

TD

Full Stack

Partie 1 : Implémentation du
rateLimit

Partie 2 : Implémentation des
ETags

Partie 3 (bonus) : Implémentation
de HATEOAS

Partie 1 : Implémentation du Rate Limit

Vous attendez une forte affluence sur la partie public de l'application (réservation de rendez-vous de vaccination). Vous souhaitez donc mettre en place une file d'attente si vous recevez trop de demandes simultanées.

- Sur le back de l'application :
 - Mettre en place un Rate Limit utilisant le Token Bucket
 - Si le service reçoit plus de 10 demandes/minutes alors il retourne une erreur 429
- Sur le front de l'application :
 - Si vous recevez une réponse 429 alors rediriger l'utilisateur vers une file d'attente
 - La file doit faire patienter jusqu'au prochain remplissage du bucket

Partie 2 : Implémentation des ETags

Pour éviter les modifications concurrentes sur la partie administration de l'application, mettez en place un versionnage des ressources.

- Sur les API d'administration utiliser un filtre sur les Entity Tags
 - Les réponses doivent contenir l'ETAG de la ressource
 - Mettre en place un cache des ETags déjà transmis
 - Les requêtes GET doivent porter le header IF-NONE-MATCH
 - Les requêtes PUT, POST, PATCH doivent porter le header IF-MATCH
- Si une requête renvoie une erreur 412
 - Le front doit en informer l'utilisateur via une modale
 - Cette modale doit proposer à l'utilisateur de recharger la ressource
- Si une requête renvoie un code 304
 - On ne met pas à jour la ressource dans le front

Partie 3 (bonus) : Implémentation de HATEOAS

- Vous souhaitez réduire la complexité de votre front public.
- Vous allez implémenter la contrainte HATEOAS sur la réservation de rendez-vous.

Cette partie est optionnelle