springboot1

January 24, 2024

1 SPRINGBOOT

1.1 application.properties

[ ]:

```
spring.datasource.url=jdbc:mysql://localhost:3306/client_order
spring.datasource.username=root
spring.datasource.password=root
spring.jpa.hibernate.ddl-auto=update
```

1.2 Models

1.2.1 –> Client (1)

[ ]:

```
@Entity
@Table(name="clients")
@Data
public class Client {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    Long id;
    String name;
    String lastname;
    Integer age;
}
```

1.2.2 –> Order (M)

[ ]:

```
@Entity
@Table(name="orders")
@Data
public class Order {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    Long id;
    String object;

    @ManyToOne
    @JoinColumn(name = "client_id")
    private Client client_id;
}
```

2 Views

2.0.1 welcome.blade.php

[ ]:

```
<html>
    <body>
        <h2>Welcome</h2>
        <a href="/clients"><button>Clients</button></a>
        <a href="/orders"><button>Orders</button></a>
    </body>
</html>
```

2.0.2 –> Client (1)

[ ]:

```
<html>
    <body>
        <h2>Index</h2>

        <form th:each="c:${clients}">
            <p>Name: <span th:text="${c.name}"></span>
            <button formaction="clients/show" method="GET">Show</button>
            <button formaction="clients/edit" method="GET">Edit</button>
            <input type="hidden" name="id" th:value="${c.id}">
        </form>
        <br>
        <form action="clients/create" method="GET">
            <button>Create new client</button>
        </form>

        <br><a href="../"><button>Back to Welcome</button></a>
    </body>
</html>
```

[ ]:

```
<html>
    <body>
        <h2>Create</h2>

        <form action="/clients/store" method="POST">
```

```
            Name:<input name="name" required><br>
            Lastname:<input name="lastname" required><br>
            Age:<input name="age" required><br><br>
            <button>Create</button>
        </form>

        <br><a href="/clients"><button>Back to home</button></a>
    </body>
</html>
```

[ ]:

```
<html>
    <body>
        <h2>Show</h2>

        <p>ID: <span th:text="${c.id}"></span></p>
        <p>Name: <span th:text="${c.name}"></span></p>
        <p>Lastname: <span th:text="${c.lastname}"></span></p>
        <p>Age: <span th:text="${c.age}"></span></p>
        <br>
        <form action="/clients/destroy" method="POST">
            <button>Delete</button>
            <input type="hidden" name="id" th:value="${c.id}">
        </form>

        <br><a href="/clients"><button>Back to home</button></a>
    </body>
</html>
```

[ ]:

```
<html>
    <body>
        <h2>Edit</h2>

        <form action="/clients/update" method="POST" th:object="${c}">
            <p>Name:<input name="name" th:field="*{name}"></p>
            <p>Lastname:<input name="lastname" th:field="*{lastname}"></p>
            <p>Age:<input name="age" th:field="*{age}"></p><br>
            <button>Update</button>
            <input type="hidden" name="id" th:field="*{id}">
        </form>

        <br><a href="/clients"><button>Back to home</button></a>
    </body>
</html>
```

2.0.3 –> Order (M)

[ ]:

```
<html>
    <body>
        <h2>Index</h2>

        <form th:each="o:${orders}">
            <p>Object: <span th:text="${o.object}"></span></p>
            <button formaction="orders/show" method="GET">Show</button>
            <button formaction="orders/edit" method="GET">Edit</button>
            <input type="hidden" name="id" th:value="${o.id}">
        </form>
        <br>
        <form action="/orders/create"  method="GET">
            <button>Create new order</button>
        </form>

        <br><a href="../"><button>Back to Welcome</button></a>
    </body>
</html>
```

[ ]:

```
<html>
    <body>
        <h2>Create</h2>

        <form action="/orders/store" method="POST">
            Object:<input name="object" required><br>
            Client:
            <select name="client_id" required>
                <div th:each="c:${clients}">
                    <option th:value="${c.id}"><span th:text="${c.name}"></
 ↪span></option>
                </div>
            </select><br><br>
            <button>Create</button>
        </form>

        <br><a href="/orders"><button>Back to home</button></a>
    </body>
</html>
```

