## Tạo dự án

<groupId>com.kiss</groupId>

<artifactId>RESTful\_API\_Document\_OpenAPI\_3</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>RESTful API Document Open API 3</name>

<description>RESTful API Document Open API 3</description>

<properties>

<java.version>17</java.version>

</properties>

## pom.xml

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<version>3.0.4</version>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-configuration-processor</artifactId>

<version>3.0.4</version>

<optional>true</optional>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>1.18.26</version>

<optional>true</optional>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<version>3.0.4</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springdoc</groupId>

<artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>

<version>2.0.4</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

<version>3.0.4</version>

</dependency>

<dependency>

<groupId>com.microsoft.sqlserver</groupId>

<artifactId>mssql-jdbc</artifactId>

<version>12.2.0.jre11</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-validation</artifactId>

<version>3.0.4</version>

</dependency>

## Entity

@Entity

@Data //Tạo getter, setter

@NoArgsConstructor //Hàm tạo không tham số

@AllArgsConstructor //Hàm tạo với đầy đủ tham sô

public class Users {

@Id

@Schema(description = "User UUID in the database") // Mô tả ý nghĩa field trong document

@JsonProperty("UserId") //Thay đôi name property cho json trả về

private int id;

private String fristName;

private String lastName;

private String email;

}

## Repository

@Repository

public interface UserRepository extends JpaRepository <Users,Integer>{

}

## Controller

@RestController //Trả về dữ liệu dạng json

@RequestMapping("/api/v1/")

//Mô tả các trạng thái phản hồi của api

@ApiResponses(value = {

@ApiResponse(responseCode = "200", description = "Thành công"),

@ApiResponse(responseCode = "401", description = "Chưa xác thực"),

@ApiResponse(responseCode = "403", description = "Truy cập bị cấm"),

@ApiResponse(responseCode="404", description = "Không tìm thấy")

})

public class Controller {

@Autowired

private UserRepository userRepository;

@Operation(description = "Xem danh sách User") //Mô tả chức năng của api

@GetMapping("/users")

public ResponseEntity<List<Users>> getUsers() {

List<Users> listUser = userRepository.findAll();

if(listUser.isEmpty()) return ResponseEntity.*noContent*().build();

return new ResponseEntity<>(listUser, HttpStatus.*OK*);

}

@Operation(description = "lấy một User bằng id")

@GetMapping("/users/{id}")

public ResponseEntity<Users> getUser(@PathVariable("id") int id) {

Users user = userRepository.findById(id).orElse(null);

System.*out*.println(user);

if(user == null) return ResponseEntity.*notFound*().build();

return ResponseEntity.*ok*(user);

}

@Operation(description = "Tạo một User")

@PostMapping("/Users")

public ResponseEntity<Users> createUser(@Valid @RequestBody Users user) {

return ResponseEntity.*ok*(userRepository.save(user));

}

@Operation(description = "Cập nhật User theo id")

@PutMapping("/Users/{id}")

public ResponseEntity<Users> updateUser(@PathVariable("id") int id, @Valid @RequestBody Users user) {

user.setId(id);

return ResponseEntity.*ok*(userRepository.save(user));

}

@Operation(description = "Xóa User theo id")

@DeleteMapping("/Users/{id}")

public void deleteUser(@PathVariable("id") int id) {

userRepository.deleteById(id);

}

}

## Custom OpenAPI

@Configuration

public class OpenApiConfig {

@Bean

public OpenAPI customOpenAPI() {

return new OpenAPI()

.servers(List.*of*(

new Server().url("http://localhost:8080").description("Localhost"),

new Server().url("https://user.kiss.me").description("kiss.me")

))

.info(new Info()

.title("Kiss Application API")

.description("Sample OpenAPI 3.0")

.contact(new Contact()

.email("honthfx14144@funix.edu.vn")

.name("kiss")

.url("https://kiss.me/"))

.license(new License()

.name("Apache 2.0")

.url("http://www.apache.org/licenses/LICENSE-2.0.html"))

.version("1.0.0"));

