

Using the previously created Customer entity and the new Orders entity, create a one-to-one mapping for both entities.

Attribute for Orders Class (orders.java):

- private Integer orderId
- private String orderDate
- private Double amount

Create repository classes for the Order entity.

Code –

Customer.java(Entity)

```
1 package com.ProductRestApi.model;
2
3 import java.time.Instant;
4
24
25 @Entity
26 public class Customer {
27
28     @Id
29     @GeneratedValue(strategy = GenerationType.IDENTITY)
30     private Long id;
31
32     @NotNull(message = "First name must not be null")
33     @Size(min = 3, message = "First name must contain at least 3 characters")
34     private String firstName;
35
36     @NotNull(message = "Last name must not be null")
37     @Size(min = 3, message = "Last name must contain at least 3 characters")
38     private String lastName;
39
40     @NotNull(message = "Email must not be null")
41     @Email(message = "Email should be valid")
42     private String email;
43
44     @NotNull(message = "Password must not be null")
45     @Pattern(regexp = "^(?=.*[a-zA-Z])(?=.*\\d)[A-Za-z\\d]{6,20}$",
46             message = "Password must be alphanumeric and 6-20 characters long")
47     private String password;
48
49     @NotNull(message = "Confirm password must not be null")
50     @Pattern(regexp = "^(?=.*[a-zA-Z])(?=.*\\d)[A-Za-z\\d]{6,20}$",
51             message = "Password must be alphanumeric and 6-20 characters long")
52     // @Transient // Exclude from persistence since it won't be stored in the database
53     private String confirmPassword;
```

```

55• @NotNull(message = "Mobile number must not be null")
56 @NotEmpty(message = "Mobile number must not be empty")
57 @Pattern(regexp = "\\d{10}", message = "Mobile number must be exactly 10 digits")
58 private String mobileNo;
59
60• @Min(value = 10000, message = "Salary must be at least 10,000")
61 @Max(value = 50000, message = "Salary must not exceed 50,000")
62 private double salary;
63
64 // @CreatedDate
65 // private Instant createdAt; // Automatically set the creation timestamp
66 //
67 // @LastModifiedDate
68 // private Instant modifiedAt; // Automatically set the modification timestamp
69 //
70• @Column(name = "created_at", updatable = false)
71 private Instant createdAt;
72
73• @Column(name = "modified_at")
74 private Instant modifiedAt;
75
76• @NotNull(message = "Address must not be null")
77 private String address; // Added address field
78
79• @PrePersist
80 protected void onCreate() {
81     createdAt = Instant.now();
82 }
83
84• @PreUpdate
85 protected void onUpdate() {
86     modifiedAt = Instant.now();
87 }

```

```

84• @PreUpdate
85 protected void onUpdate() {
86     modifiedAt = Instant.now();
87 }
88
89• @OneToOne(mappedBy = "customer", cascade = CascadeType.ALL)
90 private Orders order;
91
92
93 // Getters and setters
94
95
96• public Long getId() {
97     return id;
98 }
99
100• public void setId(Long id) {
101     this.id = id;
102 }
103
104• public String getFirstName() {
105     return firstName;
106 }
107
108• public void setFirstName(String firstName) {
109     this.firstName = firstName;
110 }
111
112• public String getLastName() {
113     return lastName;
114 }
115

```

```
108• public void setFirstName(String firstName) {
109    this.firstName = firstName;
110}
111
112• public String getLastName() {
113    return lastName;
114}
115
116• public void setLastName(String lastName) {
117    this.lastName = lastName;
118}
119
120• public String getEmail() {
121    return email;
122}
123
124• public void setEmail(String email) {
125    this.email = email;
126}
127
128• public String getPassword() {
129    return password;
130}
131
132• public void setPassword(String password) {
133    this.password = password;
134}
135
136• public String getConfirmPassword() {
137    return confirmPassword;
138}
```

```
40• public void setConfirmPassword(String confirmPassword) {
41    this.confirmPassword = confirmPassword;
42}
43
44• public String getAddress() {
45    return address;
46}
47
48• public void setAddress(String address) {
49    this.address = address;
50}
51
52• public String getMobileNo() {
53    return mobileNo;
54}
55
56• public void setMobileNo(String mobileNo) {
57    this.mobileNo = mobileNo;
58}
59
60• public double getSalary() {
61    return salary;
62}
63
64• public void setSalary(double salary) {
65    this.salary = salary;
66}
67
68• public Instant getCreatedAt() {
69    return createdAt;
70}
71
```

```

168 public Instant getCreatedAt() {
169     return createdAt;
170 }
171
172 public void setCreatedAt(Instant createdAt) {
173     this.createdAt = createdAt;
174 }
175
176 public Instant getModifiedAt() {
177     return modifiedAt;
178 }
179
180 public void setModifiedAt(Instant modifiedAt) {
181     this.modifiedAt = modifiedAt;
182 }
183
184 public Orders getOrder() {
185     return order;
186 }
187
188 public void setOrder(Orders order) {
189     this.order = order;
190 }
191
192 }
193

