```
mysql> select * from customers;
+------------------+-------+---------+
| email            | fname | lname   |
+------------------+-------+---------+
| gavin@adobe.com  | Gavin | King    |
| kate@adobe.com   | Kate  | Cross   |
| rod@adobe.com    | Rod   | Johnson |
+------------------+-------+---------+
```

```
mysql> select * from products;
+----+-------------+--------+----------+---------+
| id | name        | price  | quantity | version |
+----+-------------+--------+----------+---------+
|  1 | iPhone 15   | 98000  |       99 |       1 |
|  2 | Wacom       | 4500.5 |      100 |       0 |
|  3 | LG AC       | 46000  |       98 |       1 |
|  4 | Sony Bravia | 250000 |      100 |       0 |
|  5 | Tata Play   | 4500   |     2000 |       0 |
+----+-------------+--------+----------+---------+
```

```
mysql> select * from orders;
+-----+-------------------------+--------+-----------------+
| oid | order_date              | total  | customer_fk     |
+-----+-------------------------+--------+-----------------+
|   1 | 2024-04-03 12:22:06.701000 | 190000 | kate@adobe.com  |
+-----+-------------------------+--------+-----------------+
```

Katee

```
mysql> select * from line_items;
+--------+--------+----------+------------+----------+
| itemid | amount | quantity | product_fk | order_fk |
+--------+--------+----------+------------+----------+
|      1 | 92000  |        2 |          3 |        1 |
|      2 | 98000  |        1 |          1 |        1 |
+--------+--------+----------+------------+----------+
```
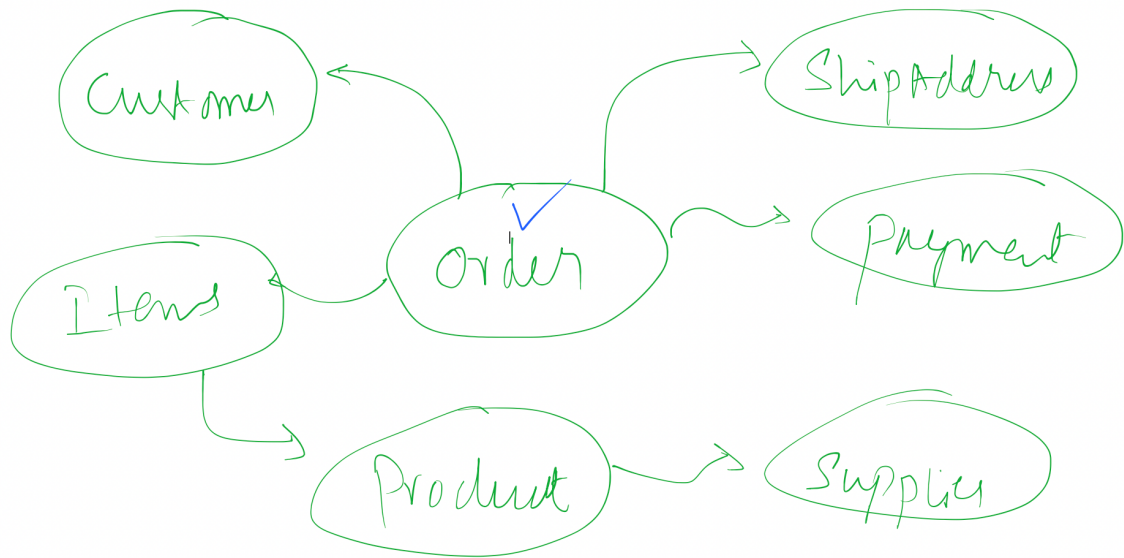


```
mysql> select * from customers;
+-------------------+--------+-------+
| email             | fname  | lname |
+-------------------+--------+-------+
| asha@adobe.com    | Asha   | Rao   |
| harish@adobe.com  | Harish | Sinha |
+-------------------+--------+-------+
```

SQL

```
select * from customers;

select * from customers where fname = 'Harish';

select email, fname from customers;
```

```java
public class Customer {
    @Id
    private String email;

    @Column(name="FNAME", length = 100)
    private String firstName;

    @Column(name="LNAME", length = 100)
    private String lastName;
}
```

