

Санкт-Петербургский национальный исследовательский университет информационных
технологий, механики и оптики

Мегафакультет компьютерных технологий и управления

Факультет программной инженерии и компьютерной техники



Лабораторная работа

Рефакторинг баз данных и приложений

Группа: Р34151

Студент: Дау Конг Туан Ань

Ван Минчжи

Преподаватель: Логинов Иван Павлович

г. Санкт-Петербург, 2024

Оглавление

1. Задачи	3
2. Выполнение.....	3
✓ изменение названий REST ручек для соответствия стандарту	3
✓ Измените аннотацию метода и функцию в соответствии с назначением URL-адреса.....	4
✓ Добавление lombok	5
✓ Добавление логгера.....	7
Код изменений.....	8

1. Задачи

В качестве исходного проекта может быть использован курсовой проект по курсу «Базы данных» или иной сопоставимый по объему программный проект

2. Выполнение

- ✓ изменение названий REST ручек для соответствия стандарту

```
- @Path("/account")
+ @Path("/accounts")
```

```
- @Path("/checkToken")
+ @Path("/check-token")
```

```
- @Path("/getUser")
+ @Path("/user/{id}")
```

```
- @Path("/getRecord")
+ @Path("/records")
```

```
- @Path("/deleteRecord")
+ @Path("/records/{id}")
```

```
- @Path("addRecord")
+ @Path("/records")
```

```
@Path("/getDepartment")
public Response getDepartment(@Context HttpHeaders httpHeaders, coordinate coor) {
    @GET
    @Consumes({MediaType.APPLICATION_JSON, MediaType.TEXT_PLAIN})
    @Produces({MediaType.APPLICATION_JSON, MediaType.TEXT_PLAIN})
    @Path("/departments")
```

Ошибка: Цель запроса должна описываться не через URL, а методом HTTP, запросы во множественном числе возвращают список сущностей, запросы с {id} возвращают сущность с этим ID, Сложные слова следует писать через дефис

- ✓ Измените аннотацию метода и функцию в соответствии с назначением URL-адреса

```
@POST
@DELETE
@Consumes({MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML, MediaType.TEXT_PLAIN})
@Produces({MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML, MediaType.TEXT_PLAIN})
@Path("/deleteRecord")
@Path("/records/{id}")
public Response deleteRecord(@Context HttpHeaders httpHeaders, String time) {
```

```
@POST
@PUT
@Consumes({MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML})
@Produces({MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML})
@Path("/addRecord")
@Path("/records")
public Response addRecord(@Context HttpHeaders httpHeaders, RawRecord raw) {
```

```
@POST
@Consumes({MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML, MediaType.TEXT_PLAIN})
@Produces({MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML, MediaType.TEXT_PLAIN})
@Path("/getDepartment")
public Response getDepartment(@Context HttpHeaders httpHeaders, coordinate coor) {

@GET
@Consumes({MediaType.APPLICATION_JSON, MediaType.TEXT_PLAIN})
@Produces({MediaType.APPLICATION_JSON, MediaType.TEXT_PLAIN})
@Path("/departments")
public Response getDepartment(
```

Ошибка: использование метода http, который не соответствует цели запроса, например использование метода POST для отправки запроса на удаление объекта.

✓ Добавление lombok

Цель: использовать встроенные методы Lombok для сокращения кода, предотвращения дублирования кода и отсутствия кода.

```
<dependencies>
+   <dependency>
+       <groupId>org.projectlombok</groupId>
+       <artifactId>lombok</artifactId>
+       <version>1.18.30</version>
+       <scope>provided</scope>
+   </dependency>
</dependencies>
```

```
+ @Getter
+ @Setter
+ @AllArgsConstructor
public class ListId {
    List<Long> listId;

-   public ListId(List<Long> listId) {
-       this.listId = listId;
-   }
-
-   public List<Long> getListId() {
-       return listId;
-   }
-
-   public void setListId(List<Long> listId) {
-       this.listId = listId;
-   }
- }
```

```

+ @Getter
+ @Setter
+ @AllArgsConstructor
+ @NoArgsConstructor

public class RawComment {

    private String name;

    @@ -27,54 +35,4 @@ public RawComment(String name,
        this.rate = rate;
        this.time = time;
    }

-
- public RawComment(){};
-
- public String getAvatar() {
-     return avatar;
- }
-
-

```

```

+ @Getter
+ @AllArgsConstructor

public class RawAccount {

    private final String username;
    private final String password;
    private final Long id;

- public RawAccount(String username,
-                     String password,
-                     Long id) {
-     this.username = username;
-     this.password = password;
-     this.id = id;
- }
-
- public String getUsername() {
-     return username;
- }
-
- public String getPassword() {
-     return password;
- }
-
- public Long getId() {
-     return id;
- }

```

```

+ @Getter
+ @AllArgsConstructor

public class RawChat {
    private String from;
    private String to;
    List<RawMessage> messages;

-
-     public RawChat(String from,
-                     String to,
-                     List<RawMessage> messages) {
-         this.from = from;
-         this.to = to;
-         this.messages = messages;
-     }
-
-     public String getFrom() {
-         return from;
-     }
-
-     public String getTo() {
-         return to;
-     }
-
-     public List<RawMessage> getMessages() {
-         return messages;
-     }
- }

```

✓ Добавление логгера

Цель: вместо вывода информации на консоль, что затрудняет получение и мониторинг информации о сервере, используйте регистраторы для контроля состояния системы.

```

+
+     <dependency>
+         <groupId>org.slf4j</groupId>
+         <artifactId>slf4j-api</artifactId>
+         <version>1.7.36</version> <!-- 最新版 -->
+     </dependency>
+     <dependency>
+         <groupId>org.slf4j</groupId>
+         <artifactId>slf4j-simple</artifactId> <!-- 或 logback-classic -->
+         <version>1.7.36</version>
+     </dependency>
+
+
+

```

```
-      System.out.println(coor.toString());  
-      System.out.println(comment.toString());  
-  
+      // System.out.println(coor.toString());  
+      // System.out.println(comment.toString());  
+      log.info("Coordinate received: " + coor.toString());  
+      log.info("Comment received: " + comment.toString());  
+  
+
```

```
-      System.out.println(comment.toString());  
+  
+      // System.out.println(comment.toString());  
+      log.info("Comment received: " + comment.toString());
```

```
-      System.out.println(list.toString());  
+      // System.out.println(list.toString());  
+      log.info("Location received: " + list.toString());
```

```
-      System.out.println(e.getMessage());  
+      // System.out.println(e.getMessage());  
+      log.error("JWT Verification failed: {}", e.getMessage(), e);
```

Код изменений

Всю проделанную работу(все изменения в коде) с кратким описанием изменений можно найти в [PR](#) на гитхабе проекта: [refactoring app and db](#)