# Mise en place de springdoc-openapi

# Démonstration 1 du module 7

|  |
| --- |
| Les objectifs de cette démonstration sont   * Ajout du starter pour springdoc-openapi * Manipulation de l’API du projet demo-nosql au travers de Swagger-UI |

Contexte

* Nous reprenons notre deuxième application, qui permettra de gérer les avis des stagiaires.
* Elle permettra plus tard, de réaliser des statistiques sur les cours.
* Elle est indépendante de l’application demo-eni-ecole.
* Les informations en base sont en anglais pour différencier les BO et les Document
* Il s’agit de l’application demo-no-sql
  + Si votre application n’est pas complète, vous pouvez reprendre le code Java dans les ressources
  + Attention, le paramétrage du projet Spring Boot n’est pas inclus
  + Il vous faut reprendre les démonstrations du module 3 et 5 pour valider les starters de Spring Boot
* Dans ce module, nous allons créer la documentation de l’API

Déroulement

# Dépendances Spring Boot :

* Dans build.gradle ; ajouter le starter de de springdoc-openapi

//Utilisation de Swagger - Open API UI

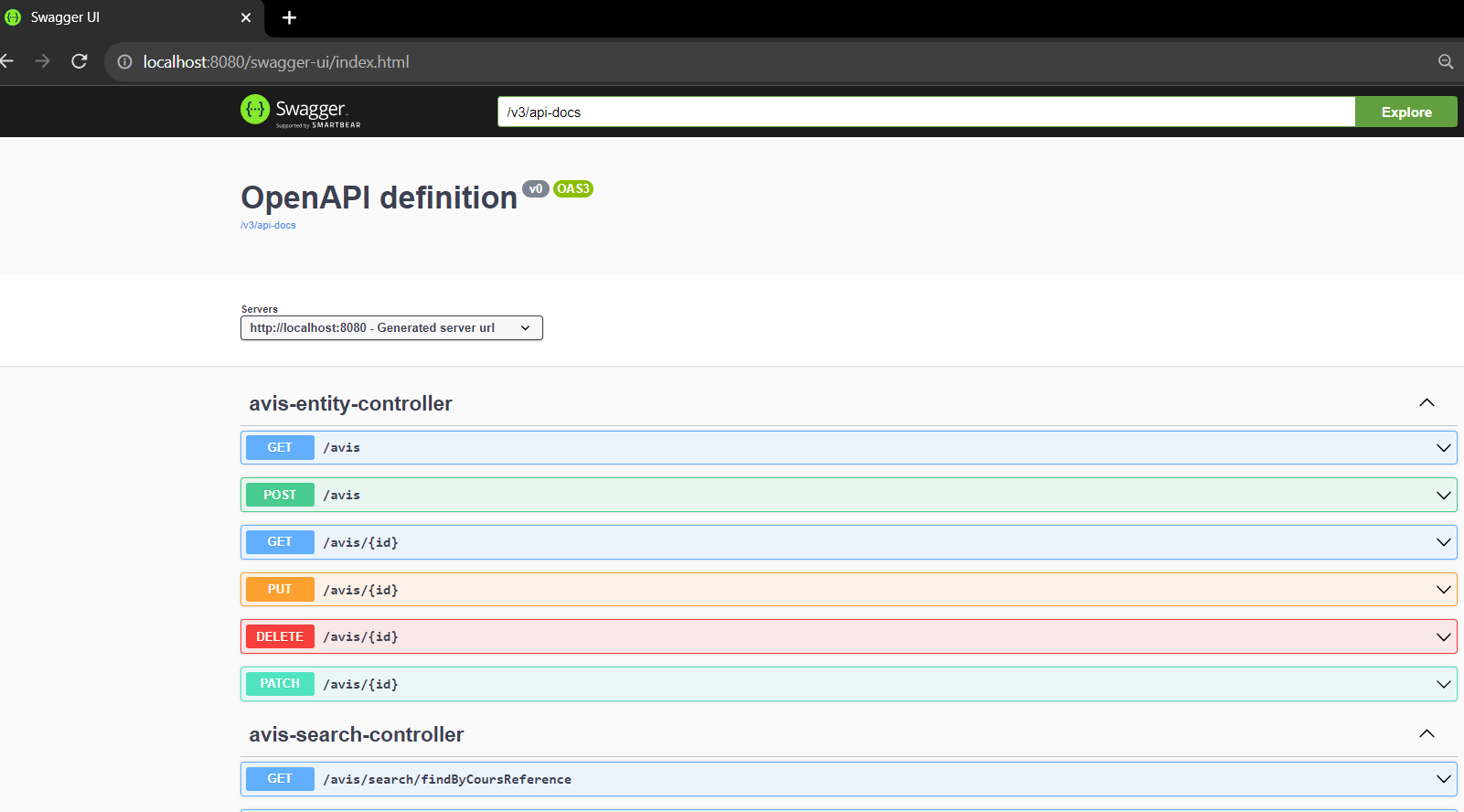
implementation 'org.springdoc:springdoc-openapi-starter-webmvc-ui:2.1.0'

