

# Gestion d'équipes de projet

---

Ce TP consiste à créer un système pour gérer les équipes de devs d'un éditeur de logiciels.

Le management reçoit une demande de projet qui liste tous les détails nécessaires pour sa réalisation.

On doit d'abord constituer une équipe d'au moins 4 devs et maximum 8 ayant les compétences requises. Puis lancer le projet dès que possible.

Vous êtes en charge de réaliser la partie chargée de gérer les projets et les équipes.

## Le projet :

La demande d'un nouveau projet contient : - le nom du projet - une priorité : normale, best effort, critique - une description - une date de début qui est forcément dans le futur - une date de fin (obligatoirement après la date de début) - une stack contenant la liste des technos nécessaires et le nombre de devs requis par techno

A partir de là, le management va interroger notre système pour connaître la liste des devs disponibles à la date de démarrage de projet et qui correspondent aux critères demandés. Ensuite, il va pouvoir affecter une équipe sur le projet.

Tant qu'un projet n'a pas commencé (il est en attente), on peut le repousser ou changer la future équipe. Si plusieurs projets sont en attente et demandent les mêmes expertises, c'est le projet avec la priorité la plus haute qui commencera en premier. En cas de priorité identique, c'est celui qui finira le plus tôt en premier.

Un projet ne peut durer moins d'un mois Un projet qui commence se termine à la date de fin (pas d'annulation ou de report possible) Un projet en attente peut être annulé. Une demande de projet peut être rejetée si aucun.e dev n'a les compétences requises ou si aucune équipe n'est disponible avant 1 an

## L'équipe :

- Elle est constituée pour un projet donné et dissoute à la fin du projet
- Elle doit être constituée d'au moins 3 devs et 8 au maximum
- Elle ne peut accueillir un junior qu'à condition qu'il y ait un.e expert.e
- On peut pas avoir plus de 3 juniors dans la même équipe
- Les devs intermédiaires n'ont pas besoin d'expert.e mais ne pourront pas prendre un projet de plus de 6 mois sans expert.e
- Un.e expert ne peut pas être dans une équipe de moins de 5 devs, sauf pour un projet critique
- Un.e dev peut demander à changer d'équipe à condition de respecter les règles précédentes. Et que son dernier changement d'équipe date de plus de 6 mois

## Les devs

- Un.e dev a un nom et une adresse email (qui lui sert aussi d'identifiant unique)
- Un.e dev a une liste de compétences techniques
- Son niveau technique sur une compétence est définie par son nombre d'années d'expérience : 0-3 ans, junior.e. 3-5 ans, expérimenté.e. Plus de 5 ans, expert.e

# Réalisation

---

L'application se présentera sous la forme d'une webapp exposant une API REST que nous pourrons interroger avec un outil comme curl ou postman

## Les fonctionnalités :

- Accueillir un.e dev au sein de la société (embauche)
- Exclure un.e dev de la société (démission)- Lister tous les devs correspondant à une compétence et un niveau donné- Lister tous les projets terminés, en cours ou prévus- Connaître le prochain projet qui va démarrer
- Annuler un projet prévu
- Traiter une demande de nouveau projet- Connaître le CV d'un.e dev (la liste de ses projets et ses compétences)- Transférer un.e dev dans une autre équipe

## La stack :

- Java 18 (au moins)
- le framework Javalin pour la partie REST
- pour le stockage de données, un fichier (texte, json, xml, etc.) sera amplement suffisant

## Règles générales :

---

- Pour la partie Javalin, le cours abordera les concepts de programmation fonctionnelle nécessaire courant janvier. Si vous souhaitez commencer avant, faites-moi signe
- Concentrez-vous sur une fonctionnalité à la fois et terminez-la avant de passer à la suivante. N'essayez pas de faire des bouts de chaque
- On s'accorde à ce que le système puisse mettre quelques secondes pour répondre. Comprendre : concentrez-vous sur le code et les principes objets plutôt que sur de potentielles optimisations
- La lisibilité et la clarté de votre code comptent dans la notation. Pensez-y
- Ce serait bien dommage que je trouve un code uniquement en lisant votre code. Pensez aux tests !
- Ne perdez pas votre temps à faire plus que l'énoncé : pas besoin de base de données ou d'UI...
- Si vous avez la moindre interrogation, posez vos questions. N'essayez surtout pas de deviner. C'est le meilleur moyen de vous tromper