

Postman

File Edit View Help

+ New Import Runner

My Workspace Invite

Untitled Request

GET http://localhost:8080/Comment POST http://localhost:8080/friends + ...

No Environment

Send Save

History Collections APIs

+ New Collection

first 0 requests

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

Status: 404 Not Found Time: 9 ms Size: 275 B Save Response

```
1 {
2   "timestamp": 1605677402435,
3   "status": 404,
4   "error": "Not Found",
5   "message": "No message available",
6   "path": "/Comment"
7 }
```

Postman

File Edit View Help

+ New Import Runner

My Workspace Invite

Untitled Request

GET http://localhost:8080/Post POST http://localhost:8080/friends + ...

No Environment

Send Save

History Collections APIs

+ New Collection

first 0 requests

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

Status: 404 Not Found Time: 16 ms Size: 272 B Save Response

```
1 {
2   "timestamp": 1605677554962,
3   "status": 404,
4   "error": "Not Found",
5   "message": "No message available",
6   "path": "/Post"
7 }
```

The screenshot shows the Postman interface. At the top, there are tabs for 'New', 'Import', 'Runner', and 'My Workspace'. Below that is a search bar and a 'Collections' tab which is selected. Under 'Collections', there is a '+ New Collection' button and a 'Trash' section containing one item named 'first'. In the main workspace, there are two requests listed: a 'GET' request to 'http://localhost:8080/User' and a 'POST' request to 'http://localhost:8080/friends'. The 'GET' request is currently selected. It has a 'Params' tab where 'Key' and 'Value' are listed, and a 'Headers' tab with 6 items. The 'Body' tab is selected, showing a JSON response:

```

1 {
2   "timestamp": 160677578217,
3   "status": 404,
4   "error": "Not Found",
5   "message": "No message available",
6   "path": "/User"
7 }

```

At the bottom of the interface, there are tabs for 'Body', 'Cookies', 'Headers (4)', and 'Test Results'. The 'Test Results' tab is active, showing the status: 404 Not Found, Time: 11 ms, Size: 272 B, and a 'Save Response' button.

The screenshot shows the Eclipse IDE interface. The top menu includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help. The left side shows the Package Explorer with a project named 'socialMediaApi' containing packages like com.promineotech.socialMediaApi, com.promineotech.socialMediaApi.controller, com.promineotech.socialMediaApi.entity, com.promineotech.socialMediaApi.repository, and com.promineotech.socialMediaApi.service. The right side shows the Java Console with the following log output:

```

ERROR StatusLogger No log4j2 configuration file found. Using default configuration: logging only errors to the console.

[2020-11-17 22:22:49.695] boot - 11156 INFO [main] --- App: Starting App on DESKTOP-KS690A6 with PID 11156 (C:\Users\Anissa\eclipse-workspace\socialMediaAp
[2020-11-17 22:22:49.721] boot - 11156 INFO [main] --- AnnotationConfigEmbeddedWebApplicationContext: Refreshing org.springframework.boot.context.embedded.
[2020-11-17 22:22:50.154] boot - 11156 INFO [main] --- DefaultListableBeanFactory: Overriding bean definition for bean 'beanNameViewResolver': replacing [R
[2020-11-17 22:22:50.570] boot - 11156 INFO [main] --- PostProcessorRegistrationDelegate$BeanPostProcessorChecker: Bean 'org.springframework.transaction.an
[2020-11-17 22:22:50.584] boot - 11156 INFO [main] --- PostProcessorRegistrationDelegate$BeanPostProcessorChecker: Bean 'transactionAttributeSource' of type [c
[2020-11-17 22:22:50.593] boot - 11156 INFO [main] --- PostProcessorRegistrationDelegate$BeanPostProcessorChecker: Bean 'transactionInterceptor' of type [c
[2020-11-17 22:22:50.593] boot - 11156 INFO [main] --- PostProcessorRegistrationDelegate$BeanPostProcessorChecker: Bean 'org.springframework.transaction.co
[2020-11-17 22:22:50.973] boot - 11156 INFO [main] --- TomcatEmbeddedServletContainer: Tomcat initialized with port(s): 8080 (http)
[2020-11-17 22:22:51.073] boot - 11156 INFO [main] --- StandardService: Starting service Tomcat
[2020-11-17 22:22:51.073] boot - 11156 INFO [main] --- StandardEngine: Starting Servlet Engine: Apache Tomcat/8.0.26
[2020-11-17 22:22:51.164] boot - 11156 INFO [localhost-startStop-1] --- [/]: Initializing Spring embedded WebApplicationContext
[2020-11-17 22:22:51.164] boot - 11156 INFO [localhost-startStop-1] --- ContextLoader: Root WebApplicationContext: initialization completed in 1445 ms
[2020-11-17 22:22:51.576] boot - 11156 INFO [localhost-startStop-1] --- ServletRegistrationBean: Mapping servlet: 'dispatcherServlet' to [/]
[2020-11-17 22:22:51.580] boot - 11156 INFO [localhost-startStop-1] --- FilterRegistrationBean: Mapping filter: 'characterEncodingFilter' to: [/]
[2020-11-17 22:22:51.570] boot - 11156 INFO [localhost-startStop-1] --- FilterRegistrationBean: Mapping filter: 'hiddenHttpMethodFilter' to: [/]
[2020-11-17 22:22:51.745] boot - 11156 INFO [main] --- LocalContainerEntityManagerFactoryBean: Building.jpa container EntityManagerFactory for persistence
[2020-11-17 22:22:51.752] boot - 11156 INFO [main] --- LogHelper: HHH00294: Processing PersistenceUnitInfo [
    name: default
    ...
]
[2020-11-17 22:22:51.803] boot - 11156 INFO [main] --- Version: HHH000412: Hibernate Core (4.3.11.Final)
[2020-11-17 22:22:51.804] boot - 11156 INFO [main] --- Environment: HHH000206: hibernate.properties not found
[2020-11-17 22:22:51.804] boot - 11156 INFO [main] --- Environment: HHH000321: Bytecode provider name: javassist
[2020-11-17 22:22:51.869] boot - 11156 INFO [main] --- Version: HHH000030: Annotations (4.0.5.Final)
[2020-11-17 22:22:52.148] boot - 11156 ERROR [main] --- ConnectionPool: Unable to create initial connections of pool.
com.mysql.jdbc.exceptions.jdbc4.MySQLNonTransientConnectionException: Could not create connection to database server.
at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method) ~[?:1.8.0_261]
at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62) ~[?:1.8.0_261]
at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45) ~[?:1.8.0_261]
at java.lang.reflect.Constructor.newInstance(Constructor.java:423) ~[?:1.8.0_261]

