

With reference to the Customer entity class created in the previous assignments, apply @CreateDate and @LastModifiedDate to your Customer entity class.

Hint:

Apply in the following way:

@CreateDate

private Instant CreatedAt;

@LastModifiedDate

private Instant ModifiedAt;

Code

Updated Customer.java (Entity)

```
1 package com.ProductRestApi.model;
2
3 import java.time.Instant;
4
22
23 @Entity
24 public class Customer {
25
26     @Id
27     @GeneratedValue(strategy = GenerationType.IDENTITY)
28     private Long id;
29
30     @NotNull(message = "First name must not be null")
31     @Size(min = 3, message = "First name must contain at least 3 characters")
32     private String firstName;
33
34     @NotNull(message = "Last name must not be null")
35     @Size(min = 3, message = "Last name must contain at least 3 characters")
36     private String lastName;
37
38     @NotNull(message = "Email must not be null")
39     @Email(message = "Email should be valid")
40     private String email;
41
42     @NotNull(message = "Password must not be null")
43     @Pattern(regexp = "^(?=.*[a-zA-Z])(?=.*\\d)[A-Za-z\\d]{6,20}$",
44             message = "Password must be alphanumeric and 6-20 characters long")
45     private String password;
46
47     @NotNull(message = "Confirm password must not be null")
48     @Pattern(regexp = "^(?=.*[a-zA-Z])(?=.*\\d)[A-Za-z\\d]{6,20}$",
49             message = "Password must be alphanumeric and 6-20 characters long")
50     // @Transient // Exclude from persistence since it won't be stored in the database
51     private String confirmPassword;
```

```

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

```

```

84 // Getters and setters
85
86• public Long getId() {
87     return id;
88 }
89
90• public void setId(Long id) {
91     this.id = id;
92 }
93
94• public String getFirstName() {
95     return firstName;
96 }
97
98• public void setFirstName(String firstName) {
99     this.firstName = firstName;
100 }
101
102• public String getLastName() {
103     return lastName;
104 }
105
106• public void setLastName(String lastName) {
107     this.lastName = lastName;
108 }
109
110• public String getEmail() {
111     return email;
112 }
113
114• public void setEmail(String email) {
115     this.email = email;
116 }

```

```

118● public String getPassword() {
119     return password;
120 }
121
122● public void setPassword(String password) {
123     this.password = password;
124 }
125
126● public String getConfirmPassword() {
127     return confirmPassword;
128 }
129
130● public void setConfirmPassword(String confirmPassword) {
131     this.confirmPassword = confirmPassword;
132 }
133
134● public String getMobileNo() {
135     return mobileNo;
136 }
137
138● public void setMobileNo(String mobileNo) {
139     this.mobileNo = mobileNo;
140 }
141
142● public double getSalary() {
143     return salary;
144 }
145
146● public void setSalary(double salary) {
147     this.salary = salary;
148 }
149

```

```

149
150● public Instant getCreatedAt() {
151     return createdAt;
152 }
153
154● public void setCreatedAt(Instant createdAt) {
155     this.createdAt = createdAt;
156 }
157
158● public Instant getModifiedAt() {
159     return modifiedAt;
160 }
161
162● public void setModifiedAt(Instant modifiedAt) {
163     this.modifiedAt = modifiedAt;
164 }
165
166 }
167

```

Output –

HTTP

http://localhost:8080/api/customers

Save

</>

POST

http://localhost:8080/api/customers

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

JSON

Beautify

```
1 {
2   "firstName": "Sourav",
3   "lastName": "Joshi",
4   "email": "sourav@gmail.com",
5   "password": "pass123",
6   "confirmPassword": "pass123",
7   "mobileNo": "1234567890",
8   "salary": 30000
9 }
```

Body

Cookies

Headers (5)

Test Results

200 OK

43 ms

392 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "id": 4,
3   "firstName": "Sourav",
4   "lastName": "Joshi",
5   "email": "sourav@gmail.com",
6   "password": "pass123",
7   "confirmPassword": "pass123",
8   "mobileNo": "1234567890",
9   "salary": 30000.0,
10  "createdAt": "2025-01-02T12:25:12.130198900Z",
11  "modifiedAt": null
12 }
```