Documentation API

OpenAPI Specification

UP ASI
Bureaux E204 | E304

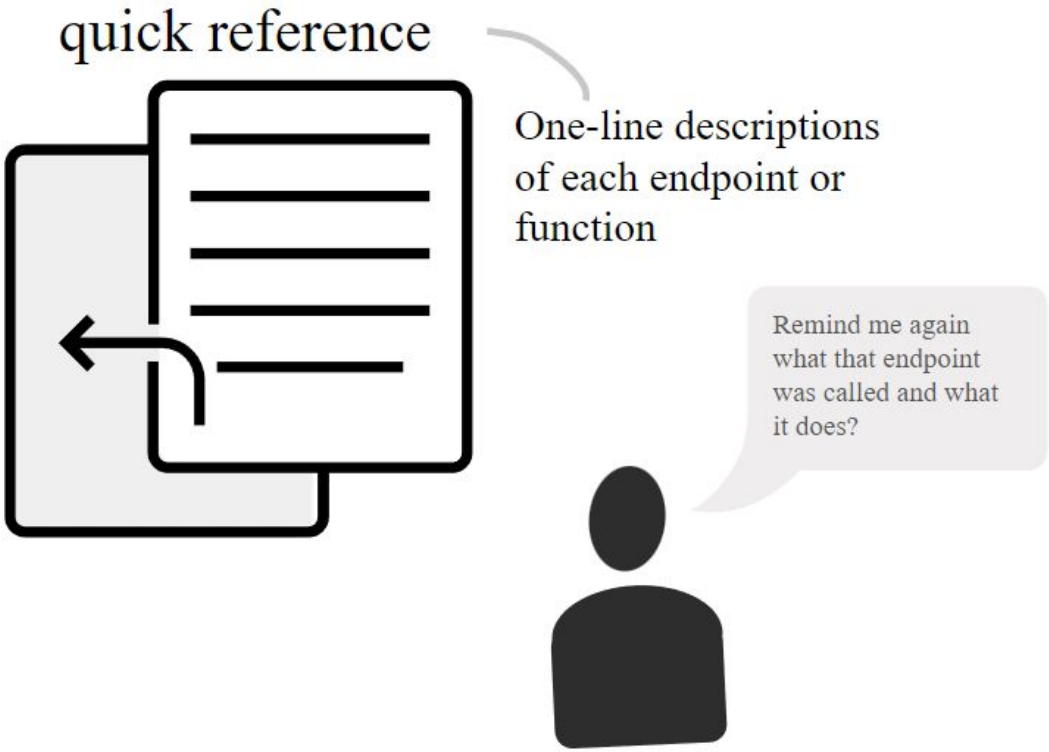
PLAN DU COURS

- Introduction
- OpenAPI Spécification
- Implémentations
- Intégration SpringDoc
- Configuration SpringDoc
- TP

Introduction

Que signifie documentation API ?

C'est un manuel de référence précis qui contient les informations nécessaires pour travailler avec l'API, notamment des détails sur les fonctions, les classes, les types de retour et les arguments.