```

eclipse-workspace - socialMediaApi/pom.xml - Eclipse IDE

```
<project>
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.promineotech.socialMediaApi</groupId>
    <artifactId>socialMediaApi</artifactId>
    <version>1.0.0-SNAPSHOT</version>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>1.2.6.RELEASE</version>
    </parent>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
            <exclusions>
                <exclusion>
                    <groupId>org.springframework.boot</groupId>
                    <artifactId>spring-boot-starter-logging</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-log4j2</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <scope>runtime</scope>
        </dependency>
    </dependencies>

```

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

Search Console

```
<terminated> App [8] (Java Application) C:\Program Files\Java\jdk1.8.0_261\bin\javaw.exe (Nov 17, 2020 10:22:48 PM – 10:36:23 PM)
[2020-11-17 22:22:49.695] boot - 11156 INFO [main] --- App: Starting App on DESKTOP-K5690AG6 with PID 11156 (C:\Users\Anissa\eclipse-workspace\socialMediaAp
```

eclipse-workspace - socialMediaApi/src/main/java/com/promineotech/socialMediaApi/App.java - Eclipse IDE

```
package com.promineotech.socialMediaApi;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@ComponentScan("com.promineotech")
@SpringBootApplication
public class App {
    public static void main( String[] args )
    {
        SpringApplication.run(App.class, args);
    }
}
```

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit

socialMediaA... App.java CommentContr... PostControll... UserController... application.... PostService.j... Comment.java Post.java User.java

src/main/java com.promineotech.socialMediaApi > App.java com.promineotech.socialMediaApi.controller > CommentController.java > PostController.java > UserController.java com.promineotech.socialMediaApi.entity > Comment.java > Post.java > User.java com.promineotech.socialMediaApi.repository > CommentRepository.java > PostRepository.java > UserRepository.java com.promineotech.socialMediaApi.service > CommentService.java > PostService.java > UserService.java com.promineotech.socialMediaApi.view > Following.java

src/test/java JRE System Library [jdk1.8.0_261] Maven Dependencies pictures socialMediaApi src main resources application.properties src main test target pom.xml

eclipse-workspace - socialMediaApi/src/main/java/com/promineotech/socialMediaApi/controller/CommentController.java - Eclipse IDE

```
1 package com.promineotech.socialMediaApi.controller;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.http.HttpStatus;
5 import org.springframework.http.ResponseEntity;
6 import org.springframework.web.bind.annotation.PathVariable;
7 import org.springframework.web.bind.annotation.RequestBody;
8 import org.springframework.web.bind.annotation.RequestMapping;
9 import org.springframework.web.bind.annotation.RequestMethod;
10 import org.springframework.web.bind.annotation.RestController;
11
12 import com.promineotech.socialMediaApi.entity.Comment;
13 import com.promineotech.socialMediaApi.service.CommentService;
14
15 @RestController
16 @RequestMapping("/users/{userId}/posts/{postId}/comments")
17 public class CommentController {
18
19     @Autowired
20     private CommentService service;
21
22     public ResponseEntity<Object> createComment(@RequestBody Comment comment, @PathVariable Long userId, @PathVariable Long postId) {
23         try {
24             return new ResponseEntity<Object>(service.createComment(comment, userId, postId), HttpStatus.OK);
25         } catch (Exception e) {
26             return new ResponseEntity<Object>(e.getMessage(), HttpStatus.BAD_REQUEST);
27         }
28     }
29
30     @RequestMapping(value = "{commentId}", method = RequestMethod.DELETE)
31     public ResponseEntity<Object> deleteComment (@PathVariable Long commentId) {
32         service.deleteComment(commentId);
33         return new ResponseEntity<Object>("Delete comment with id: " + commentId, HttpStatus.OK);
34     }
35 }
```

