



Web API Design with SpringBoot Week 1 Coding Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Instructions: In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your Java project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

Follow the tutorial in the homework section to build an API. Paste screenshots of your code and your postman requests and responses to show the API works. Push your project to GitHub and paste the link below.

Screenshots of Code:



PROMINEO TECH

```
1 package com.promineotech.socialMediaApi;
2
3 import org.springframework.boot.SpringApplication;
4
5
6
7 @ComponentScan("com.promineotech.socialMediaApi")
8 @SpringBootApplication
9 public class App {
10     public static void main(String[] args) {
11         SpringApplication.run(App.class, args);
12     }
13 }
```

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



PROMINEO TECH

```
1 package com.promineotech.socialMediaApi.service;
2
3+ import java.util.Date;
12
13 @Service
14 public class PostService {
15
16-     @Autowired
17     private PostRepository repo;
18
19-     @Autowired
20     private UserRepository userRepo;
21
22-     public Iterable<Post> getAllPosts() {
23         return repo.findAll();
24     }
25
26-     public Post getPost(Long id) {
27         return repo.findOne(id);
28     }
29
30-     public Post updatePost(Post post, Long id) throws Exception {
31         Post foundPost = repo.findOne(id);
32         if (foundPost == null) {
33             throw new Exception("Post not found.");
34         }
35         foundPost.setContent(post.getContent());
36         return repo.save(foundPost);
37     }
38
39-     public Post createPost(Post post, Long userId) throws Exception {
40         User user = userRepo.findOne(userId);
41         if (user == null) {
42             throw new Exception("User not found.");
43         }
44         post.setDate(new Date());
45         post.setUser(user);
46         return repo.save(post);
47     }
48 }
49
```



PROMINEO TECH

```
1 package com.promineotech.socialMediaApi.service;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5
6
7
8
9
10 @Service
11 public class UserService {
12
13     @Autowired
14     private UserRepository repo;
15
16     public User createUser(User user) {
17         return repo.save(user);
18     }
19
20     public User login(User user) throws Exception {
21         User foundUser = repo.findByUsername(user.getUsername());
22         if (foundUser != null && foundUser.getPassword().equals(user.getPassword())) {
23             return foundUser;
24         } else {
25             throw new Exception("Invalid username or password.");
26         }
27     }
28
29     public Following follow(Long userId, Long followId) throws Exception {
30         User user = repo.findOne(userId);
31         User follow = repo.findOne(followId);
32         if (user == null || follow == null) {
33             throw new Exception("User does not exist.");
34         }
35         user.getFollowing().add(follow);
36         repo.save(user);
37         return new Following(user);
38     }
39
40     public Following getFollowedUsers(Long userId) throws Exception {
41         User user = repo.findOne(userId);
42         if (user == null) {
43             throw new Exception("User does not exist");
44         }
45         return new Following(user);
46     }
47
48     public User updateProfilePicture(Long userId, String url) throws Exception {
49         User user = repo.findOne(userId);
50         if (user == null) {
51             throw new Exception("User does not exist.");
52         }
53         user.setProfilePictureUrl(url);
54         return repo.save(user);
55     }
56
57 }
58
```



PROMINEO TECH

```
1 package com.promineotech.socialMediaApi.view;
2
3 import java.util.Set;
4
5
6
7 public class Following {
8
9     private Set<User> following;
10
11     public Following(User user) {
12         following = user.getFollowing();
13     }
14
15     public Set<User> getFollowing() {
16         return following;
17     }
18
19     public void setFollowing(Set<User> following) {
20         this.following = following;
21     }
22 }
23
```

```
1 spring.datasource.url=jdbc:mysql://localhost/social_api
2 spring.datasource.username=root
3 spring.datasource.password=root1234
4 spring.datasource.driver-class-name=com.mysql.jdbc.Driver
5
6 spring.jpa.hibernate.ddl-auto=create
7 spring.jpa.show-sql=true
8 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
9
10 multipart.max-file-size=10MB
11 multipart.max-request-size=10MB
12
```



