

# TP3 - Spring + GraphQL

- create a category

The screenshot shows the GraphiQL interface for a Spring GraphQL application. A POST request is made to `http://localhost:8080/graphql` with the following mutation:

```
1 mutation {
2   addCategory(input: { name: "Informatique" }) {
3     id
4     name
5   }
6 }
```

The response is a 200 OK status with a response time of 210 ms and 57 B. The preview pane shows the JSON data returned:

```
1 { "data": {
2   "addCategory": {
3     "id": "1",
4     "name": "Informatique"
5   }
6 }}
```

- create a product

The screenshot shows the GraphiQL interface for a Spring GraphQL application. A POST request is made to `http://localhost:8080/graphql` with the following mutation:

```
1 mutation {
2   addProduct(input: { name: "PC Dell", price: 9500, categoryId: 1 }) {
3     name
4     price
5     category { name }
6   }
7 }
```

The response is a 200 OK status with a response time of 554 ms and 92 B. The preview pane shows the JSON data returned:

```
1 { "data": {
2   "addProduct": {
3     "name": "PC Dell",
4     "price": 9500,
5     "category": {
6       "name": "Informatique"
7     }
8   }
9 }}
```

- list of products

POST ▾ http://localhost:8080/graphql

Send	200 OK	137 ms	101 B	Just Now ▾		
Params	Body	Auth	Headers (4)	Scripts	Docs	Preview Headers (3) Cookies Tests 0 / 0 → Mock Console
GraphQL ▾						Preview ▾
Operations schema ↗ <pre> 1 + query { 2 +   products { 3 +     id 4 +     name 5 +     price 6 +     category { 7 +       name 8 +     } 9 +   } 10 } 11 }</pre>						<pre> 1 + { 2 +   "data": [ 3 +     { 4 +       "id": "1", 5 +       "name": "PC Dell", 6 +       "price": 9500.0, 7 +       "category": { 8 +         "name": "Informatique" 9 +       } 10      } 11     ] 12   } 13 }</pre>

- update product

POST ▾ http://localhost:8080/graphql

Send	200 OK	287 ms	75 B	Just Now ▾		
Params	Body	Auth	Headers (4)	Scripts	Docs	Preview Headers (3) Cookies Tests 0 / 0 → Mock Console
GraphQL ▾						Preview ▾
Operations schema ↗ <pre> 1 + mutation { 2 +   updateProduct(id: 1, input: { name: "PC HP updated", price: 8900, categoryId: 1 }) 3 +   { 4 +     id 5 +     name 6 +     price 7 +   } 8 }</pre>						<pre> 1 + { 2 +   "data": { 3 +     "updateProduct": { 4 +       "id": "1", 5 +       "name": "PC HP updated", 6 +       "price": 8900.0 7 +     } 8 +   } 9 }</pre>

- delete product

POST ▾ http://localhost:8080/graphql

Send	200 OK	84 ms	31 B	Just Now ▾		
Params	Body	Auth	Headers (4)	Scripts	Docs	Preview Headers (3) Cookies Tests 0 / 0 → Mock Console
GraphQL ▾						Preview ▾
Operations schema ↗ <pre> 1 + mutation { 2 +   deleteProduct(id: 1) 3 + }</pre>						<pre> 1 + { 2 +   "data": { 3 +     "deleteProduct": true 4 +   } 5 }</pre>