eclipse-workspace - socialMediaApi/src/main/java/com/promineotech/socialMediaApi/controller/CommentController.java - Eclipse IDE

```
1 package com.promineotech.socialMediaApi.controller;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.http.HttpStatus;
5 import org.springframework.http.ResponseEntity;
6 import org.springframework.web.bind.annotation.PathVariable;
7 import org.springframework.web.bind.annotation.RequestBody;
8 import org.springframework.web.bind.annotation.RequestMapping;
9 import org.springframework.web.bind.annotation.RequestMethod;
10 import org.springframework.web.bind.annotation.RestController;
11
12 import com.promineotech.socialMediaApi.entity.Comment;
13 import com.promineotech.socialMediaApi.service.CommentService;
14
15 @RestController
16 @RequestMapping("/users/{userId}/posts/{postId}/comments")
17 public class CommentController {
18
19     @Autowired
20     private CommentService service;
21
22     public ResponseEntity<Object> createComment(@RequestBody Comment comment, @PathVariable Long userId, @PathVariable Long postId) {
23         try {
24             return new ResponseEntity<Object>(service.createComment(comment, userId, postId), HttpStatus.OK);
25         } catch (Exception e) {
26             return new ResponseEntity<Object>(e.getMessage(), HttpStatus.BAD_REQUEST);
27         }
28     }
29
30     @RequestMapping(value = "{commentId}", method = RequestMethod.DELETE)
31     public ResponseEntity<Object> deleteComment (@PathVariable Long commentId) {
32         service.deleteComment(commentId);
33         return new ResponseEntity<Object>("Delete comment with id: " + commentId, HttpStatus.OK);
34     }
35 }
```

eclipse-workspace - socialMediaApi/src/main/java/com/promineotech/socialMediaApi/controller/UserController.java - Eclipse IDE

```
2 import java.nio.file.Files;
3 import java.nio.file.Path;
4 import java.nio.file.Paths;
5 
6 import org.springframework.http.HttpStatus;
7 import org.springframework.http.ResponseEntity;
8 import org.springframework.web.bind.annotationPathVariable;
9 import org.springframework.web.bind.annotation.RequestBody;
10 import org.springframework.web.bind.annotation.RequestMapping;
11 import org.springframework.web.bind.annotation.RequestMethod;
12 import org.springframework.web.bind.annotation.RequestParam;
13 import org.springframework.web.multipart.MultipartFile;
14 
15 import com.promineotech.socialMediaApi.entity.User;
16 import com.promineotech.socialMediaApi.service.UserService;
17 
18 public class UserController {
19     private static String UPLOADED_FOLDER = "./pictures/";
20 
21     @RequestMapping(value = "/register", method = RequestMethod.POST)
22     public ResponseEntity<Object> register(@RequestBody User user) {
23         return new ResponseEntity<Object>(<Object>service.createUser(user), HttpStatus.CREATED);
24     }
25 
26     @RequestMapping(value = "/login", method = RequestMethod.POST)
27     public ResponseEntity<Object> login(@RequestBody User user) {
28         try {
29             return new ResponseEntity<Object>(service.login(user), HttpStatus.OK);
30         } catch (Exception e) {
31             return new ResponseEntity<Object>(e.getMessage(), HttpStatus.BAD_REQUEST);
32         }
33     }
34 
35     @RequestMapping(value = "{id}/follows")
36     public ResponseEntity<Object> showFollowedUsers(@PathVariable long id) {
```

eclipse-workspace - socialMediaApi/src/main/java/com/promineotech/socialMediaApi/controller/UserController.java - Eclipse IDE

```
2 import java.nio.file.Files;
3 import java.nio.file.Path;
4 import java.nio.file.Paths;
5 
6 import org.springframework.http.HttpStatus;
7 import org.springframework.http.ResponseEntity;
8 import org.springframework.web.bind.annotationPathVariable;
9 import org.springframework.web.bind.annotation.RequestBody;
10 import org.springframework.web.bind.annotation.RequestMapping;
11 import org.springframework.web.bind.annotation.RequestMethod;
12 import org.springframework.web.bind.annotation.RequestParam;
13 import org.springframework.web.multipart.MultipartFile;
14 
15 import com.promineotech.socialMediaApi.entity.User;
16 import com.promineotech.socialMediaApi.service.UserService;
17 
18 public class UserController {
19     private static String UPLOADED_FOLDER = "./pictures/";
20 
21     private UserService service;
22 
23     @RequestMapping(value = "/register", method = RequestMethod.POST)
24     public ResponseEntity<Object> register(@RequestBody User user) {
25         return new ResponseEntity<Object>(<Object>service.createUser(user), HttpStatus.CREATED);
26     }
27 
28     @RequestMapping(value = "/login", method = RequestMethod.POST)
29     public ResponseEntity<Object> login(@RequestBody User user) {
30         try {
31             return new ResponseEntity<Object>(service.login(user), HttpStatus.OK);
32         } catch (Exception e) {
33             return new ResponseEntity<Object>(e.getMessage(), HttpStatus.BAD_REQUEST);
34         }
35     }
36 
37     @RequestMapping(value = "{id}/follows")
38     public ResponseEntity<Object> showFollowedUsers(@PathVariable long id) {
```