JP-QL

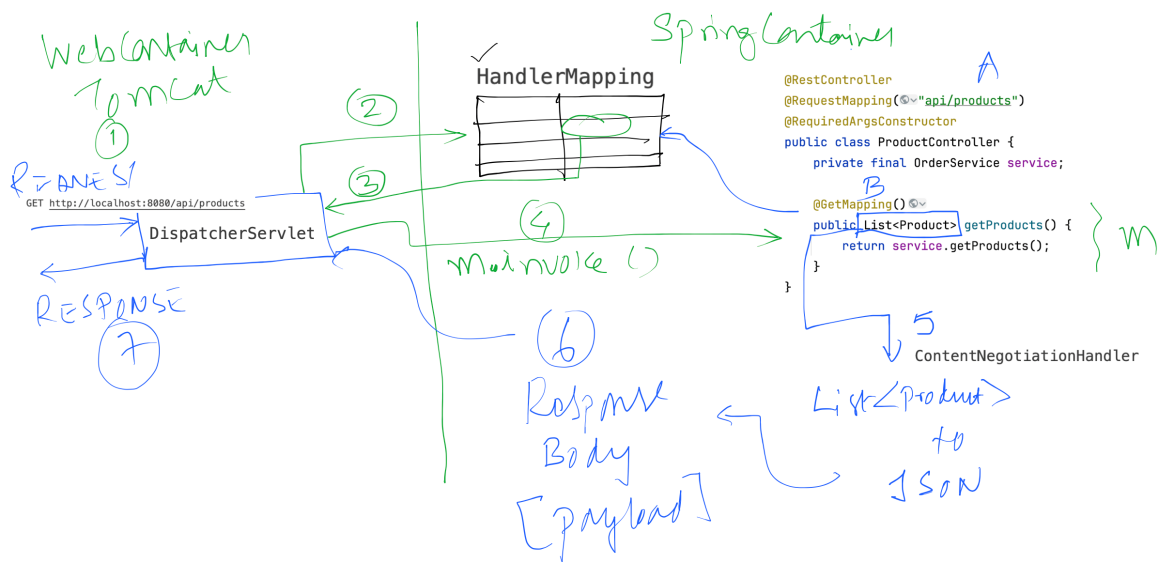```
from Customer

from Customer where firstName = 'Harish'

select email, firstName from Customer
```

WebContainer
Tomcat
①

REQUEST
GET http://localhost:8080/api/products

DispatcherServlet

RESPONSE
⑦

✓ HandlerMapping

②

③

④

MulInvoke ()

⑥

Response
Body
[Payload]

Spring Container

@RestController
@RequestMapping(⌖"api/products")
@RequiredArgsConstructor
public class ProductController {
    private final OrderService service;

    @GetMapping()⌖
    public List<Product> getProducts() {
        return service.getProducts();
    }
}

A

B

M

5
ContentNegotiationHandler

List<Product>
to
JSON

---

| POST ⌄ | http://localhost:8080/api/products |
|---|---|

Params   Authorization   **Headers (9)**   Body   Pre-request Script   Tests   Settings

Headers   👁 7 hidden

| | KEY | VALUE | |
|---|---|---|---|
| ☑ | Accept | application/json | |
| ☑ | Content-Type | application/json | |
| | Key | Value | |

---

| POST ⌄ | http://localhost:8080/api/products | **Send** ⌄ |
|---|---|---|

Params   Authorization   Headers (10)   **Body ●**   Pre-request Script   Tests   Settings               Cookies

○ none  ○ form-data  ○ x-www-form-urlencoded  ● raw  ○ binary  ○ GraphQL    JSON ⌄          Beautify

```
1  {
2      "name": "Macbook Pro",
3      "price": 230000.00,
4      "quantity": 100
5  }
```

Body   Cookies   Headers (5)   Test Results   Status: 201 Created   me: 175 ms   Size: 230 B   Save Response ⌄

Pretty   Raw   Preview   Visualize   JSON ⌄

```
1  {
2      "id": 5,
3      "name": "Macbook Pro",
4      "price": 230000.0,
5      "quantity": 100
6  }
```