CRÉER UNE API POUR LE PORT DE PLAISANCE RUSSELL

Objectifs :

Grâce à ce devoir, vous serez capable de :

Développer dans un langage objet

Développer la partie dynamique de l'application avec des composants serveurs, dans un style défensif, et éventuellement en asynchrone Appeler des Web Services dans un composant serveur Coder de façon sécurisée les accès aux données relationnelles ou non relationnelles en consultation, création, mise à jour et suppression Utiliser un service distant (Representational State Transfer Application Program Interface (API Rest)

Consignes complémentaires :

Le livrable attendu est un lien repository GitHub du projet, avec le lien permettant d’accéder à l’application hébergée. 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.

Voici la liste des fonctionnalités nécessaires à la capitainerie :

* 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 en particulier
* Supprimer un catway
* Prendre la réservation d’un catway
* Supprimer une réservation • Lister l’ensemble des réservations
* Afficher les détails d’une réservation en particulier

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

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
* ➢ DELETE / catway/:id/reservations/:idReservation

Vous utiliserez la méthode PUT pour venir remplacer l’intégralité de l’objet à chaque fois. Parfois, vous n’aurez pas toutes les données de votre objet lorsque vous souhaiterez le mettre à jour. Dans ce cas, il faudra utiliser la méthode PATCH au lieu de PUT. Vous ajouterez donc la route :

PATCH /catways/:id à votre AP

Dans cette mission, vous devez :

➢ Créer une application express qui s’appuie sur une base mongoDB.

➢ Créer un dépôt Github pour versionner votre API et être en mesure d’en partager le code.

➢ Mettre en place un système d’authentification.

➢ Alimenter votre base de données avec les collections catways et reservations fournies avec le TP.

➢ Réaliser une page d’accueil pour la connexion et l’accès à la documentation de l’API.

➢ Réaliser une page tableau de bord pour l’utilisateur connecté.

➢ Réaliser un test pour chaque fonctionnalité.

➢ Documenter votre API.

➢ Déployer votre API avec la solution de votre choix. Ce peut être Heroku par exemple ou tout autre solution permettant de déployer votre application Node.Js sur le web. Vous pouvez également vous intéresser aux Github Actions qui offrent un workflow permettant d’automatiser le déploiement d’une application en continu (CI/CD).

# 2. LES FONCTIONNALITÉS

## 2.1. L’utilisateur de la capitainerie

Il est caractérisé par un nom (name), une adresse de messagerie (e-mail), un mot de passe (password).

## 2.2. Les catways

Ils sont caractérisés par un numéro de pont (catwayNumber), un type (type) et une description de l’état du catway (catwayState). Le type peut prendre deux valeurs « long » ou « short ».

## 2.3. Les réservations

Elles sont caractérisées par le numéro du catway réservé (catwayNumber), le nom du client (clientName), le nom du bateau amarré (boatName), une date de début de réservation (checkIn), une date de fin de réservation (checkOut).

## 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 TP. 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 la commande suivante : mongoimport --jsonArray --db nom de la bdd --collection catways --file catways.json

# 3. LIVRABLE ATTENDU

Le livrable attendu est un lien repository GitHub du projet. Le lien permettant d’accéder à l’application hébergée.

Votre application doit contenir :

## 1 page d’accueil qui répond à la route / avec :

* + Une courte présentation de l’application
  + Un formulaire pour se connecter (si la connexion se passe bien, l’utilisateur sera redirigé vers son tableau de bord)
  + Un lien vers la documentation de l’API

## 1 page tableau de bord pour un utilisateur connecté avec :

o Un formulaire pour créer un utilisateur

o Un formulaire pour modifier un utilisateur grâce à son id

o Un formulaire pour supprimer un utilisateur grâce à son id

o Un formulaire pour créer un catway

o Un formulaire pour modifier la description de l’état d’un catway grâce à son id o Un formulaire pour supprimer un catway grâce à son id

o Un formulaire pour afficher les détails d’un catway.

o Un formulaire pour enregistrer une réservation.

o Un formulaire pour supprimer une réservation grâce à son id

o Un formulaire pour afficher les détails d’une réservation grâce à son id.

o Un lien pour accéder à la liste des catways o Un lien pour accéder à la liste des réservations

## 1 page catways avec :

la liste des catways

## 1 page réservations avec :

la liste des réservations

## 1 page catway avec :

les détails d’un catway

## 1 page réservation avec :

les détails d’une réservation

## 1 page documentation avec :

la documentation de l’API

GRILLE D’ÉVALUATION

# L’APPLICATION (12 points)

CRITÈRE D’ÉVALUATION NA

L’application respecte le principe de séparation des préoccupations :

➢ L’objet req ou res n’est pas transmis aux services 0 0,25 0,5

➢ Pas de code d’état ou d’entêtes Http transmis aux services. 0 0,25 0,5

Les fichiers de code de l’application sont séparés dans des répertoires dédiés à leurs rôles : models, services, middlewares, tests, docs, etc. 0 0,5 1

La page d’accueil possède un formulaire pour se connecter 0 0,25 0,5

La connexion à mongoDB fonctionne 0 0,25 0,5

La connexion d’un utilisateur fonctionne 0 0,5 1

L’utilisateur n’a pas besoin de se connecter à chaque requête 0 0,5 1

L’application est accessible et fonctionnelle avec un navigateur web. 0 1 2

Les tests s’exécutent au lancement de l’application et les résultats des tests s’affichent dans la console. 0 0,5 1

Les 9 fonctionnalités demandées par le client font l’objet d’un test unitaire avec Mocha 0 1 2

Le code est commenté et l’application est documentée avec JSDoc. 0 0,5 1

Une documentation de L’API a été réalisée, elle est accessible via l’application et celle-ci respecte le plan

➢ Vue d’ensemble

➢ Tutoriel

➢ Exemples Glossaire 0 0,5 1

Sous total sur 12 points ……… / 12

# CATWAY (3 points)

CRITÈRE D’ÉVALUATION NA ECA A COMMENTAIRES

L’ajout d’un catway fonctionne. 0 0,25 0,5

La suppression d’un catway fonctionne 0 0,25 0,5 Les routes pour l’ajout et la suppression d’un catway sont privatisées avec un middleware d’authentification 0 0,5 1 La liste s’affiche et la route pour y accéder est privatisée avec un middleware d’authentification. 0 0,5 1 Sous total sur 3 points ……… / 3

# RÉSERVATION (3 points)

CRITÈRE D’ÉVALUATION NA ECA A COMMENTAIRES

L’ajout d’une réservation fonctionne. 0 0,25 0,5

La suppression d’une réservation fonctionne 0 0,25 0,5

Les routes pour l’ajout et la suppression d’une réservation sont privatisées avec un middleware d’authentification 0 La liste s’affiche et la route pour y accéder est privatisée avec un middleware d’authentification. 0,5 1 0 Sous total sur 3 points 0,5 1

COMPTE UTILISATEUR (2 points) ………

CRITÈRE D’ÉVALUATION NA ECA

L’ajout d’un utilisateur fonctionne 0 0,25

La suppression d’un utilisateur fonctionne 0,5 0,25 0,5

Les routes pour l’ajout et la suppression d’un utilisateur sont privatisées avec un middleware d’authentification

Sous total sur 2 points 0,5 1 ……… / 2

TOTAL SUR 20 POINTS …… / 20