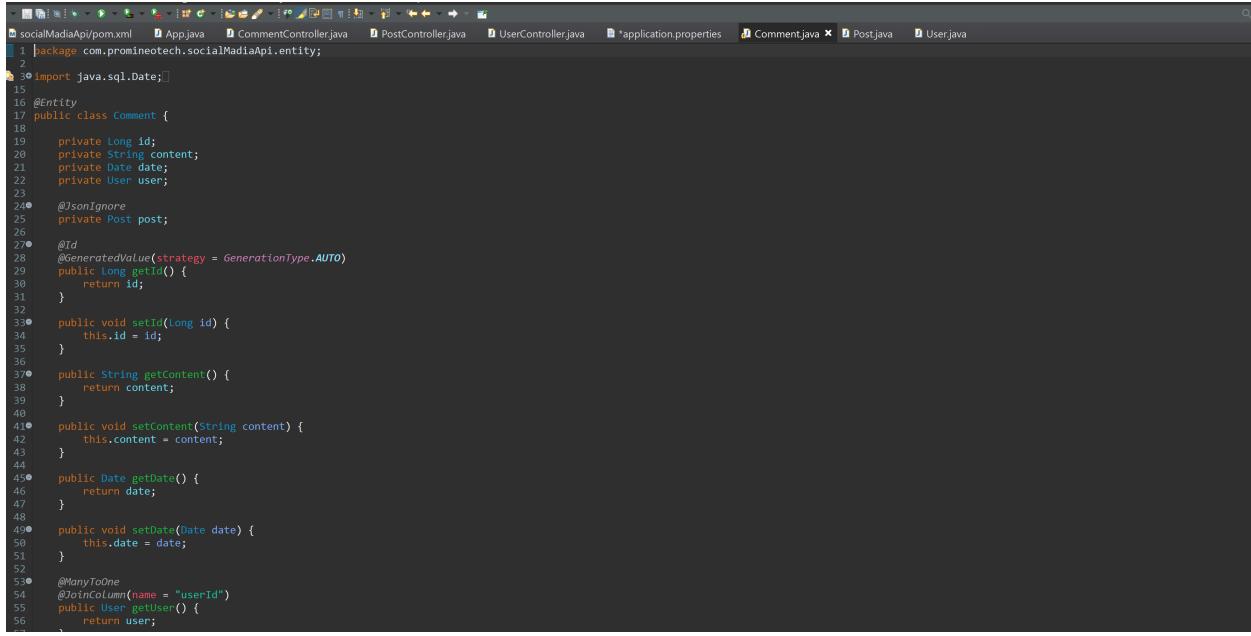
eclipse-workspace - socialMediaApi/src/main/java/com/promineotech/socialMediaApi/controller/UserController.java - Eclipse IDE

```
socialMediaApi
src/main/java
  com.promineotech.socialMediaApi
    App.java
    com.promineotech.socialMediaApi.controller
      CommentController.java
      PostController.java
      UserController.java
    com.promineotech.socialMediaApi.entity
      Comment.java
      Post.java
      User.java
    com.promineotech.socialMediaApi.repository
      CommentRepository.java
      PostRepository.java
      UserRepository.java
  com.promineotech.socialMediaApi.service
    CommentService.java
    PostService.java
    UserService.java
  com.promineotech.socialMediaApi.view
    Following.java
    Following.java
src/test/java
Maven Dependencies
pictures
socialMediaApi
src
  main
    resources
      application.properties
  test
    main
      test
```

