






GUIDE de développement Fullstack : *online Marketplace*

Ce que nous allons faire :

- **Backend**/application SaaS
 - **Node & Express** : serveur et API RESTful
 - **MongoDB Atlas** (*cluster*, BD)
 - démarrer un projet **Realm**
- **Client**/application ReactJS
 - **Redux** : gestionnaire état application
 - Class service avec **axios**
 - Email/Password authentication
 - **Stripe** : traitement paiement en ligne

Les outils :

-  [Node](#) / [npm](#) / [yarn](#)
 -  [iTerm2](#)
 -  [Visual Code Studio](#)
 -  [Compass](#)
 -  [Postman](#)
-

1. Créer le serveur avec Node et Express

- server.js :
 - `touch server.js`
- package.json :
 - `yarn init -y`
- npm/yarn dependencies :
 - `yarn add express && yarn add nodemon && yarn add body-parser && yarn add mongodb && yarn add cors && yarn add dotenv`

2. Application SaaS : MongoDB Atlas





- **MongoDB Atlas**
 - Se [connecter](#) (compte Gmail)
 - Créer un cluster (fournisseur AWS)
 - Créer une nouvelle base de donnée et une nouvelle collection
 - [Accès](#) Base de donnée (utilisateur + mot de passe)
 - Application Compass
 - [Installer](#)
 - Ajouter des données
- **Connexion Base de données**
 - connect w/ mongo shell
 - connect using **mongoDB compass**
 - connect your application
- **Tester l'API avec [Postman](#)**
 - méthodes GET et POST

3. Créer une nouvelle application *Realm*

- Connecter un cluster
- définir un *schema*
- définir les des règles et les filtres
- **Ajouter fournisseur d'authentification**
 - activer Email/Password
 - voir instructions Application React
- **Déployer son application**

4. Client - Application ReactJS

- Outils
 - Visual code studio
 - ESLint

- Bootstrap 4.5.3
- Font Awesome 5.11.2
- npm dependencies : `yarn install`
 - react-router-dom
 - redux (gestionnaire d'état), redux-thunk
 - axios (networking)
 - concurrently
 - stripe (traitement des paiements en ligne)
 - realm-web sdk
 - besoin de revoir les bases et les prérequis ?
 -  Développement Front-End - React.js 16.8
- (API REST) Service
 - method : GET, POST
 - besoin de revoir les bases et les prérequis ?
 -  Créer une API RESTful avec Node.js et Express.js
- (Redux) Gestionnaire Etat
 - actions/reducer/store
 - combineReducers
 - createSelector
 - besoin de revoir les bases et les prérequis ?
 -  Développement Front - ReactJS, Redux et l'API Contexte
 -  Reactv16 & intégration Redux : fondamentaux et avancé
- Authentification Email/Password
 - Installer web sdk ([quickstart](#))
 - `npm install realm-web`
 - `import * as Realm from "realm-web";`
 - [Authenticate user](#) : `App.logIn()`

- **(Stripe)** Traitement des paiement en ligne
 - Créer un compte développeur Stripe
 - [Quickstart](#)
 - API Test keys
 - publishable key (pk_)
 - secret key (sk_)
 - public/index.html : `<script src="https://js.stripe.com/v3/"></script>`
 - `npm install @stripe/react-stripe-js`
`@stripe/stripe-js`
 - Données pour tester intégration : [Test Cards](#)