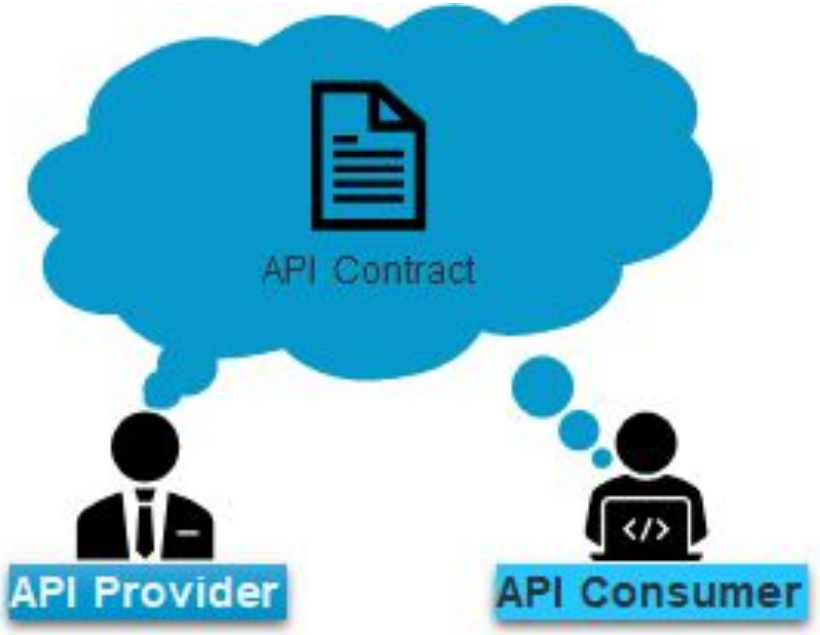


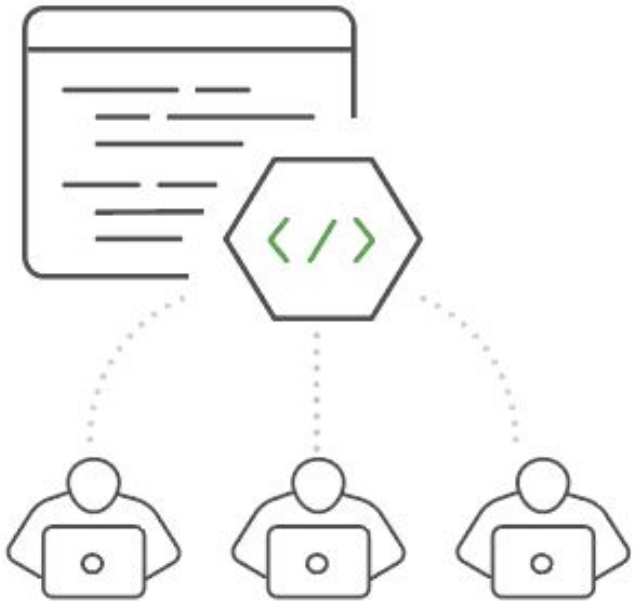
Introduction

Pourquoi ?

Les API ont pour vocation de servir de nombreux développeurs.
Par conséquent, le développement d'une API nécessite une documentation accessible et facilement utilisable.

Cette dernière doit toujours être à jour au fur et à mesure que le code et les fonctionnalités de l'API évoluent.





OpenAPI Specification

La documentation d'un service RESTful consiste essentiellement à décrire les détails des requêtes HTTP qu'elle consomme et des réponses HTTP qu'elle produit.

Préparer une documentation de qualité est une tâche difficile.

Il est donc important d'utiliser des outils appropriés à cette tâche.

Les formats de description d'API tels que **la spécification OpenAPI**(Swagger v3) ont permis de simplifier la création et la maintenance de la documentation.

Elle garantit également que votre documentation soit à jour au fur et à mesure de l'évolution de votre API.

Implémentations

Pour l'intégration d'OAS (openApi Specification) dans notre application, nous devons faire appel à l'implémentation qui nous convient.

Language	Implémentation
Java	springdoc-openapi
.Net	Microsoft.OpenApi.net
Node.js	oas3-remote-refs

