

Mini-projet

Mini-Ordonnanceur**1 – Présentation***Objet*

Un ordonnanceur (*scheduler* en anglais) est un service qui coordonne la réservation des ressources dans un ensemble de serveurs de calcul (*cluster*).

Ce mini-projet consiste à réimplémenter certaines fonctionnalités d'un ordonnanceur en s'inspirant de ce qui a été fait en TP.

Informations Complémentaires

Une **ressource** est un nœud de calcul. Un cluster contient au moins une ressource. Chaque ressource se distingue des autres par un identifiant et un nom. De plus différents types de ressources peuvent avoir différentes propriétés telles que l'architecture CPU, le nombre de cœurs, la quantité de RAM etc... Dans un premier temps seul le support de l'architecture est nécessaire pour le projet. Par exemple :

- Rpi3
- AMD
- INTEL
- ARM

Une **réservation** est créée par un utilisateur et porte sur un ensemble de ressources (au moins une) du cluster, et une durée en minutes. Une réservation peut porter sur des ressources de différents types. Une réservation a une date de début et une date de fin. Durant cet intervalle de temps, l'utilisateur créateur de la réservation a un accès exclusif aux ressources incluses dans la réservation.

Organisation

Ce projet est évalué individuellement.

Deux séances de TP sont consacrées au suivi de ce projet, mais ceci n'exclut pas de travailler sur ce projet chez soi (c'est même vivement recommandé !).

Rendu

Ce projet est à rendre sur la page Moodle du cours, en suivant la procédure suivante :

- Zippez l'intégralité du répertoire contenant votre projet.
- Nommez cette archive {nom de famille}.zip, où {nom de famille} est à remplacer par votre nom de famille. Par exemple : dupond.zip pour Jean DUPOND.
- Déposez votre archive dans l'activité de type devoir nommée « Rendu mini-projet » :

<https://web.isen-ouest.fr/moodle/mod/assign/view.php?id=32322>

La date de rendu est fixée au 27 janvier 2022, 23h55.

0.1 point de malus sera appliqué par heure de retard.

2 – Fonctionnalités obligatoires

Fonctionnalité	
1	Se connecter / Se déconnecter. Erreur 401 si non connecté, affichage du pseudo sinon.
2	Création d'une réservation et gestion des conflits
3	Affichage des réservations sous forme de calendrier ou de diagramme de gantt
4	Stockage des informations (notamment des utilisateurs, les ressources et les réservations) dans une base de données
5	Recherche des utilisateurs par pseudo
6	Création d'une ressource par l'administrateur : <ul style="list-style-type: none"> - Id - Nom - Architecture
7	Affichage / recherche de ressources disponibles
8	Accès à l'application dans le cloud (Atlas + Heroku). Attention : pour que cette partie soit évaluée, il est obligatoire de renseigner l'URL de production de votre application dans le champ « homepage » de votre fichier package.json.

Pour vous faire gagner du temps sur la fonctionnalité 1, une ébauche d'application, basée sur des sessions Express, vous est fournie sur la page Moodle du cours.

3 – Contraintes techniques

Les technologies suivantes sont imposées :

- Node.js pour toute la partie serveur
- Express pour l'application Web
- MongoDB pour la base de données (client de votre choix)

4 – Barème indicatif

Qualité du code : 4 points

Critères d'évaluation (non exhaustif) :

- Fichier package.json complet
- Découpage du code en classes, fonctions, modules
- Utilisation de patrons pour les pages Web
- Prise en compte des erreurs
- Mauvaises pratiques : noms de variable inappropriés, empilement de callbacks, valeurs en dur, etc.
- Commentaires

Fonctionnalités : 16 points

Fonctionnalité	Barème indicatif
1 – (Dé)Connexion	2 points
2 – Création de réservation	4 points
3 – Affichage des réservations	1 point
4 – Persistance des données en base	3 points
5 – Recherche d'utilisateurs	2 points
6 – Création d'une ressource	1 point
7 – Affichage / recherche de ressource(s)	2 point
8 – Déploiement dans le cloud	2 points

5 – Fonctionnalités optionnelles

- Mise à jour des réservations en temps réel, sans avoir à rafraîchir la page.
- Notifications de début et de fin de réservations
- Gestion des ressources et utilisateurs :
 - Affichage des utilisateurs les plus consommateurs de ressources
 - Affichage des ressources les plus utilisées
- Profils utilisateurs :
 - Permettre à un utilisateur de définir un profil avec nom, prénom, pseudo, avatar
 - Afficher l'avatar des utilisateurs dans les tweets et les réponses.
 - Afficher les réservations d'un utilisateur
- Authentification à base de JSON Web Tokens (JWT) (principe [ici](#)).
- Définition de propriétés multiples pour les ressources
- Modification de l'état des ressources réservées pendant la réservation. Ceci n'est possible que par l'auteur de la réservation