

✓ 技术准备

- Java 17+
 - Spring Boot 3.x
 - IDE：IntelliJ IDEA
 - 使用 Spring Web 模块即可（不需 JPA、数据库等）
-

项目结构建议

1. 创建 User 实体 (model/User.java)

```
1
2 public class User {
3     private Integer id;
4     private String name;
5     private String gender;
6     private Integer age;
7     private String email;
8     private String phone;
9
10    // 构造方法
11    public User() {}
12
13    public User(Integer id, String name, String gender, Integer age, String email,
14      String phone) {
15        this.id = id;
16        this.name = name;
17        this.gender = gender;
18        this.age = age;
19        this.email = email;
20        this.phone = phone;
21    }
22
23    // Getter & Setter
24    public Integer getId() { return id; }
25    public void setId(Integer id) { this.id = id; }
26
27    public String getName() { return name; }
28    public void setName(String name) { this.name = name; }
29
30    public String getGender() { return gender; }
31    public void setGender(String gender) { this.gender = gender; }
32
33    public Integer getAge() { return age; }
34    public void setAge(Integer age) { this.age = age; }
35
36    public String getEmail() { return email; }
37    public void setEmail(String email) { this.email = email; }
38
39    public String getPhone() { return phone; }
40    public void setPhone(String phone) { this.phone = phone; }
```

40 }

41

2. 创建 Controller (controller/UserController.java)

```
1
2 import io.codescience.model.User;
3 import org.springframework.http.ResponseEntity;
4 import org.springframework.web.bind.annotation.*;
5
6 import java.util.*;
7
8 @RestController
9 @RequestMapping("/api/users")
10 public class UserController {
11
12     // 模拟内存数据存储
13     private final Map<Integer, User> userMap = new HashMap<>();
14     private int currentId = 1;
15
16     // 获取所有用户
17     @GetMapping
18     public ResponseEntity<List<User>> getAllUsers() {
19         return ResponseEntity.ok(new ArrayList<>(userMap.values()));
20     }
21
22     // 创建用户
23     @PostMapping
24     public ResponseEntity<User> createUser(@RequestBody User user) {
25         user.setId(currentId++);
26         userMap.put(user.getId(), user);
27         return ResponseEntity.ok(user);
28     }
29
30     // 获取指定 ID 用户
31     @GetMapping("/{id}")
32     public ResponseEntity<User> getUserById(@PathVariable Integer id) {
33         User user = userMap.get(id);
34         if (user != null) {
35             return ResponseEntity.ok(user);
36         }
37         return ResponseEntity.notFound().build();
38     }
39
```

```
40 // 更新用户
41 @PutMapping("/{id}")
42 public ResponseEntity<?> updateUser(@PathVariable Integer id, @RequestBody User
updatedUser) {
43     User existing = userMap.get(id);
44     if (existing == null) {
45         return ResponseEntity.notFound().build();
46     }
47     updatedUser.setId(id);
48     userMap.put(id, updatedUser);
49     return ResponseEntity.ok().build();
50 }
51
52 // 删除用户
53 @DeleteMapping("/{id}")
54 public ResponseEntity<?> deleteUser(@PathVariable Integer id) {
55     if (userMap.remove(id) != null) {
56         return ResponseEntity.ok().build();
57     }
58     return ResponseEntity.notFound().build();
59 }
60 }
61
```

3. 主程序入口（UserManagementApplication.java）

```
1  
2  
3 import org.springframework.boot.SpringApplication;  
4 import org.springframework.boot.autoconfigure.SpringBootApplication;  
5  
6 @SpringBootApplication  
7 public class UserManagementApplication {  
8     public static void main(String[] args) {  
9         SpringApplication.run(UserManagementApplication.class, args);  
10    }  
11 }  
12
```

4. application.properties (任选其一)

```
1 server:  
2     port: 8080
```

测试建议

下一步：实验二的分层结构合并到此项目

请自行实现该部分