

Final Project Development Journal

(This document is a journal of the development process that I made day by day.)

First Week

04.07.2024

LogMessage class implemented for logging and debugging purposes. This class is coloring the errors and exceptions as **red**, and success messages or info as **green**.

```
// This class below is coloring the log messages for better debugging
public class LogMessage { 3 usages  ⚡ erkam.karaca *
    public static final String generate(MessageStatus status, String message) { no usages  ⚡ erkam.karaca
        String color = status.equals(MessageStatus.NEG) ? StringColor.ANSI_RED : StringColor.ANSI_GREEN;
        return color + message + StringColor.ANSI_RESET;
    }
    // This function is for String parameters
    public static final String generate(MessageStatus status, String message, String name) { 2 usages  ⚡ erkam.karaca
        String color = status.equals(MessageStatus.NEG) ? StringColor.ANSI_RED : StringColor.ANSI_GREEN;
        return color + message + StringColor.ANSI_RESET + " = " + name;
    }
    // This function is for Long parameters
    public static final String generate(MessageStatus status, String message, Long id) { no usages  ⚡ erkam.karaca
        String color = status.equals(MessageStatus.NEG) ? StringColor.ANSI_RED : StringColor.ANSI_GREEN;
        return color + message + StringColor.ANSI_RESET + " = " + id;
    }
}
```

```
.UserService : User created = aliveli@gmail.com

.UserService : User already exists = aliveli@gmail.com
0

.UserService : All users fetched
0 where u1_0.id=?

.UserService : User not found with id = 99
0 where u1_0.id=?

.UserService : User fetched = a@a12.com
0 where u1_0.id=?

.UserService : User fetched = aliveli@gmail.com
0 where u1_0.id=?

tus,l1_1.title,l1_1.type from users_listings l1_0 join l
quantity,p1_1.title from users_packages p1_0 join packages

.UserService : User deleted = a@a12.com
```

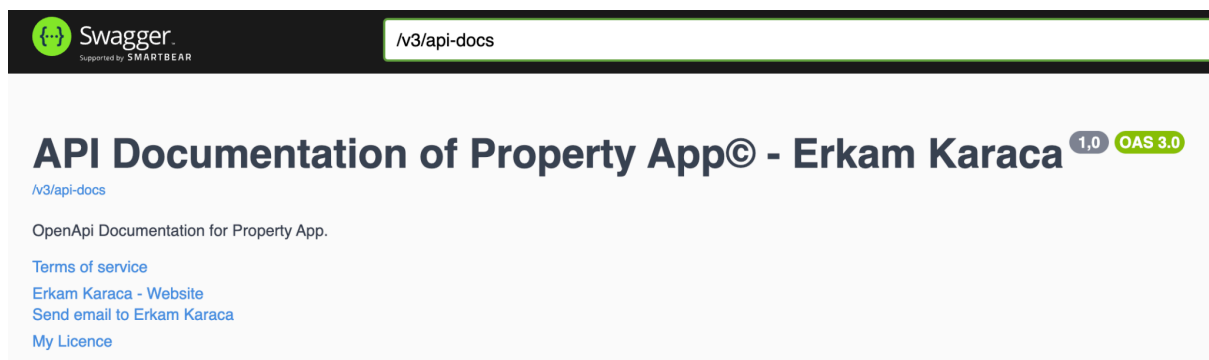
Global Exception Handler and Error Details classes implemented.

```
@ControllerAdvice  @erkam.karaca
public class GlobalExceptionHandler {

    @ExceptionHandler(UserException.UserAlreadyExistException.class)  @erkam.karaca
    public ResponseEntity<> handleUserAlreadyExistsException(UserException.UserAlreadyExistException exception, WebRequest request) {
        ErrorDetails errorDetails = new ErrorDetails(HttpStatus.BAD_REQUEST.value(), exception.getMessage(), request.getDescription(includeClientInfo: false));
        return new ResponseEntity<>(errorDetails, HttpStatus.BAD_REQUEST);
    }

    @ExceptionHandler(Exception.class)  @erkam.karaca
    public ResponseEntity<> handleGlobalException(Exception ex, WebRequest request) {
        ErrorDetails errorDetails = new ErrorDetails(HttpStatus.INTERNAL_SERVER_ERROR.value(), ex.getMessage(), request.getDescription(includeClientInfo: false));
        return new ResponseEntity<>(errorDetails, HttpStatus.INTERNAL_SERVER_ERROR);
    }
}
```

SwaggerUI implemented.



Unit tests of User Service and User Controller added at the beginnings of the project.

| Coverage | | UserControllerTest | | | |
|--------------------------------|--|--------------------|------------|------------|------------|
| Element | | Class, % | Method, % | Line, % | Branch, % |
| org.erkam.propertyapp.controll | | 100% (1/1) | 100% (5/5) | 100% (5/5) | 100% (0/0) |
| UserController | | 100% (1/1) | 100% (5/5) | 100% (5/5) | 100% (0/0) |

| Coverage | | UserServiceTest | | | |
|-------------------------------|--|-----------------|------------|--------------|------------|
| Element | | Class, % | Method, % | Line, % | Branch, % |
| org.erkam.propertyapp.service | | 100% (1/1) | 100% (8/8) | 100% (25/25) | 100% (4/4) |
| UserService | | 100% (1/1) | 100% (8/8) | 100% (25/25) | 100% (4/4) |