Intégration SpringDoc

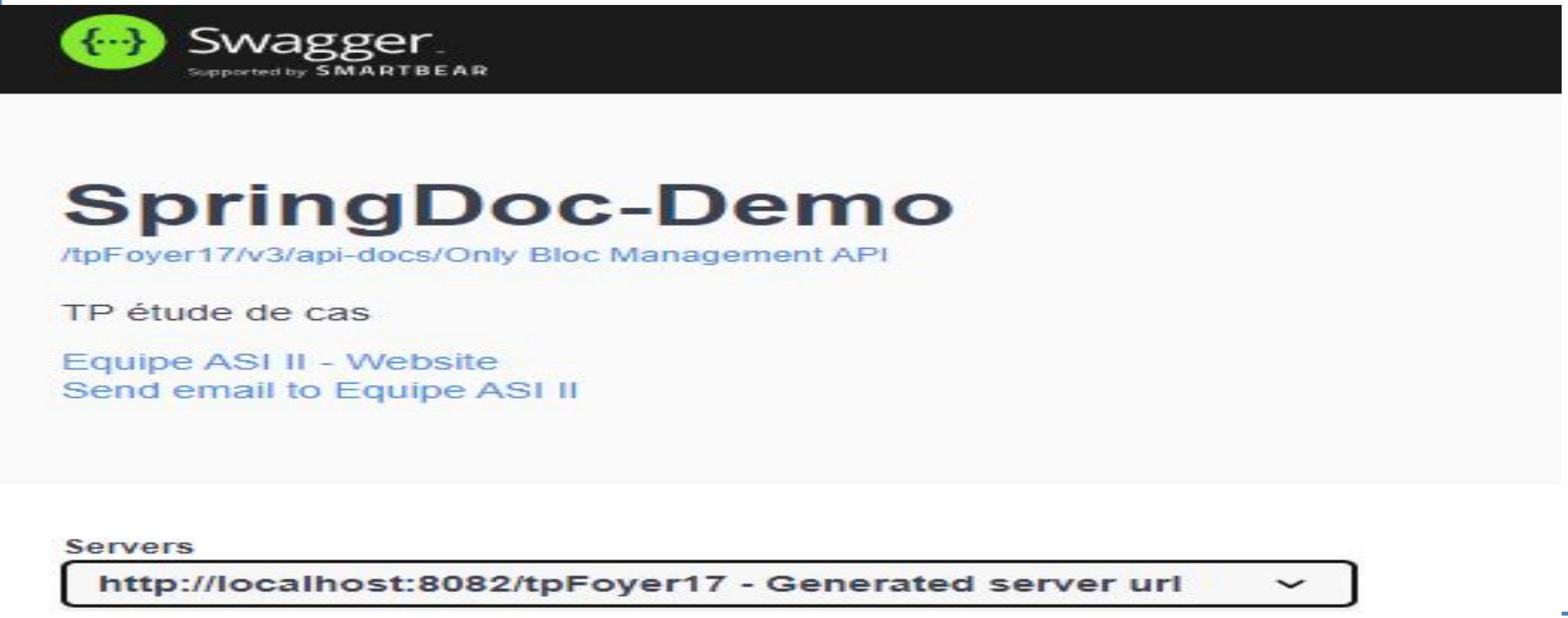
Pour commencer, nous allons simplement ajouter la dépendance springdoc-openapi-ui dans le fichier **pom.xml.**
Et puis faire maven update de votre projet.

```
<dependency>
<groupId>org.springdoc</groupId>
<artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
<version>2.0.0</version>
</dependency>
```

c'est tout, aucune configuration supplémentaire n'est nécessaire.
La documentation sera disponible au format HTML en utilisant l'outil swagger-ui.

Intégration SpringDoc

La page Swagger UI sera alors disponible à l'adresse:
http://server:port/context-path/swagger-ui.html
serveur : Le nom ou l'IP du serveur
port : Le port du serveur
context-path : Le chemin du contexte de l'application



Configuration SpringDoc

- Pour personnaliser l'interface utilisateur swagger-ui, Créez une classe **OpenAPIConfig** dans un nouveau package appelé **configuration**.