PROMINEO TECH

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <project xmlns="http://maven.apache.org/POM/4.0.0"
4         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"
6         <modelVersion>4.0.0</modelVersion>
7         <groupId>com.promineotech</groupId>
8         <artifactId>socialMediaApi</artifactId>
9         <version>0.0.1-SNAPSHOT</version>
10
11         <name>socialMediaApi</name>
12         <!-- FIXME change it to the project's website -->
13         <url>http://www.maven.apache.org</url>
14
15     <parent>
16         <groupId>org.springframework.boot</groupId>
17         <artifactId>spring-boot-starter-parent</artifactId>
18         <version>1.2.6.RELEASE</version>
19     </parent>
20
21     <properties>
22         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
23         <maven.compiler.source>1.7</maven.compiler.source>
24         <maven.compiler.target>1.7</maven.compiler.target>
25     </properties>
26
27     <dependencies>
28         <dependency>
29             <groupId>io.jsonwebtoken</groupId>
30             <artifactId>jjwt-api</artifactId>
31             <version>0.10.5</version>
32         </dependency>
33         <dependency>
34             <groupId>io.jsonwebtoken</groupId>
35             <artifactId>jjwt-impl</artifactId>
36             <version>0.10.5</version>
37         </dependency>
38         <dependency>
39             <groupId>io.jsonwebtoken</groupId>
40             <artifactId>jjwt-jackson</artifactId>
41             <version>0.10.5</version>
42         </dependency>
43         <dependency>
44             <groupId>org.springframework.boot</groupId>
45             <artifactId>spring-boot-starter-web</artifactId>
46             <exclusions>
47                 <exclusion>
48                     <groupId>org.springframework.boot</groupId>
49                     <artifactId>spring-boot-starter-logging</artifactId>
50                 </exclusion>
51             </exclusions>
52         </dependency>
53         <dependency>
54             <groupId>org.springframework.boot</groupId>
55             <artifactId>spring-boot-starter-log4j2</artifactId>
56         </dependency>
```



PROMINEO TECH

```
57 </dependencies>
58 <dependency>
59   <groupId>org.springframework.boot</groupId>
60   <artifactId>spring-boot-starter-test</artifactId>
61   <scope>test</scope>
62 </dependency>
63 <dependency>
64   <groupId>mysql</groupId>
65   <artifactId>mysql-connector-java</artifactId>
66   <version>8.0.13</version>
67   <scope>runtime</scope>
68 </dependency>
69 <dependency>
70   <groupId>com.jayway.jsonpath</groupId>
71   <artifactId>json-path</artifactId>
72   <scope>test</scope>
73 </dependency>
74 <dependency>
75   <groupId>org.springframework.boot</groupId>
76   <artifactId>spring-boot-starter-data-jpa</artifactId>
77 </dependency>
78 <dependency>
79   <groupId>org.springframework.security</groupId>
80   <artifactId>spring-security-crypto</artifactId>
81 </dependency>
82 <dependency>
83   <groupId>junit</groupId>
84   <artifactId>junit</artifactId>
85   <scope>test</scope>
86 </dependency>
87 </dependencies>
88 <build>
89   <pluginManagement><!-- lock down plugins versions to avoid using Maven
90     defaults (may be moved to parent pom) -->
91   <plugins>
92     <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean_Lifecycle -->
93     <plugin>
94       <artifactId>maven-clean-plugin</artifactId>
95       <version>3.1.0</version>
96     </plugin>
97     <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bindings.html#Default_Bindings -->
98     <plugin>
99       <artifactId>maven-resources-plugin</artifactId>
100      <version>3.0.2</version>
101    </plugin>
102    <plugin>
103      <artifactId>maven-compiler-plugin</artifactId>
104      <version>3.8.0</version>
105    </plugin>
106    <plugin>
107      <artifactId>maven-surefire-plugin</artifactId>
108      <version>2.22.1</version>
109    </plugin>
110    <plugin>
111      <artifactId>maven-jar-plugin</artifactId>
112      <version>3.0.2</version>
113    </plugin>
114    <plugin>
115      <artifactId>maven-install-plugin</artifactId>
116      <version>2.5.2</version>
117    </plugin>
118    <plugin>
119      <artifactId>maven-deploy-plugin</artifactId>
120      <version>2.8.2</version>
121    </plugin>
122    <!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#site_Lifecycle -->
123    <plugin>
124      <artifactId>maven-site-plugin</artifactId>
125      <version>3.7.1</version>
126    </plugin>
127    <plugin>
128      <artifactId>maven-project-info-reports-plugin</artifactId>
129      <version>3.0.0</version>
130    </plugin>
131  </plugins>
132 </pluginManagement>
133 </build>
</project>
```

Talking:

Talking:



PROMINEO TECH