[ ]:

```
<html>
    <body>
        <h2>Show</h2>

        <p>ID: <span th:text="${o.id}"></span></p>
        <p>Object: <span th:text="${o.object}"></span></p>
        <p>Client: <span th:text="${o.client_id.name}"></span></p>
        <br>
        <form action="/orders/destroy" method="POST">
            <button>Delete</button>
            <input type="hidden" name="id" th:value="${o.id}">
        </form>

        <br><a href="/orders"><button>Back to home</button></a>
    </body>
</html>
```

[ ]:

```
<html>
    <body>
        <h2>Edit</h2>

        <form action="/orders/update" method="POST" th:object="${o}">
            Object:<input name="name" th:field="*{object}"><br>
            Client:
            <select name="client_id">
                <div th:each="c:${clients}">
                    <option th:value="${c.id}"><span th:text="${c.name}"></
 ↪span></option>
                </div>
            </select><br><br>
            <button>Update</button>
            <input type="hidden" name="id" th:field="*{id}">
        </form>

        <a href="/orders"><button>Back to home</button></a>
    </body>
</html>
```

2.1 Repository

2.1.1 –> Client (1)

public interface ClientRepository extends JpaRepository<Client, Long>

2.1.2 –> Order (M)

public interface OrderRepository extends JpaRepository<Order, Long>

3 Controllers

3.0.1 –> Client (1)

[ ]:

```
@Controller
public class ClientController {

    final ClientRepository repo;

    public ClientController(ClientRepository repo) {
        this.repo = repo;
    }

    @GetMapping("/")
    String welcome(){
        return "welcome";
    }

    @GetMapping("/clients")
    String index(Model model){
        model.addAttribute("clients", repo.findAll());
        return "/clients/index";
    }

    @GetMapping("/clients/create")
    String create(){
       return "/clients/create";
    }

    @PostMapping("/clients/store")
    String store(Client c){
        repo.save(c);
        return "redirect:/clients";
    }

    @GetMapping("/clients/show")
    String show(Long id, Model model){
        model.addAttribute("c", repo.findById(id).orElse(null));
        return "/clients/show";
    }

    @GetMapping("/clients/edit")
    String edit(Long id, Model model){
        model.addAttribute("c", repo.findById(id).orElse(null));
        return "/clients/edit";
    }

    @PostMapping("/clients/update")
    String update(Client c){
        repo.save(c);
        return "redirect:/clients";
    }

    @PostMapping("/clients/destroy")
    String destroy(Long id){
        repo.deleteById(id);
        return "redirect:/clients";
    }
}
```

3.0.2 –> Order (M)

[ ]:

```
@Controller
public class OrderController {
    final OrderRepository repo;
    final ClientRepository clientRepo;

    public OrderController(OrderRepository repo, ClientRepository clientRepo) {
        this.repo = repo;
        this.clientRepo = clientRepo;
    }

    @GetMapping("/orders")
    String index(Model model){
        model.addAttribute("orders", repo.findAll());
        return "/orders/index";
    }

    @GetMapping("/orders/create")
    String create(Model model){
       model.addAttribute("clients", clientRepo.findAll());
       return "/orders/create";
    }

    @PostMapping("/orders/store")
    String store(Order o){
        repo.save(o);
        return "redirect:/orders";
    }

    @GetMapping("/orders/show")
    String show(Long id, Model model){
        model.addAttribute("o", repo.findById(id).orElse(null));
        return "/orders/show";
    }
    @GetMapping("/orders/edit")
    String edit(Long id, Model model){
        model.addAttribute("clients", clientRepo.findAll());
        model.addAttribute("o", repo.findById(id).orElse(null));
        return "/orders/edit";
    }

    @PostMapping("/orders/update")
    String update(Order o){
        repo.save(o);
        return "redirect:/orders";
    }

    @PostMapping("/orders/destroy")
    String destroy(Long id){
        repo.deleteById(id);
        return "redirect:/orders";
    }
}
```