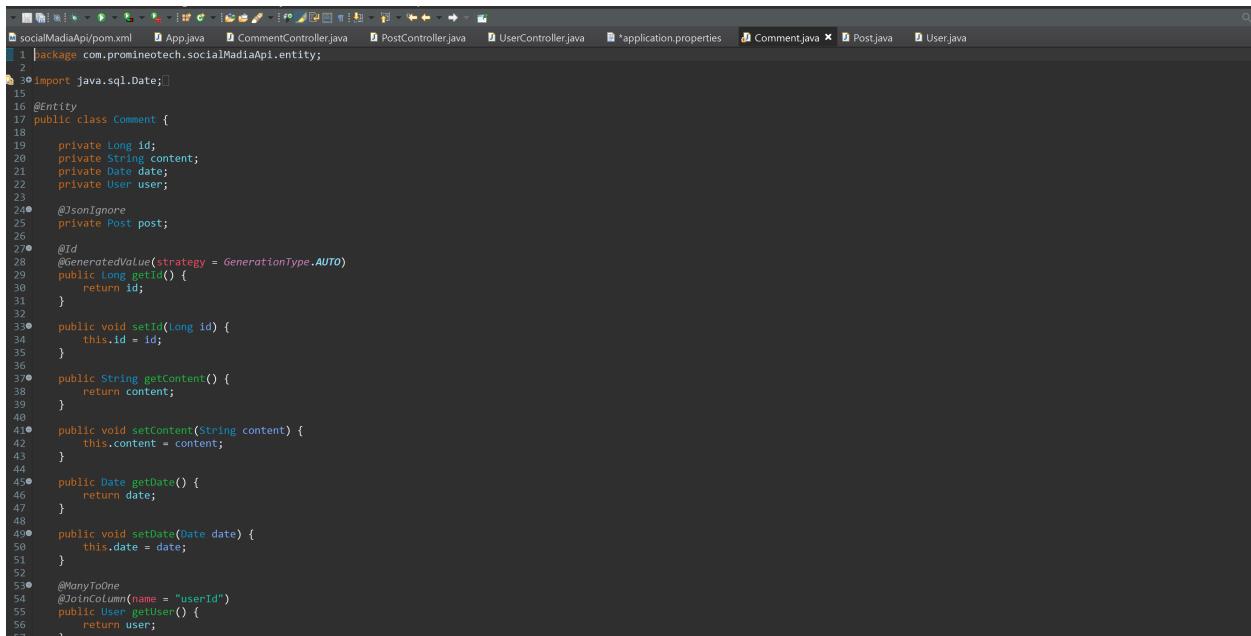
```
25
26  @RequestMapping(value = "/register", method = RequestMethod.POST)
27  public ResponseEntity<Object> register(@RequestBody User user) {
28      return new ResponseEntity<Object>(service.createUser(user), HttpStatus.CREATED);
29  }
30
31
32  @RequestMapping(value = "/login", method = RequestMethod.POST)
33  public ResponseEntity<Object> login(@RequestBody User user) {
34      try {
35          return new ResponseEntity<Object>(service.login(user), HttpStatus.OK);
36      } catch (Exception e) {
37          return new ResponseEntity<Object>(e.getMessage(), HttpStatus.BAD_REQUEST);
38      }
39  }
40
41  @RequestMapping(value = "/{(id)}/follows")
42  public ResponseEntity<Object> showFollowedUsers(@PathVariable Long id) {
43      try {
44          return new ResponseEntity<Object>(service.getFollowedUsers(id), HttpStatus.CREATED);
45      } catch (Exception e) {
46          return new ResponseEntity<Object>(e.getMessage(), HttpStatus.BAD_REQUEST);
47      }
48  }
49
50  @RequestMapping(value = "{(id)}/profilePicture", method = RequestMethod.POST)
51  public ResponseEntity<Object> singleFileUpload(@PathVariable Long id, @RequestParam("file") MultipartFile file) {
52      if (file.isEmpty()) {
53          return new ResponseEntity<Object>("Please upload a file.", HttpStatus.CREATED);
54      }
55      try {
56          String url = UPLOADED_FOLDER + file.getOriginalFilename();
57          byte[] bytes = file.getBytes();
58          Path path = Paths.get(url);
59          Files.write(path, bytes);
60          return new ResponseEntity<Object>(service.updateProfilePicture(id, url), HttpStatus.CREATED);
61      } catch (Exception e) {
62          return new ResponseEntity<Object>(e.getMessage(), HttpStatus.INTERNAL_SERVER_ERROR);
63      }
64  }
65 }
```