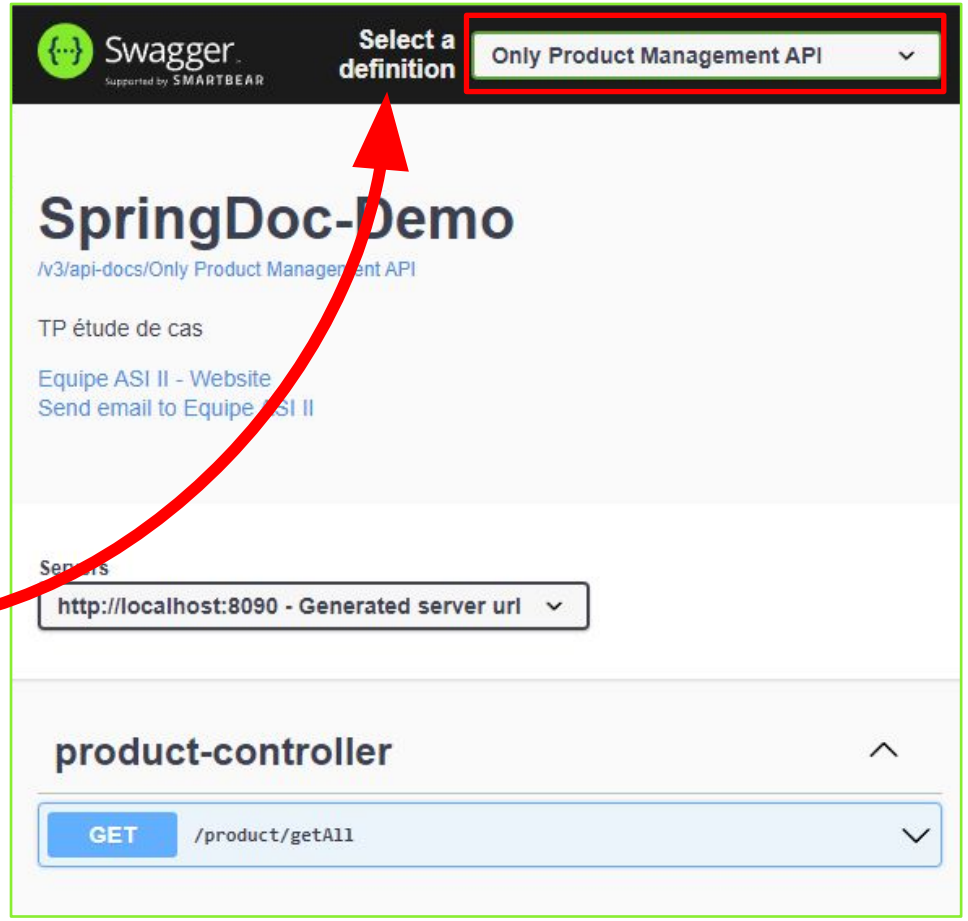
```
@Configuration
public class SpringDocConfig {

@Bean
public OpenAPI springShopOpenAPI() {
      return new OpenAPI()
                  .info(infoAPI());
}

public Info infoAPI() {
      return new Info().title("SpringDoc-Demo")
                  .description("TP étude de cas")
                  .contact(contactAPI());
}

public Contact contactAPI() {
      Contact contact = new Contact().name("Equipe ASI
II")
                  .email("**********@esprit.tn")
                .url("https://www.linkedin.com/in/**********/");

       return contact;
}
```

Configuration SpringDoc

On peut aussi regrouper nos API en fonction des paquets, des chemins, etc. en utilisant le **GroupedOpenApi.**

```
@Bean
public GroupedOpenApi productPublicApi() {
      return GroupedOpenApi.builder()
                              .group("Only Product Management
API")
                              .pathsToMatch("/product/**")
                              .pathsToExclude("**")
                              .build();
}
```

Vous pouvez ensuite les sélectionner dans swagger-ui en sélectionnant la définition voulue.