```
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.Post;
13 import com.promineotech.socialMediaApi.service.PostService;
14
15 @RestController
16 @RequestMapping("/users/{userId}/posts")
17 public class PostController {
18
19     @Autowired
20     private PostService service;
21
22     @RequestMapping(method = RequestMethod.GET)
23     public ResponseEntity<Object> getAllPosts() {
24         return new ResponseEntity<Object>(service.getAllPosts(), HttpStatus.OK);
25     }
26
27     @RequestMapping(value = "/{postId}", method = RequestMethod.GET)
28     public ResponseEntity<Object> getPost(@PathVariable Long postId) {
29         return new ResponseEntity<Object>(service.getPost(postId), HttpStatus.OK);
30     }
31
32     @RequestMapping(value = "/{postId}", method = RequestMethod.PUT)
33     public ResponseEntity<Object> updatePost(@RequestBody Post post, @PathVariable Long postId) {
34         try {
35             return new ResponseEntity<Object>(service.updatePost(post, postId), HttpStatus.OK);
36         } catch (Exception e) {
37             return new ResponseEntity<Object>(e.getMessage(), HttpStatus.BAD_REQUEST);
38         }
39     }
40
41     @RequestMapping(method = RequestMethod.POST)
42     public ResponseEntity<Object> createPost(@RequestBody Post post, @PathVariable Long userId) {
43         try {
44             return new ResponseEntity<Object>(service.createPost(post, userId), HttpStatus.OK);
45         } catch (Exception e) {
46             return new ResponseEntity<Object>(e.getMessage(), HttpStatus.BAD_REQUEST);
47         }
48     }
49 }
50
```




PROMINEO TECH

```
1 package com.promineotech.socialMediaApi.controller;
2
3 import java.nio.file.Files;
4 import java.nio.file.Path;
5 import java.nio.file.Paths;
6
7 import org.springframework.beans.factory.annotation.Autowired;
8 import org.springframework.http.HttpStatus;
9 import org.springframework.http.ResponseEntity;
10 import org.springframework.web.bind.annotation.PathVariable;
11 import org.springframework.web.bind.annotation.RequestBody;
12 import org.springframework.web.bind.annotation.RequestMapping;
13 import org.springframework.web.bind.annotation.RequestMethod;
14 import org.springframework.web.bind.annotation.RequestParam;
15 import org.springframework.web.bind.annotation.RestController;
16 import org.springframework.web.multipart.MultipartFile;
17
18 import com.promineotech.socialMediaApi.entity.User;
19 import com.promineotech.socialMediaApi.service.UserService;
20
21 @RestController
22 @RequestMapping("/users")
23 public class UserController {
24
25     private static String UPLOADED_FOLDER = "./pictures/";
26
27     @Autowired
28     private UserService service;
29
30     @RequestMapping(value = "/register", method = RequestMethod.POST)
31     public ResponseEntity<Object> register(@RequestBody User user) {
32         return new ResponseEntity<Object>(service.createUser(user), HttpStatus.CREATED);
33     }
34
35     @RequestMapping(value = "/login", method = RequestMethod.POST)
36     public ResponseEntity<Object> login(@RequestBody User user) {
37         try {
38             return new ResponseEntity<Object>(service.login(user), HttpStatus.OK);
39         } catch (Exception e) {
40             return new ResponseEntity<Object>(e.getMessage(), HttpStatus.BAD_REQUEST);
41         }
42     }
43
44     @RequestMapping(value =("/{id}/follows")
45     public ResponseEntity<Object> showFollowedUsers(@PathVariable Long id) {
46         try {
47             return new ResponseEntity<Object>(service.getFollowedUsers(id), HttpStatus.CREATED);
48         } catch (Exception e) {
49             return new ResponseEntity<Object>(e.getMessage(), HttpStatus.BAD_REQUEST);
50         }
51     }
52
53     @RequestMapping(value =("/{id}/follows/{followId}", method = RequestMethod.POST)
54     public ResponseEntity<Object> follow(@PathVariable Long id, @PathVariable Long followId) {
55         try {
56             return new ResponseEntity<Object>(service.follow(id, followId), HttpStatus.CREATED);
57         } catch (Exception e) {
58             return new ResponseEntity<Object>(e.getMessage(), HttpStatus.BAD_REQUEST);
59         }
60     }
61
62     @RequestMapping(value =("/{id}/profilePicture", method = RequestMethod.POST)
63     public ResponseEntity<Object> singleFileUpload(@PathVariable Long id, @RequestParam("file") MultipartFile file) {
64         if (file.isEmpty()) {
65             return new ResponseEntity<Object>("Please upload a file.", HttpStatus.BAD_REQUEST);
66         }
67
68         try {
69             String url = UPLOADED_FOLDER + file.getOriginalFilename();
70             byte[] bytes = file.getBytes();
71             Path path = Paths.get(url);
72             Files.write(path, bytes);
73             return new ResponseEntity<Object>(service.updateProfilePicture(id, url), HttpStatus.CREATED);
74         } catch (Exception e) {
75             return new ResponseEntity<Object>(e.getMessage(), HttpStatus.INTERNAL_SERVER_ERROR);
76         }
77     }
78 }
79
```



PROMINEO TECH

