

Questions – Labo 3 – LOG430

Question 1 :

Les méthodes utilisant GET sont sûres et idempotentes, car elles ne modifient pas l'état du système et la répétition de la méthode ne change rien. Les méthodes utilisant POST sont non sûres et non idempotentes, car elles changent l'état du système (elles créent une ressource) et leur répétition crée plusieurs commandes. Enfin, les méthodes utilisant DELETE sont non sûres, car elles changent l'état du système, mais elles sont idempotentes, car leur répétition a le même effet final.

Question 2 :

Dans ce cas, on aurait pu utiliser soit *Simple Relationship Joins* ou *Joins to a Target with an ON Clause*. J'ai décidé d'utiliser la deuxième façon de faire en mentionnant explicitement la condition de liaison, mais cela n'était pas nécessaire. J'aurais pu le faire avec `.join(Stock.product)`.

```
def get_stock_for_all_products():
    """Get stock quantity for all products"""
    session = get_sqlalchemy_session()
    # TODO: ajoutez un join avec Product
    results = session.query(
        Stock.product_id,
        Stock.quantity,
        Product.name,
        Product.sku,
        Product.price,
    ).join(Product, Stock.product_id == Product.id).all()

    stock_data = []
    for row in results:
        stock_data.append({
            'Article': row.name,
            'Numéro SKU': row.sku,
            'Prix unitaire': row.price,
            'Unités en stock': int(row.quantity),
        })

    return stock_data
```

Question 3 :

Cette requête nous donne ce que le *body* demande donc l'*id* et la quantité du produit demandé.

The screenshot shows a GraphQL playground interface. At the top, there is a header with 'POST' and a URL field containing '({baseURL})/stocks/graphql-query'. To the right of the URL is a 'Send' button. Below the header are tabs for 'Docs', 'Params', 'Authorization', 'Headers (9)', 'Body' (which is selected and highlighted in green), 'Scripts', and 'Settings'. Under 'Body', there is a radio button group with options: 'none', 'form-data', 'x-www-form-urlencoded', 'raw', 'binary', and 'GraphQL'. The 'GraphQL' option is selected and has 'Auto Fetch' checked. A green status bar at the bottom of this section says 'Schema Fetched'. In the main area, there is a 'QUERY' text editor with the following code:

```
1 {
2   product(id: "1") {
3     id
4     quantity
5   }
6 }
7
```

To the right of the query editor is a 'GRAPHQL VARIABLES' panel with a single entry '1'. Below the query editor is a navigation bar with 'Body' (selected), 'Cookies', 'Headers (5)', 'Test Results', and a refresh icon. On the far right of the navigation bar are status metrics: '200 OK', '24 ms', '224 B', and a circular progress bar. To the right of these metrics are buttons for 'Save Response' and three dots. The main content area below the navigation bar displays the JSON response:

```
1 {
2   "data": {
3     "product": {
4       "id": 1,
5       "quantity": 56
6     }
7   },
8   "errors": null
9 }
```

Question 4 :

Il faut regrouper les données des tables *Stocks* et *Products* en les liant avec *product_id* mais je n'ai pas été capable.

Question 5 :

Il faudrait voir tous les éléments appartenant aux deux tables donc *id*, *quantity*, *name*, *sku* et *price*

Question 6 :

Il partage les deux le même réseau (labo03-network) ce qui leur permet de communiquer ensemble.

```
1  ⌂ Run All Services
2  services:
3    ⌂ Run Service
4    supplier_app:
5      build: .
6      environment:
7        - PYTHONUNBUFFERED=1
8      volumes:
9        - ./app
10     networks:
11       - labo03-network
12
13   networks:
14     labo03-network:
15       driver: bridge
16       external: true
```

```
1   ◁Run All Services
2 services:
3   ▷Run Service
4     store_manager:
5       build: .
6       env_file:
7         - .env
8       environment:
9         - PYTHONUNBUFFERED=1
10      volumes:
11        - ./app
12      ports:
13        - "5000:5000"
14      networks:
15        - labo03-network
16      depends_on:
17        mysql:
18          condition: service_healthy
19        redis:
20          condition: service_healthy
```