, de gestion d'état global etc)
- Next.js: Server Side Rendering, avec une gestion de route intégré mais avec une complexité supplémentaire dû au SSR
- Gatsby: Générateur de page statique, avec routing intégré, mais avec des choix d'outils comme GraphQL pour gérer les assets (images, fonts…)
- Et d'autres encore...

Live coding: Créer une application et choix des outils et démo

State et Props: live coding

TP

- écrire un composant qui permet à l'utilisateur de rentrer un texte (balise input) et une balise pour afficher le contenu tapé, se mettant à jour à chaque lettre tapée ou supprimée
- écrire un composant enfant qui reçoit, en props, le contenu de ce texte et l'affiche dans une autre couleur