```
1 package com.promineotech.socialMediaApi.entity;
2
3 import java.util.Date;
4
5 import javax.persistence.Entity;
6 import javax.persistence.GeneratedValue;
7 import javax.persistence.GenerationType;
8 import javax.persistence.Id;
9 import javax.persistence.JoinColumn;
10 import javax.persistence.ManyToOne;
11
12 import com.fasterxml.jackson.annotation.JsonIgnore;
13 import com.promineotech.socialMediaApi.entity.Post;
14 import com.promineotech.socialMediaApi.entity.User;
15
16 @Entity
17 public class Comment {
18
19     private Long id;
20     private String content;
21     private Date date;
22     private User user;
23
24     @JsonIgnore
25     private Post post;
26
27     @Id
28     @GeneratedValue(strategy = GenerationType.AUTO)
29     public Long getId() {
30         return id;
31     }
32
33     public void setId(Long id) {
34         this.id = id;
35     }
36
37     public String getContent() {
38         return content;
39     }
40
41     public void setContent(String content) {
42         this.content = content;
43     }
44
45     public Date getDate() {
46         return date;
47     }
48
49     public void setDate(Date date) {
50         this.date = date;
51     }
52
53     @ManyToOne
54     @JoinColumn(name = "postId")
55     public Post getPost() {
56         return post;
57     }
58
59     public void setPost(Post post) {
60         this.post = post;
61     }
62
63     @ManyToOne
64     @JoinColumn(name = "userId")
65     public User getUser() {
66         return user;
67     }
68
69     public void setUser(User user) {
70         this.user = user;
71     }
72
73 }
```

```
1 package com.promineotech.socialMediaApi.entity;
2
3 import java.util.Date;
4 import java.util.Set;
5
6 import javax.persistence.Entity;
7 import javax.persistence.GeneratedValue;
8 import javax.persistence.GenerationType;
9 import javax.persistence.Id;
10 import javax.persistence.JoinColumn;
11 import javax.persistence.ManyToOne;
12 import javax.persistence.OneToMany;
13
14 @Entity
15 public class Post {
16
17     private Long id;
18     private String content;
19     private Date date;
20     private User user;
21     private Set<Comment> comments;
22
23     @Id
24     @GeneratedValue(strategy = GenerationType.AUTO)
25     public Long getId() {
26         return id;
27     }
28
29     public void setId(Long id) {
30         this.id = id;
31     }
32
33     public String getContent() {
34         return content;
35     }
36
37     public void setContent(String content) {
38         this.content = content;
39     }
40
41     public Date getDate() {
42         return date;
43     }
44
45     public void setDate(Date date) {
46         this.date = date;
47     }
48
49     @ManyToOne
50     @JoinColumn(name = "userId")
51     public User getUser() {
52         return user;
53     }
54
55     public void setUser(User user) {
56         this.user = user;
57     }
58
59     @OneToMany(mappedBy = "post")
60     public Set<Comment> getComments() {
61         return comments;
62     }
63
64     public void setComments(Set<Comment> comments) {
65         this.comments = comments;
66     }
67 }
```

```
1 package com.promineotech.socialMediaApi.repository;
2
3 import org.springframework.data.repository.CrudRepository;
4
5 import com.promineotech.socialMediaApi.entity.User;
6
7 public interface UserRepository extends CrudRepository<User, Long> {
8
9     public User findByUsername(String username);
10
11 }
12
```



PROMINEO TECH

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

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

