

## Brief de votre mission

# Créer une API pour le Port de Plaisance Russell

## 1. VOTRE MISSION

Le port de plaisance de Russell aimerait se doter d'une application web de gestion des réservations de catway (petit appontement pour amarrer un bateau). La capitainerie souhaite mettre en place une API privée. Elle vous a contacté pour créer cette API ainsi qu'une partie frontend simple pour gérer les informations.

Pour cette mission, vous devez :

- Créer une application **express** qui s'appuie sur la base de données **MongoDB**,
- Mettre en place un système d'authentification,
- Alimenter la base de données avec les collections **catways** et **reservations** fournies,
- Créer l'API répondant aux besoins détaillés dans la partie "Les fonctionnalités"
- Réaliser une page d'accueil permettant la connexion et l'accès à la documentation de l'API,
- Réaliser un tableau de bord pour l'utilisateur connecté (opérations CRUD pour les collections **catways**, **reservations** et **users**),
- Créer un dépôt GitHub pour versionner votre travail et permettre le partage du code associé,
- Documenter votre API,
- Déployer votre API avec la solution de votre choix.

## 2. LES DONNÉES

### 2.1. Les utilisateurs de la capitainerie

Les utilisateurs sont caractérisés par :

- Un nom d'utilisateur (**username**),
- Une adresse de messagerie (**email**),
- Un mot de passe (**password**).

Vous utiliserez les règles courantes (et de bon sens) de contrôle des valeurs de ces données (exemples : unicité de l'adresse de messagerie, longueur minimale du mot de passe, ...).

## 2.2. Les catways

Les caractéristiques des catways sont :

- Un numéro (**catwayNumber**), unique,
- Un type (**catwayType**), pouvant prendre deux valeurs : "**long**" ou "**short**",
- Un état (**catwayState**), correspondant à une description de l'état de la passerelle.

Le fichier **catways.json** fourni vous donne quelques exemples.

Les règles de contrôle des valeurs n'étant pas précisées, vous mettrez en place celles qui vous semblent utiles et importantes.

## 2.3. Les réservations

Les caractéristiques des réservations sont :

- Le numéro du catway réservé (**catwayNumber**),
- Le nom du client ayant effectué la réservation (**clientName**),
- Le nom du bateau amarré (**boatName**),
- La date de début de la réservation (**startDate**),
- La date de fin de la réservation (**endDate**).

Le fichier **reservations.json** fourni vous donne quelques exemples.

Les règles de contrôle des valeurs n'étant pas précisées, vous mettrez en place celles qui vous semblent utiles et importantes.

## 2.4. Importer des données dans mongoDB

Pour que votre API fonctionne, vous aurez besoin de données échantillon. Un fichier **catways.json** pour importer une collection de catways et un fichier **reservations.json** pour les réservations déjà passées sont fournis avec ce devoir.

Pour importer ces deux collections dans votre base de données mongoDB, vous pouvez utiliser un client graphique MongoDB tel que [Mongo Compass](#) ou si vous êtes à l'aise avec les requêtes mongo, utiliser le *shell* directement avec les commandes suivantes :

```
mongoimport --jsonArray --db nom de la bdd --collection catways --file catways.json
```

```
mongoimport --jsonArray --db nom de la bdd --collection reservations --file reservations.json
```

# 3. LES FONCTIONNALITÉS

## 3.1. Gestion des catways

La capitainerie doit pouvoir :

- Créer un catway,
- Lister l'ensemble des catways,
- Récupérer les détails d'un catway en particulier

- Modifier la description de l'état d'un catway (le numéro et le type ne doivent pas être modifiables),
- Supprimer un catway.

Le standard d'API REST sous-entend que vos routes soient centrées autour de vos ressources et que la méthode HTTP utilisée reflète l'intention de l'action.

Vous aurez donc besoin des routes suivantes :

- GET /catways
- GET /catways/:id
- POST /catways
- PUT /catways/:id
- DELETE /catways/:id

**id** représente le numéro de catway.

### 3.2. Gestion des réservations

La capitainerie doit pouvoir :

- Créer une réservation,
- Lister l'ensemble des réservations,
- Récupérer les détails d'une réservation en particulier,
- Modifier une réservation,
- Supprimer une réservation.

Les réservations étant une sous-ressource de la ressource catway, vous devrez créer les routes suivantes :

- GET /catways/:id/reservations
- GET /catway/:id/reservations/:idReservation
- POST /catways/:id/reservations
- PUT /catways/:id/reservations
- DELETE /catway/:id/reservations/:idReservation

**id** représente le numéro de catway.

### 3.3. Gestion des utilisateurs

La capitainerie doit pouvoir :

- Créer un utilisateur,
- Lister l'ensemble des utilisateurs,
- Récupérer les détails d'un utilisateur en particulier,
- Modifier les détails d'un utilisateur,
- Supprimer un utilisateur.

Vous aurez besoin des routes suivantes :

- GET /users/
- GET /users/:email
- POST /users/
- PUT /users/:email
- DELETE /users/:email

La gestion de la connexion et de la déconnexion sera réalisée avec les routes :

- POST /login
- GET /logout

## 4. LIVRABLE ATTENDU

Le livrable attendu est :

- Un lien vers le dépôt GitHub du projet,
- Un lien permettant d'accéder à l'application hébergée ainsi que les identifiants d'un compte permettant d'accéder au tableau de bord de l'application.

L'application doit contenir :

- 1 page d'accueil répondant à la route "/" avec :
  - Une courte présentation de l'application,
  - Un formulaire pour se connecter (l'utilisateur est redirigé vers la page "tableau de bord",
  - Un lien vers la documentation de l'API.
- 1 page permettant d'accéder à toutes les opérations CRUD pour les catways,
- 1 page permettant d'accéder à toutes les opérations CRUD pour les réservations,
- 1 page permettant d'accéder à toutes les opérations CRUD pour les utilisateurs,
- 1 page "tableau de bord" avec :
  - Un menu permettant :
    - D'accéder aux pages précédentes et de se déconnecter,
    - D'accéder à la documentation de l'API,
    - De se déconnecter (et d'être redirigé vers la page d'accueil),
  - Le nom et l'adresse de messagerie de l'utilisateur connecté,
  - La date du jour,
  - Un tableau présentant les réservations en cours,
- Les pages que vous jugerez nécessaires pour les opérations CRUD (listes, détails, formulaires).