```

Orders.java(Entity)

```

1 package com.ProductRestApi.model;
2
3 import jakarta.persistence.Entity;
4
10
11 @Entity
12 public class Orders {
13
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     private Integer orderId;
17
18     @NotNull(message = "Order date must not be null")
19     private String orderDate;
20
21     @NotNull(message = "Amount must not be null")
22     private Double amount;
23
24     @OneToOne
25     @JoinColumn(name = "customer_id", referencedColumnName = "id")
26     private Customer customer;
27
28     // Getters and setters
29     public Integer getOrderId() {
30         return orderId;
31     }
32
33     public void setOrderId(Integer orderId) {
34         this.orderId = orderId;
35     }
36
37     public String getOrderDate() {
38         return orderDate;
39     }

```

```

41 public void setOrderDate(String orderDate) {
42     this.orderDate = orderDate;
43 }
44
45 public Double getAmount() {
46     return amount;
47 }
48
49 public void setAmount(Double amount) {
50     this.amount = amount;
51 }
52
53 public Customer getCustomer() {
54     return customer;
55 }
56
57 public void setCustomer(Customer customer) {
58     this.customer = customer;
59 }
60
61 }
62

```

OrdersController.java

```

30 import java.util.List;
17
18 @RestController
19 @RequestMapping("/orders")
20 public class OrdersController {
21
22     @Autowired
23     private OrdersService ordersService;
24
25     @PostMapping
26     public Orders createOrder(@RequestBody Orders order) {
27         return ordersService.saveOrder(order);
28     }
29
30     @GetMapping
31     public List<Orders> getAllOrders() {
32         return ordersService.getAllOrders();
33     }
34
35     @GetMapping("/{orderId}")
36     public ResponseEntity<Orders> getOrderById(@PathVariable Integer orderId) {
37         return ordersService.getOrderById(orderId)
38             .map(ResponseEntity::ok)
39             .orElse(ResponseEntity.notFound().build());
40     }
41
42     @DeleteMapping("/{orderId}")
43     public ResponseEntity<Void> deleteOrder(@PathVariable Integer orderId) {
44         ordersService.deleteOrder(orderId);
45         return ResponseEntity.noContent().build();
46     }
47
48 }

```

OrdersService.java

```
3 import java.util.List;
4 import java.util.Optional;
5
6 import org.springframework.beans.factory.annotation.Autowired;
7 import org.springframework.stereotype.Service;
8
9 import com.ProductRestApi.model.Orders;
10 import com.ProductRestApi.repository.OrdersRepository;
11
12 @Service
13 public class OrdersService {
14
15     @Autowired
16     private OrdersRepository ordersRepository;
17
18     public Orders saveOrder(Orders order) {
19         return ordersRepository.save(order);
20     }
21
22     public List<Orders> getAllOrders() {
23         return ordersRepository.findAll();
24     }
25
26     public Optional<Orders> getOrderById(Integer orderId) {
27         return ordersRepository.findById(orderId);
28     }
29
30     public void deleteOrder(Integer orderId) {
31         ordersRepository.deleteById(orderId);
32     }
33
34 }
```

OrdersRepository.java

```
1 package com.ProductRestApi.repository;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5 import com.ProductRestApi.model.Orders;
6
7 public interface OrdersRepository extends JpaRepository<Orders, Integer>{
8
9 }
10
```