```
1 package com.promineotech.socialMediaApi.entity;
2
3 import java.util.Set;
4
5 import javax.persistence.CascadeType;
6 import javax.persistence.Entity;
7 import javax.persistence.GeneratedValue;
8 import javax.persistence.GenerationType;
9 import javax.persistence.Id;
10 import javax.persistence.JoinColumn;
11 import javax.persistence.JoinTable;
12 import javax.persistence.ManyToMany;
13 import javax.persistence.OneToOne;
14
15 import com.fasterxml.jackson.annotation.JsonIgnore;
16 import com.fasterxml.jackson.annotation.JsonProperty;
17
18 @Entity
19 public class User {
20
21     private Long id;
22     private String username;
23     private String profilePictureUrl;
24
25     @JsonIgnore
26     private Set<User> following;
27
28     private String password;
29
30     @JsonIgnore
31     private Set<Post> posts;
32
33     @JsonIgnore
34     private Set<Comment> comments;
35
36     @Id
37     @GeneratedValue(strategy = GenerationType.AUTO)
38     public Long getId() {
39         return id;
40     }
41
42     public void setId(Long id) {
43         this.id = id;
44     }
45
46     public String getUsername() {
47         return username;
48     }
49
50     public void setUsername(String username) {
51         this.username = username;
52     }
53
54     @JsonIgnore
55     public String getPassword() {
56         return password;
57     }
58
59     @JsonProperty
60     public void setPassword(String password) {
61         this.password = password;
62     }
63
64     @OneToMany(mappedBy = "user")
65     public Set<Post> getPosts() {
66         return posts;
67     }
68
69     public void setPosts(Set<Post> posts) {
70         this.posts = posts;
71     }
72 }
```



PROMINEO TECH

```
72 @OneToMany(mappedBy = "user")
73 public Set<Comment> getComments() {
74     return comments;
75 }
76
77
78 public void setComments(Set<Comment> comments) {
79     this.comments = comments;
80 }
81
82 @ManyToOne(cascade = CascadeType.ALL)
83 @JoinTable(name = "following", joinColumns = @JoinColumn(name = "userId", referencedColumnName = "id"), inverseJoinColumns = @JoinColumn(name = "followingId", referencedColumnName = "id"))
84 public Set<User> getFollowing() {
85     return following;
86 }
87
88 public void setFollowing(Set<User> following) {
89     this.following = following;
90 }
91
92 public String getProfilePictureUrl() {
93     return profilePictureUrl;
94 }
95
96 public void setProfilePictureUrl(String profilePictureUrl) {
97     this.profilePictureUrl = profilePictureUrl;
98 }
99 }
```

= "id"))

```
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.promineotech.socialMediaApi >-----
[INFO] Building socialMediaApi 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ socialMediaApi ---
[INFO] Deleting /Users/Dtawn/git/SocialMediaAPI/socialMediaApi/target
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:resources (default-resources) @ socialMediaApi ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ socialMediaApi ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 14 source files to /Users/Dtawn/git/SocialMediaAPI/socialMediaApi/target/classes
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:testResources (default-testResources) @ socialMediaApi ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /Users/Dtawn/git/SocialMediaAPI/socialMediaApi/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ socialMediaApi ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /Users/Dtawn/git/SocialMediaAPI/socialMediaApi/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.22.1:test (default-test) @ socialMediaApi ---
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.promineotech.socialMediaApi.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.318 s - in com.promineotech.socialMediaApi.AppTest
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] --- maven-jar-plugin:3.0.2:jar (default-jar) @ socialMediaApi ---
[INFO] Building jar: /Users/Dtawn/git/SocialMediaAPI/socialMediaApi/target/socialMediaApi-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ socialMediaApi ---
[INFO] Installing /Users/Dtawn/git/SocialMediaAPI/socialMediaApi/target/socialMediaApi-0.0.1-SNAPSHOT.jar to /Users/Dtawn/.m2/repository/com/promineotech/socialMediaApi/0.0.1-SNAPSHOT/socialMediaApi-0.0.1-SNAPSHOT.jar
[INFO] Installing /Users/Dtawn/git/SocialMediaAPI/socialMediaApi/pom.xml to /Users/Dtawn/.m2/repository/com/promineotech/socialMediaApi/0.0.1-SNAPSHOT/socialMediaApi-0.0.1-SNAPSHOT.pom
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 18.881 s
[INFO] Finished at: 2020-06-27T10:05:16-07:00
[INFO]
[INFO] -----
```