eclipse-workspace - socialMediaApi/src/main/resources/application.properties - Eclipse IDE

```
socialMediaApi/pom.xml App.java CommentController.java PostController.java UserController.java *application.properties PostService.java Comment.java Post.java User.java
1 spring.datasource.url=jdbc:mysql://localhost:3306/social_api
2 spring.datasource.username=root
3 spring.datasource.password=root
4
5
6
7 spring.jpa.hibernate.ddl-auto=create
8 spring.jpa.show-sql=true
9 spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect
10
11 multipart.max-file-size=10MB
12 multipart.max-request-size=10MB
```



```
1 package com.promineotech.socialMediaApi.entity;
2
3 import java.sql.Date;
4
5 @Entity
6 public class Comment {
7
8     private Long id;
9     private String content;
10    private Date date;
11    private User user;
12
13    @JsonIgnore
14    private Post post;
15
16    @Id
17    @GeneratedValue(strategy = GenerationType.AUTO)
18    public long getId() {
19        return id;
20    }
21
22    public void setId(long id) {
23        this.id = id;
24    }
25
26    public String getContent() {
27        return content;
28    }
29
30    public void setContent(String content) {
31        this.content = content;
32    }
33
34    public Date getDate() {
35        return date;
36    }
37
38    public void setDate(Date date) {
39        this.date = date;
40    }
41
42    @ManyToOne
43    @JoinColumn(name = "userId")
44    public User getUser() {
45        return user;
46    }
47}
```



```
1 package com.promineotech.socialMediaApi.entity;
2
3 import java.sql.Date;
4
5 @Entity
6 public class Comment {
7
8     private Long id;
9     private String content;
10    private Date date;
11    private User user;
12
13    @JsonIgnore
14    private Post post;
15
16    @Id
17    @GeneratedValue(strategy = GenerationType.AUTO)
18    public long getId() {
19        return id;
20    }
21
22    public void setId(long id) {
23        this.id = id;
24    }
25
26    public String getContent() {
27        return content;
28    }
29
30    public void setContent(String content) {
31        this.content = content;
32    }
33
34    public Date getDate() {
35        return date;
36    }
37
38    public void setDate(Date date) {
39        this.date = date;
40    }
41
42    @ManyToOne
43    @JoinColumn(name = "userId")
44    public User getUser() {
45        return user;
46    }
47}
```

```
eclipse-workspace - socialMediaApi/src/main/java/com/prominotech/socialMediaApi/entity/Post.java - Eclipse IDE
e Edit Source Refactor Navigate Search Project Run Window Help
socialMediaApi/pom.xml App.java CommentController.java PostController.java UserController.java application.properties Comment.java Post.java User.java
1 package com.prominotech.socialMediaApi.entity;
2
3 import java.sql.Date;
4
5 @Entity
6 public class Post {
7
8     private Long id;
9     private String content;
10    private Date date;
11    private User user;
12    private Set<Comment> comments;
13
14    @Id
15    @GeneratedValue(strategy = GenerationType.AUTO)
16    public Long getId() {
17        return id;
18    }
19
20    public void setId(Long id) {
21        this.id = id;
22    }
23
24    public String getContent() {
25        return content;
26    }
27
28    public void setContent(String content) {
29        this.content = content;
30    }
31
32    public Date getDate() {
33        return date;
34    }
35
36    public void setDate(Date date) {
37        this.date = date;
38    }
39
40    @ManyToOne
41    @JoinColumn(name = "userId")
42    public User getUser() {
43        return user;
44    }
45
46    public void setUser(User user) {
```

eclipse-workspace - socialMediaApi/src/main/java/com/promineotech/socialMediaApi/entity/User.java - Eclipse IDE

```
socialMediaApi/pom.xml App.java CommentController.java PostController.java UserController.java application.properties Comment.java Post.java User.java

30 import java.util.Set;
17
18
19 @Entity
20 public class User {
21
22     private Long id;
23     private String username;
24     private String profilePictureUrl;
25
26     @JsonIgnore
27     private Set<User> following;
28
29     private String password;
30
31     @JsonIgnore
32     private Set<Post> posts;
33
34     @JsonIgnore
35     private Set<Comment> comments;
36
37     @Id
38     @GeneratedValue(strategy = GenerationType.AUTO)
39     public long getId() {
40         return id;
41     }
42
43     public void setId(Long id) {
44         this.id = id;
45     }
46
47     public String getUsername() {
48         return username;
49     }
50
51     public void setUsername(String username) {
52         this.username = username;
53     }
54
55     public String getProfilePictureUrl() {
56         return profilePictureUrl;
57     }
58
59     public void setProfilePictureUrl(String profilePictureUrl) {
60         this.profilePictureUrl = profilePictureUrl;
61     }
62 }
```

eclipse-workspace - socialMediaApi/src/main/java/com/promineotech/socialMediaApi/entity/User.java - Eclipse IDE

```
56     return profilePictureUrl;
57   }
58
59  public void setProfilePictureUrl(String profilePictureUrl) {
60    this.profilePictureUrl = profilePictureUrl;
61  }
62
63  @ManyToMany(cascade = CascadeType.ALL)
64  @JoinTable(name = "following",
65    joinColumns = @JoinColumn(name = "userId", referencedColumnName = "id"),
66    inverseJoinColumns = @JoinColumn(name = "followingId", referencedColumnName = "id"))
67  public Set<User> getFollowing() {
68    return following;
69  }
70
71  public void setFollowing(Set<User> following) {
72    this.following = following;
73  }
74
75  @JsonIgnore
76  public String getPassword() {
77    return password;
78  }
79
80  @JsonProperty
81  public void setPassword(String password) {
82    this.password = password;
83  }
84
85  @OneToMany(mappedBy = "user")
86  public Set<Post> getPosts() {
87    return posts;
88  }
89
90  public void setPosts(Set<Post> posts) {
91    this.posts = posts;
92  }
93
94  @OneToMany(mappedBy = "user")
95  public Set<Comment> getComments() {
96    return comments;
97  }
98
99  public void setComments(Set<Comment> comments) {
100    this.comments = comments;
101 }
```

eipse-workspace - socialMediaApi/src/main/java/com/promineotech/socialMediaApi/view/Following.java - Eclipse IDE

```
1 package com.promineotech.socialMediaApi.view;
2
3 import java.util.Set;
4
5 public class Following {
6
7   private Set<User> following;
8
9   public Following(User user) {
10    following = user.getFollowing();
11  }
12
13  public Set<User> getFollowing() {
14    return following;
15  }
16
17  public void setFollowing(Set<User> following) {
18    this.following = following;
19  }
20
21 }
```

