

# Framework

Apprentissage de Spring à travers la création d'une API REST avec Spring-boot.

CSID 2020 – 2021

Par Quentin LEULY ( [q.leuly@iut.univ-paris8.fr](mailto:q.leuly@iut.univ-paris8.fr) )

Le but de ce module est l'apprentissage de Spring à travers la création d'une API REST avec Spring-boot. A travers ce TP, vous apprendrez à créer une API REST en utilisant l'ensemble des méthodes HTTP existantes (GET, POST, PUT, PATCH, DELETE) et l'injection de dépendance avec Spring.

Dans cette série de TPs, vous programmerez une API qui ira effectuer des actions sur des repositories Github à travers l'API REST de Github (<https://docs.github.com/en/free-pro-team@latest/rest>). Cette API devra permettre à un client HTTP de récupérer certaines informations sur des repositories, des statistiques le nombre d'appel effectué sur chaque repository et enfin de créer des *issues* et *pull-requests* sur ces repositories.

Fonctionnalités attendues :

- CRUD<sup>1</sup> sur les repositories
- Lecture de nombre de issues et de pull-requests sur un repository
- Statistiques sur les repositories
- Création de issues ( et pull-requests ) sur un repository donné
- Gestion basique de droits (au début sans utilisateur) : autorisation de faire (ou non) des opérations de lecture ou écriture sur un repository donné
- CRUD sur les utilisateurs
- Statistiques sur les utilisateurs

## I. CRUD sur les repositories

Create – Read – Update - Delete

On souhaite enregistrer et lire des informations sur les repositories auxquels votre API va avoir accès : nom du repository, nom du propriétaire, nombre de issues, nombre de pull-requests. Le nombre de issues et de pull-requests sera, au début, lu depuis la base de données, et à terme, récupéré directement depuis Github.

---

1 Create – Read – Update - Delete

## **II. Lecture de nombre de issues et de pull-requests sur un repository**

Ici, nous souhaitons utiliser l'API de Github afin de connaître le nombre de issues et de pull-requests sur nos repositories. Les deux informations seront retournées à partir de endpoints différents. Selon la méthode HTTP utilisée, on ira lire l'information depuis Github et on la retournera ou on enregistrera la valeur répondue par Github en base avant de la retourner.

## **III. Statistiques sur les repositories**

Pour chaque repository, il faudra retourner, sous forme de liste, le nombre de issues et de pull-requests au fil du temps. Ces données pourraient par exemple servir à tracer un graphique.

De plus, on voudra aussi savoir combien de fois on a requêté l'API Github pour les issues et les repositories.

## **IV. Création de issues ( et pull-requests ) sur un repository donné**

Un peu plus de complexité dans cette partie, on ne va plus lire des informations depuis Github, mais aussi en écrire. A partir de cette partie, vous aurez besoin de générer un token (Personal Acces Token) à partir de votre compte Github (<https://github.com/settings/tokens>). Veillez bien à garder ce token secret et à ne pas le commit sur vos repositories. Si vous le perdez, supprimez-le de votre compte et régénérer un.

## **V. Gestion basique de droits (sans utilisateur)**

Dans cette partie, nous souhaitons implémenter une gestion basique des droits sur les repositories. Par exemple, on pourra soit autoriser la lecture ou l'écriture sur les repositories en fonction du type d'objet sur lequel la requête porte (issue ou pull\_request).

Par exemple :

- Sur le repository A :
  - autorisation en lecture et écriture sur les issues
  - autorisation en lecture seulement sur les pull-requests

- Sur le repository B :
  - autorisation en lecture sur les issues
  - autorisation en lecture sur les pull-requests

Une autorisation d'écriture implique forcément l'autorisation de lecture et n'avoir aucune autorisation est aussi possible.

## **VI. Gestion des utilisateurs**

On va maintenant créer et gérer quelques utilisateurs. Les utilisateurs pourront se connecter et être authentifié (à l'aide d'un cookie ou un token JWT). La gestion des autorisations des actions sur les repositories sera maintenant reliée aux utilisateurs (un utilisateur aura les droits en lecture et/ou écriture sur un repository ou non). Sans être authentifié, on ne pourra plus effectuer les actions d'écriture sur les repository. On garde néanmoins la gestion des autorisations en lecture/écriture au niveau d'un repository : si la configuration d'un repository prohibe la lecture ou l'écriture, un utilisateur, même s'il en possède le droit ne pourra pas effectuer l'action demandée.

## **VII. Statistiques sur les utilisateurs**

Dans la partie Statistiques sur les repositories, on a enregistré les requêtes faites sur les repositories et leur type ; ici, on veut en plus enregistrer quel utilisateur a fait la requête et récupérer sous forme de liste ces informations. Les informations ne seront accessibles qu'à l'utilisateur connecté (ou à un administrateur).