```
[2020-06-27 10:06:28.230] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- Dialect: HH0000400: Using dialect: org.hibernate.dialect.MySQL5Dialect
[2020-06-27 10:06:29.252] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- ASTQueryTranslatorFactory: HH0000397: Using ASTQueryTranslatorFactory
[2020-06-27 10:06:31.116] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- SchemaExport: HH000227: Running hbm2ddl schema export
Hibernate: alter table comment drop foreign key FK_f1s10xkd2luc3v9v3ktt3tu5
Hibernate: alter table comment drop foreign key FK_mxooj79tmy888avf57mpm82 foreign key (user_id) references user (id)
Hibernate: alter table following drop foreign key FK_q3gdm800v21rktkswb3hw8iv foreign key (following_id) references user (id)
Hibernate: alter table following drop foreign key FK_k7xtuxjpa3o4t0r10bvkn0pv foreign key (user_id) references user (id)
Hibernate: alter table post drop foreign key FK_kpfmivd38f5bw3y1llxeaa foreign key (user_id) references user (id)
Hibernate: drop table if exists comment
Hibernate: drop table if exists following
Hibernate: drop table if exists post
Hibernate: create table comment (id bigint not null auto-increment, content varchar(255), date datetime, post_id bigint, user_id bigint, primary key (id))
Hibernate: create table following (user_id bigint not null, following_id bigint not null, primary key (user_id, following_id))
Hibernate: create table post (id bigint not null auto-increment, content varchar(255), date datetime, user_id bigint, primary key (id))
Hibernate: create table user (id bigint not null auto-increment, password varchar(255), profile_picture_url varchar(255), username varchar(255), primary key (id))
Hibernate: alter table comment add constraint FK_f1s10xkd2luc3v9v3ktt3tu5 foreign key (user_id) references user (id)
Hibernate: alter table comment add constraint FK_mxooj79tmy888avf57mpm82 foreign key (user_id) references user (id)
Hibernate: alter table following add constraint FK_q3gdm800v21rktkswb3hw8iv foreign key (following_id) references user (id)
Hibernate: alter table following add constraint FK_k7xtuxjpa3o4t0r10bvkn0pv foreign key (user_id) references user (id)
Hibernate: alter table post add constraint FK_kpfmivd38f5bw3y1llxeaa foreign key (user_id) references user (id)
[2020-06-27 10:06:33.088] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- SchemaExport: HH000230: Schema export complete
[2020-06-27 10:06:37.177] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- RequestMappingHandlerAdapter: Looking for @ControllerAdvice: org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext@7077772d: startup date [Sat Jun 27 10:06:06 MST 2020]: root of context hierarchy
[2020-06-27 10:06:37.458] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- RequestMappingHandlerMapping: Mapped "{/users/{userId}/posts/{postId},methods=[GET]}" onto public org.springframework.http.ResponseEntity<java.lang.Object> com.promineotech.socialMediaApi.controller.PostController.getPost(java.lang.Long)
[2020-06-27 10:06:37.459] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- RequestMappingHandlerMapping: Mapped "{/users/{userId}/posts/{postId},methods=[PUT]}" onto public org.springframework.http.ResponseEntity<java.lang.Object> com.promineotech.socialMediaApi.controller.PostController.updatePost(com.promineotech.socialMediaApi.entity.Post,java.lang.Long)
[2020-06-27 10:06:37.471] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- RequestMappingHandlerMapping: Mapped "{/users/{userId}/profilePicture,methods=[POST]}" onto public org.springframework.http.ResponseEntity<java.lang.Object> com.promineotech.socialMediaApi.controller.PostController.createPost(com.promineotech.socialMediaApi.entity.Post,java.lang.Long)
[2020-06-27 10:06:37.461] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- RequestMappingHandlerMapping: Mapped "{/users/{userId}/posts,methods=[GET]}" onto public org.springframework.http.ResponseEntity<java.lang.Object> com.promineotech.socialMediaApi.controller.PostController.getAllPosts()
[2020-06-27 10:06:37.462] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- RequestMappingHandlerMapping: Mapped "{/users/{userId}/posts/{postId}/comments,methods=[POST]}" onto public org.springframework.http.ResponseEntity<java.lang.Object> com.promineotech.socialMediaApi.controller.CommentController.createComment(com.promineotech.socialMediaApi.entity.Comment,java.lang.Long,java.lang.Long)
[2020-06-27 10:06:37.462] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- RequestMappingHandlerMapping: Mapped "{/users/{userId}/posts/{postId}/comments/{commentId},methods=[DELETE]}" onto public org.springframework.http.ResponseEntity<java.lang.Object> com.promineotech.socialMediaApi.controller.CommentController.deleteComment(java.lang.Long)