* + Pour le numéro de version, vérifier sous Maven Repository, la version courante
    - La version présentée est la minimale à avoir avec le Spring Boot actuel
* Faire un Gradle Refresh Project

springdoc-openapi va générer automatiquement la documentation de vos APIs

# Exécution

* Lancer l’application
* Pour accéder aux informations ; il faut saisir l’URL :
  + <http://localhost:8080/swagger-ui/index.html>



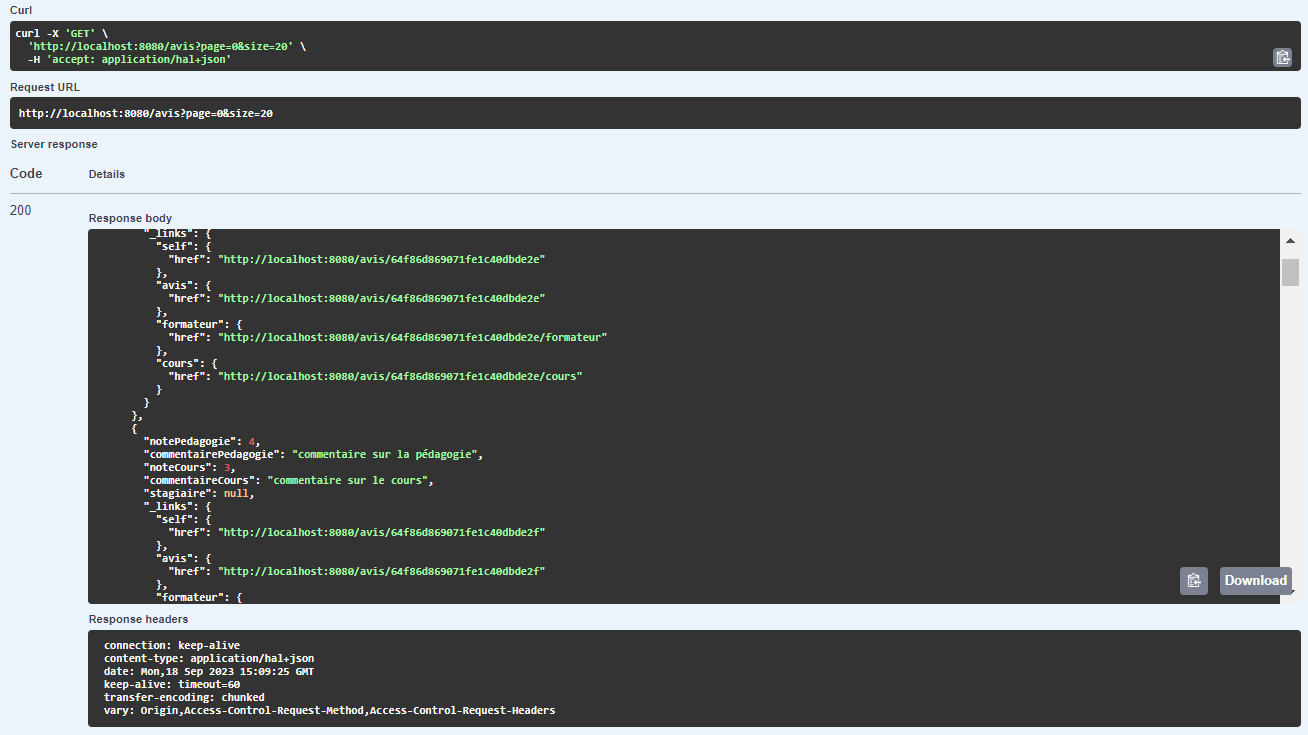
Il n’est plus nécessaire d’utiliser Postman pour tester votre application

* Vous pouvez directement tester les APIs depuis la vue courante
* Prenons l’exemple de Get /avis

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

* + Cliquer sur le bouton « Try it out » 🡪 pour pouvoir exécuter la demande
  + Remarquez que vous pouvez préciser le numéro de la page
  + Le nombre d’élément par page
  + Et un tri par défaut
* Un fois cliquer sur « Execute »
  + Vous aurez le statut http de la réponse
  + Le contenu de la réponse
  + Vous retrouverez les mêmes informations que dans Postman



Remarquez que vous avez la totalité des possibilités proposées par Spring Data REST de lister dans votre vue Swagger

Vous pouvez manipuler les GET, POST, PUT, PATCH et DELETE au travers de Swagger UI