}

}

## application.properties

spring.datasource.url=jdbc:sqlserver://172.17.0.1:1433;databaseName=QTCN;encrypt=true;trustServerCertificate=true;

spring.datasource.username=SA

spring.datasource.password=Kisbest01

spring.datasource.driver-class-name=com.microsoft.sqlserver.jdbc.SQLServerDriver

##JPA/Hibernate

spring.jpa.show-sql=true

spring.jpa.hibernate.ddl-auto=none

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.SQLServerDialect

spring.jpa.properties.hibernate.current\_session\_context\_class=org.springframework.orm.hibernate5.SpringSessionContext

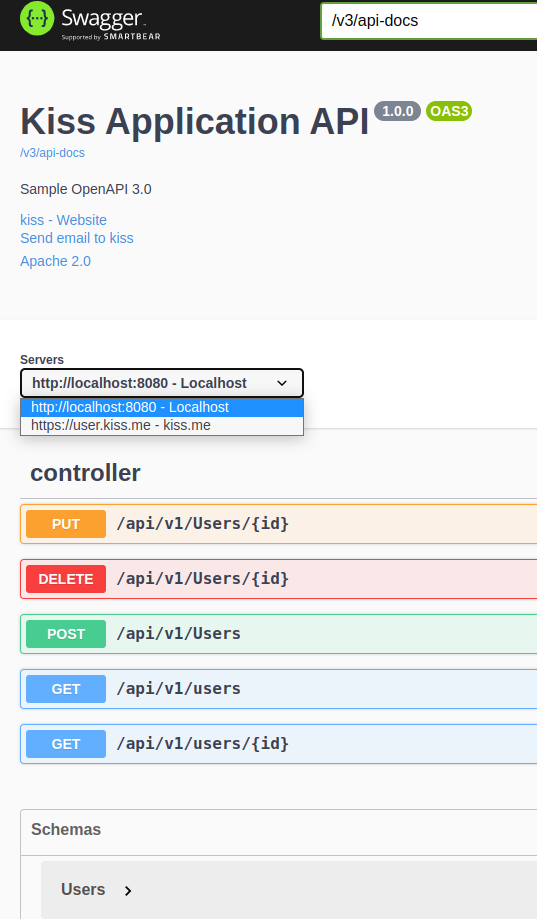
##Ghi ?è các giá tr? truy v?n m?c ??nh

spring.jpa.hibernate.naming.implicit-strategy=org.hibernate.boot.model.naming.ImplicitNamingStrategyLegacyJpaImpl

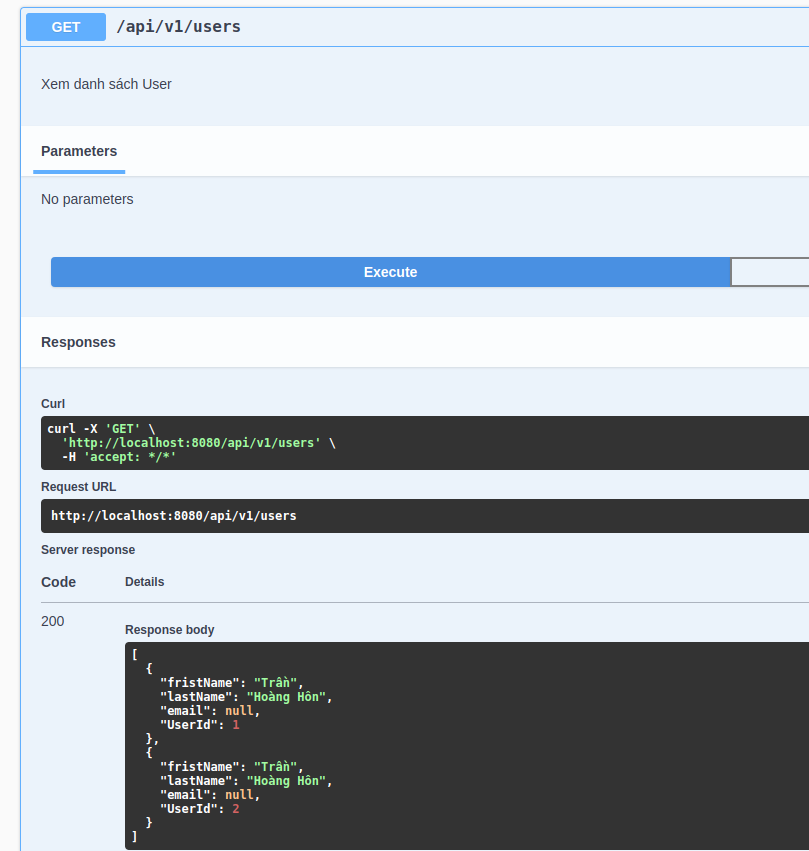
spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl

## Test

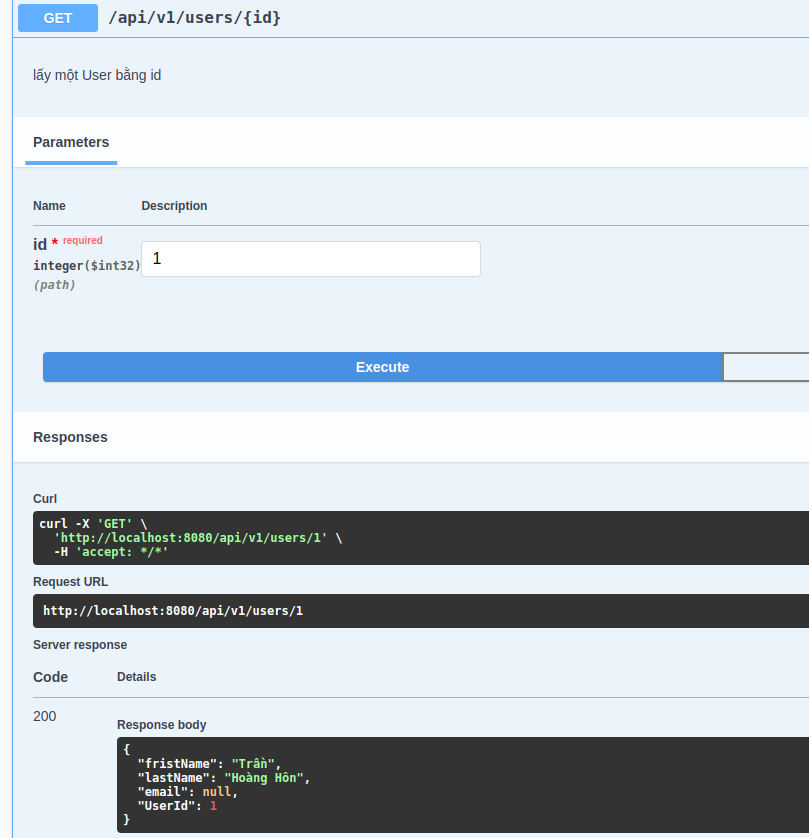
<http://localhost:8080/swagger-ui/index.html>