[2020-06-27 10:06:37.464] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- RequestMappingHandlerMapping: Mapped "{/users/{id}/follows/{followId},methods=[POST]}" onto public org.springframework.http.ResponseEntity<java.lang.Object> com.promineotech.socialMediaApi.controller.UserController.follow(java.lang.Long,java.lang.Long)
[2020-06-27 10:06:37.467] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- RequestMappingHandlerMapping: Mapped "{/users/register,methods=[POST]}" onto public org.springframework.http.ResponseEntity<java.lang.Object> com.promineotech.socialMediaApi.controller.UserController.register(com.promineotech.socialMediaApi.entity.User)
[2020-06-27 10:06:37.469] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- RequestMappingHandlerMapping: Mapped "{/users/{id}/follows}" onto public org.springframework.http.ResponseEntity<java.lang.Object> com.promineotech.socialMediaApi.controller.UserController.showFollowedUsers(java.lang.Long)
[2020-06-27 10:06:37.470] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- RequestMappingHandlerMapping: Mapped "{/users/login,methods=[POST]}" onto public org.springframework.http.ResponseEntity<java.lang.Object> com.promineotech.socialMediaApi.controller.UserController.login(com.promineotech.socialMediaApi.entity.User)
[2020-06-27 10:06:37.471] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- RequestMappingHandlerMapping: Mapped "{/users/{id}/profilePicture,methods=[POST]}" onto public org.springframework.http.ResponseEntity<java.lang.Object> com.promineotech.socialMediaApi.controller.UserController.singleFileUpload(java.lang.Long,org.springframework.web.multipart.MultipartFile)
[2020-06-27 10:06:37.476] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- RequestMappingHandlerMapping: Mapped "{/error}" onto public org.springframework.http.ResponseEntity<java.util.Map<java.lang.String,java.lang.Object>> org.springframework.boot.autoconfigure.web.BasicExceptionHandler.error(javax.servlet.http.HttpServletRequest)
[2020-06-27 10:06:37.492] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- RequestMappingHandlerMapping: Mapped "{/error,produces=[text/html]}" onto public org.springframework.web.servlet.ModelAndView org.springframework.boot.autoconfigure.web.BasicExceptionHandler.error(javax.servlet.http.HttpServletRequest)
[2020-06-27 10:06:37.606] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- SimpleUrlHandlerMapping: Mapped URL path [/webjars/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
[2020-06-27 10:06:37.606] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- SimpleUrlHandlerMapping: Mapped URL path [/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
[2020-06-27 10:06:37.901] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- SimpleUrlHandlerMapping: Mapped URL path [/**/favicon.ico] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
[2020-06-27 10:06:37.902] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- AnnotationMethodExporters: Registering beans for JMX exposure on startup
[2020-06-27 10:06:40.245] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- TomcatEmbeddedServletContainer: Tomcat started on port(s): 8080 (http)
[2020-06-27 10:06:40.247] boot - 1485 INFO [com.promineotech.socialMediaApi.App.main()] --- App: Started App in 36.264 seconds (JVM running for 59.689)
```



PROMINEO TECH

Launchpad POST http://localhost:8080/users/register No Environment

Untitled Request

POST http://localhost:8080/users/register Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "username": "Dtown",
3   "password": "352367"
4 }
```

Body Cookies Headers (4) Test Results Status: 201 Created Time: 1433 ms Size: 214 B Save

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "username": "Dtown",
4   "profilePictureUrl": null
5 }
```

```
[Dtown@Dezs-MacBook-Air ~ % mysql -u root -p
```

```
[Enter password:
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 188
```

```
Server version: 8.0.20 MySQL Community Server - GPL
```

```
Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
[mysql> USE social_api;
```

```
Reading table information for completion of table and column names
```

```
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
```

```
[mysql> SELECT * FROM user;
```

id	password	profile_picture_url	username
1	352367	NULL	Dtown

```
1 row in set (0.00 sec)
```

▼ Pictures



ABC771E5-71CC-413F-9F00-CC6CC8CB216B.jpeg



PROMINEO TECH

URL to GitHub Repository: <https://github.com/DesmondYo/SocialMediaAPI>