Package Explorer

- socialMediaApi
- src/main/java
 - com.promineotech.socialMediaApi
 - App.java
 - CommentController.java
 - PostController.java
 - UserController.java
 - com.promineotech.socialMediaApi.entity
 - Comment.java
 - Post.java
 - User.java
 - com.promineotech.socialMediaApi.repository
 - CommentRepository.java
 - PostRepository.java
 - UserRepository.java
 - com.promineotech.socialMediaApi.service
 - CommentService.java
 - PostService.java
 - UserService.java
 - com.promineotech.socialMediaApi.view
 - Following.java
- src/test/java
- JRE System Library [jdk1.8.0_261]
- Maven Dependencies
- pictures
- socialMediaApi
- src
 - main
 - resources
 - application.properties
 - src
 - main
 - test
- target
- pom.xml

eclipse-workspace - socialMediaApi/src/main/java/com/promineotech/socialMediaApi/repository/UserRepository.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X JUnit

socialMediaApi

- src/main/java
 - com.promineotech.socialMediaApi
 - App.java
 - CommentController.java
 - PostController.java
 - UserController.java
 - com.promineotech.socialMediaApi.entity
 - Comment.java
 - Post.java
 - User.java
 - com.promineotech.socialMediaApi.repository
 - CommentRepository.java
 - PostRepository.java
 - UserRepository.java
 - com.promineotech.socialMediaApi.service
 - CommentService.java
 - PostService.java
 - UserService.java
 - com.promineotech.socialMediaApi.view
 - Following.java
- src/test/java
- JRE System Library [jdk1.8.0_261]
- Maven Dependencies
- pictures
- socialMediaApi
- src
 - main
 - resources
 - application.properties
 - src
 - src
 - main
 - test
 - target
- pom.xml

CommentService.java PostService.java UserService.java CommentRepository.java PostRepository.java UserRepository.java Following.java

```
1 package com.promineotech.socialMediaApi.repository;
2
3 import org.springframework.data.repository.CrudRepository;
4
5 public interface UserRepository extends CrudRepository<User, Long>{
6
7     public User findByUsername(String username);
8
9 }
```

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X JUnit

socialMediaApi

- src/main/java
 - com.promineotech.socialMediaApi
 - App.java
 - CommentController.java
 - PostController.java
 - UserController.java
 - com.promineotech.socialMediaApi.entity
 - Comment.java
 - Post.java
 - User.java
 - com.promineotech.socialMediaApi.repository
 - CommentRepository.java
 - PostRepository.java
 - UserRepository.java
 - com.promineotech.socialMediaApi.service
 - CommentService.java
 - PostService.java
 - UserService.java
 - com.promineotech.socialMediaApi.view
 - Following.java
- src/test/java
- JRE System Library [jdk1.8.0_261]
- Maven Dependencies
- pictures
- socialMediaApi
- src
 - main
 - resources
 - application.properties
 - src
 - src
 - main
 - test
 - target

CommentService.java PostService.java UserService.java CommentRepository.java PostRepository.java UserRepository.java Following.java

```
1 package com.promineotech.socialMediaApi.repository;
2
3 import org.springframework.data.repository.CrudRepository;
4
5 public interface PostRepository extends CrudRepository<Post, Long> {
6
7 }
8
9 }
```

```
1 package com.promineotech.socialMediaApi.repository;
2
3 import org.springframework.data.repository.CrudRepository;
4
5 public interface CommentRepository extends CrudRepository<Comment, Long> {
6
7 }
8
9
10 }
```

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure under "socialMediaApi".
- CommentRepository.java Tab:** The current file being edited.
- Code:** The code implements a CrudRepository for the Comment entity.

eclipse-workspace - socialMediaApi/src/main/java/com/promineotech/socialMediaApi/service/UserService.java - Eclipse IDE

```
1 import org.springframework.beans.factory.annotation.Autowired;
2
3 @Service
4 public class UserService {
5
6     @Autowired
7     private UserRepository repo;
8
9     public User createUser(User user) {
10         return repo.save(user);
11     }
12
13     public User login(User user) throws Exception {
14         User foundUser = repo.findByUsername(user.getUsername());
15         if (foundUser == null && foundUser.getPassword().equals(user.getPassword())) {
16             return foundUser;
17         } else {
18             throw new Exception("Invalid username or password");
19         }
20     }
21
22     public Following follow(Long userId, Long followId) throws Exception {
23         User user = repo.findOne(userId);
24         User follow = repo.findOne(followId);
25         if (user == null || follow == null) {
26             throw new Exception("User does not exist.");
27         }
28         user.getFollowing().add(follow);
29         repo.save(user);
30         return new Following(user);
31     }
32
33     public Following getFollowedUsers(Long userId) throws Exception {
34         User user = repo.findOne(userId);
35         if (user == null) {
36             throw new Exception("User does not exist.");
37         }
38         return new Following(user);
39     }
40
41     public User updateProfilePicture(Long userId, String url) throws Exception {
42         User user = repo.findOne(userId);
43
44 }
```