### Xem danh sách User

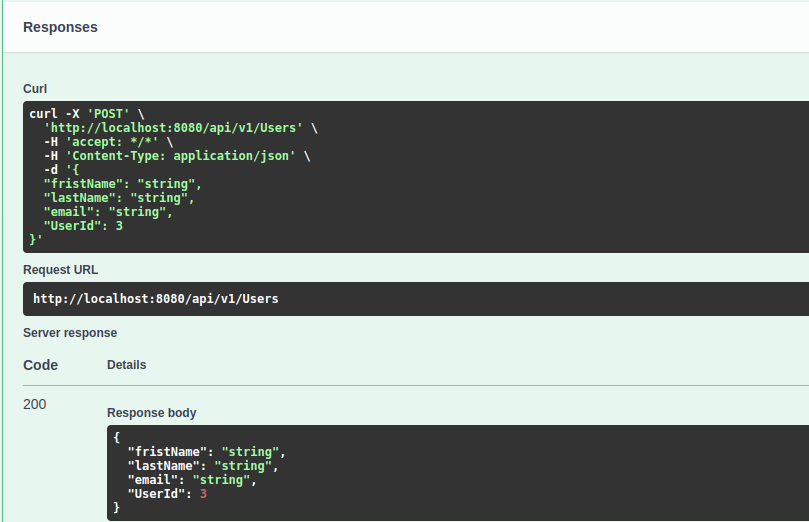


### Lấy một User bằng id

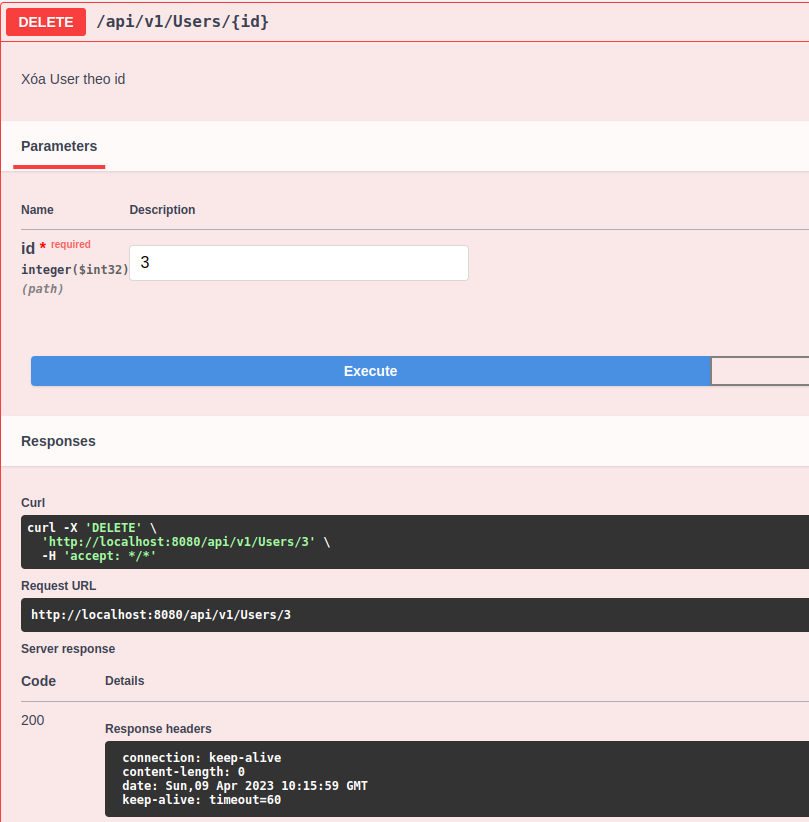


### Tạo một User

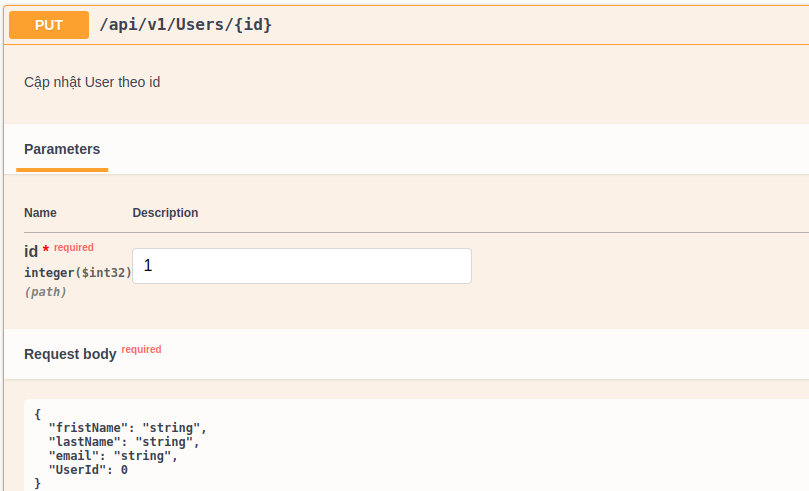




### Xóa một User theo id



### Cập nhật User theo id





### Schemas