Configuration SpringDoc

POST `/api/etudiants/addEtudiant`

Parameters

No parameters

Request body required

```
{
  "nomEtudiant": "said",
  "prenomEtudiant": "said",
  "cinEtudiant": "789654",
  "dateNaissance": "1111-07-11T13:46:08.132Z"
}
```

Request URL

```
http://localhost:8082/tpFoyer17/api/etudiants/addEtudiant
```

Server response

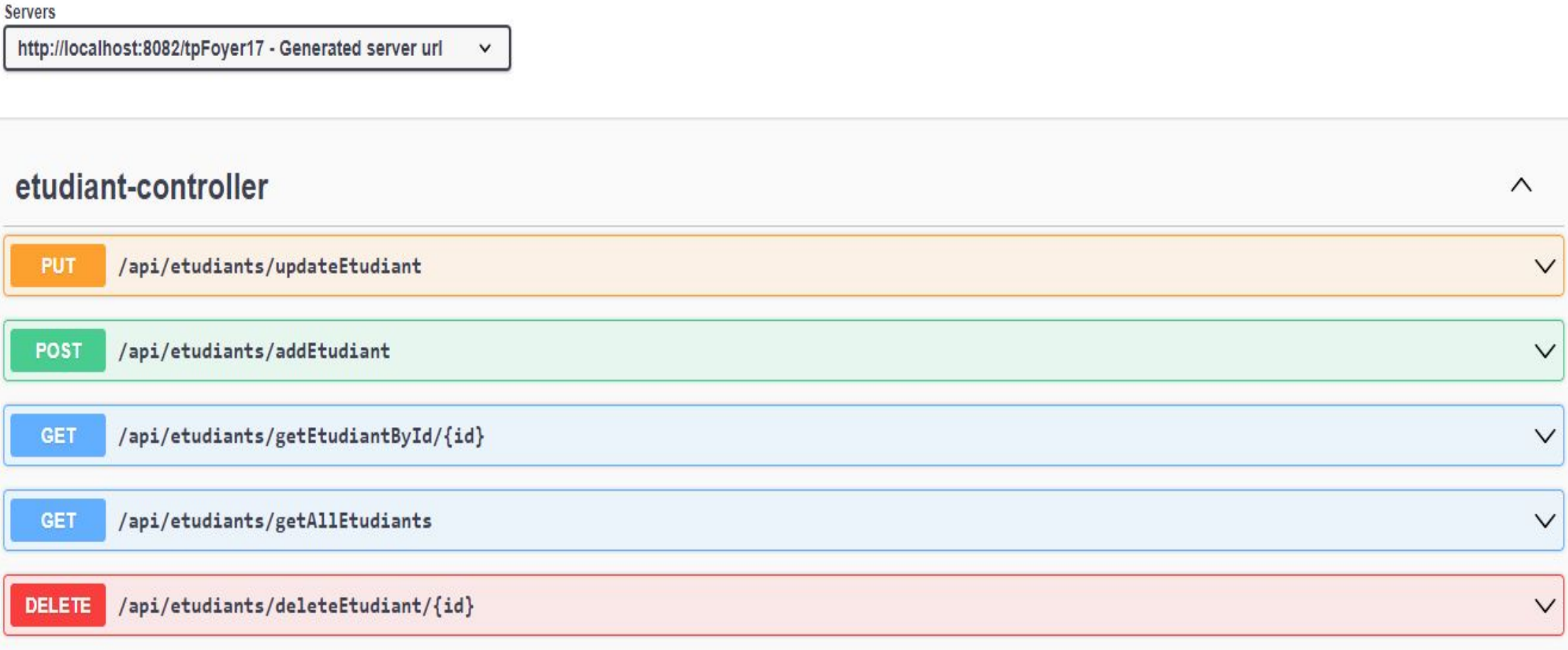
Code	Details
200	Response body

```
{
  "idEtudiant": 3,
  "nomEtudiant": "said",
  "prenomEtudiant": "said",
  "cinEtudiant": "789654",
  "dateNaissance": "1111-07-11T13:46:08.132+00:00",
  "reservations": null
}
```

Configuration SpringDoc

Nous pouvons aussi personnaliser la documentation grâce aux annotations:

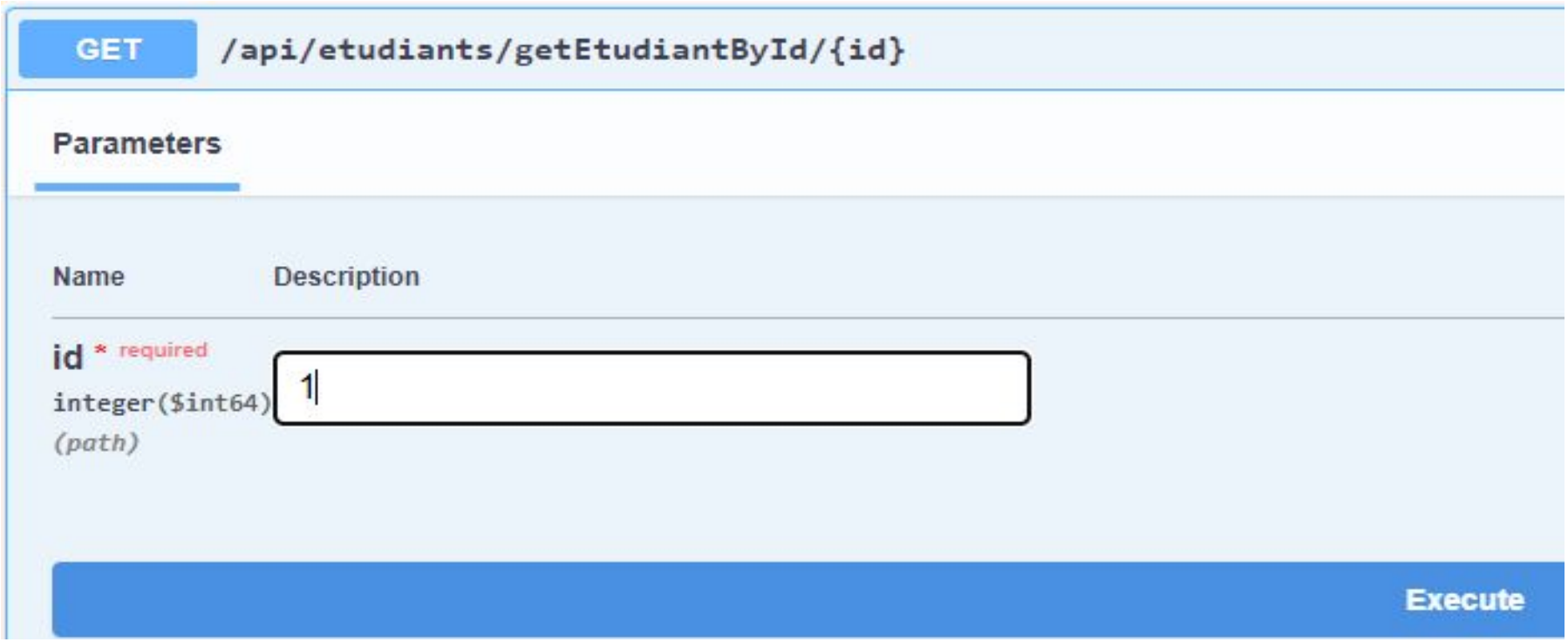
- **@Tag** (@Api in swagger 2): permet d'ajouter une description pour chaque classe RestController.
- **@Operation**(@ApiOperation in swagger 2): permet d'ajouter une description pour chaque classe RestController.



Configuration SpringDoc

Nous pouvons aussi personnaliser la documentation grâce aux annotations:

- **@Tag** (@Api in swagger 2): permet d'ajouter une description pour chaque classe RestController.
- **@Operation**(@ApiOperation in swagger 2): permet d'ajouter une description pour chaque classe RestController.



TP

Intégrer SpringDoc au niveau du projet Gestion Foyer et ajouter les annotations nécessaires pour bien documenter les services.

Si vous avez des questions, n'hésitez pas à nous contacter :

Département Informatique
UP ASI
Bureau E204