se-workspace - socialMediaApi/src/main/java/com/promineotech/socialMediaApi/service/PostService.java - Eclipse IDE

```
1 package com.promineotech.socialMediaApi.service;
2
3 import java.util.Date;
4
5 @Service
6 public class PostService {
7
8     @Autowired
9     private PostRepository repo;
10
11     @Autowired
12     private UserRepository userRepo;
13
14     public Iterable<Post> getAllPost() {
15         return repo.findAll();
16     }
17
18     public Post getPost(long id) {
19         return repo.findOne(id);
20     }
21
22     public Post updatePost(Post post, long id) throws Exception {
23         Post foundPost = repo.findOne(id);
24         if (foundPost == null) {
25             throw new Exception("Post not found.");
26         }
27         foundPost.setContent(post.getContent());
28         return repo.save(foundPost);
29     }
30
31     public Post createPost(Post post, Long userId) throws Exception {
32         User user = userRepo.findOne(userId);
33         if(user == null) {
34             throw new Exception ("User not found.");
35         }
36         post.setDate((java.sql.Date) new Date());
37         post.setUser(user);
38         return repo.save(post);
39     }
40
41 }
```

src/main/java
com.promineotech.socialMediaApi
 App.java
 CommentController.java
 PostController.java
 UserController.java
com.promineotech.socialMediaApi.entity
 Comment.java
 Post.java
 User.java
com.promineotech.socialMediaApi.repository
 CommentRepository.java
 PostRepository.java
 UserRepository.java
com.promineotech.socialMediaApi.service
 CommentService.java
 PostService.java
 UserService.java
com.promineotech.socialMediaApi.view
 Following.java
src/test/java
JRE System Library [jdk1.8.0_261]
Maven Dependencies
pictures
socialMediaApi
src
 main
 resources
 application.properties
 src
 test
target
pom.xml

lipse-workspace - socialMediaApi/src/main/java/com/promineotech/socialMediaApi/service/CommentService.java - Eclipse IDE

```
1 package com.promineotech.socialMediaApi.service;
2
3 import java.util.Date;
4
5 @Service
6 public class CommentService {
7
8     @Autowired
9     private CommentRepository repo;
10
11     @Autowired
12     private PostRepository postRepo;
13
14     @Autowired
15     private UserRepository userRepo;
16
17     public Comment createComment(Comment comment, Long userId, Long postId) throws Exception {
18         User user = userRepo.findOne(userId);
19         Post post = postRepo.findOne(postId);
20         if (user == null || post == null) {
21             throw new Exception("User or Post does not exists.");
22         }
23         comment.setDate((java.sql.Date) new Date());
24         comment.setUserId(user);
25         comment.setPost(post);
26         return repo.save(comment);
27     }
28
29     public void deleteComment(long commentId) {
30         repo.deleteById(commentId);
31     }
32
33 }
```

Eclipse Explorer - socialMediaApi/src/main/java/com/promineotech/socialMediaApi/service/UserService.java - Eclipse IDE

```
File Source Refactor Navigate Search Project Run Window Help
File Explorer JUnit
CommentService.java PostService.java UserService.java CommentRepository.java PostRepository.java UserRepository.java Following.java
socialMediaApi
src/main/java
com.promineotech.socialMediaApi
App.java
com.promineotech.socialMediaApi.controller
CommentController.java
PostController.java
UserController.java
com.promineotech.socialMediaApi.entity
Comment.java
Post.java
User.java
com.promineotech.socialMediaApi.repository
CommentRepository.java
PostRepository.java
UserRepository.java
com.promineotech.socialMediaApi.service
CommentService.java
PostService.java
UserService.java
com.promineotech.socialMediaApi.view
Following.java
src/test/java
Maven Dependencies
pictures
socialMediaApi
src
main
resources
application.properties
src
main
test
```

```
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 @Service
6 public class UserService {
7
8     @Autowired
9     private UserRepository repo;
10
11     public User createUser(User user) {
12         return repo.save(user);
13     }
14
15     public User login(User user) throws Exception {
16         User foundUser = repo.findByUsername(user.getUsername());
17         if (foundUser != null && foundUser.getPassword().equals(user.getPassword())) {
18             return foundUser;
19         } else {
20             throw new Exception("Invalid username or password");
21         }
22     }
23
24     public Following follow(long userId, Long followId) throws Exception {
25         User user = repo.findOne(userId);
26         User follow = repo.findOne(followId);
27         if (user == null || follow == null) {
28             throw new Exception("User does not exist.");
29         }
30         user.getFollowing().add(follow);
31         repo.save(user);
32         return new Following(user);
33     }
34
35     public Following getFollowedUsers(long userId) throws Exception {
36         User user = repo.findOne(userId);
37         if (user == null) {
38             throw new Exception("User does not exist.");
39         }
40         return new Following(user);
41     }
42
43     public User updateProfilePicture(long userId, String url) throws Exception {
44         User user = repo.findOne(userId);
45         user.setProfilePicture(url);
46         return user;
47     }
}
```