

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
```

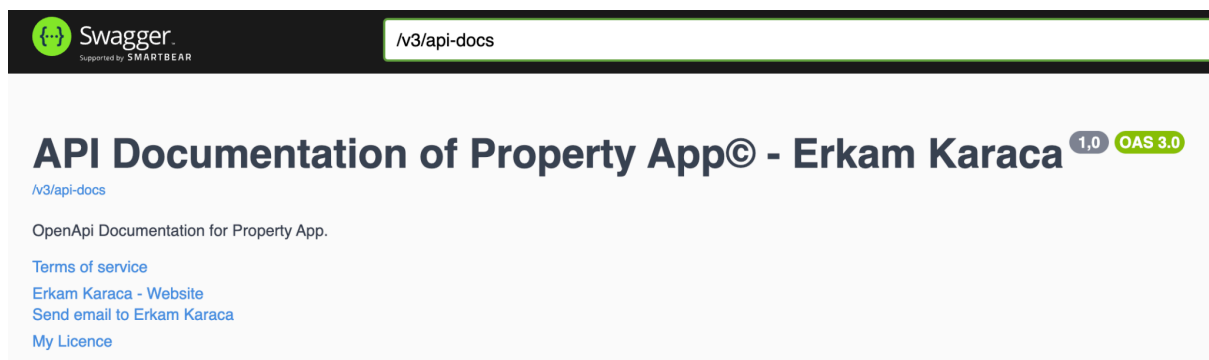
Global Exception Handler and Error Details classes implemented.

```
@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)

07.07.2024

Listing controller, service and repository added.

```
@Service 2 usages new *
@Slf4j
@RequiredArgsConstructor
public class ListingService {
    private final ListingRepository listingRepository;

    // TODO: Implement a check mechanism to know is there any duplicate
    // of this listing in database
    public GenericResponse<ListingSaveResponse> save(ListingSaveRequest request) { 1 usage new *
        // TODO: Check here and throw an exception if any duplicate exists
        listingRepository.save(ListingConverter.toListing(request));
        log.info(LogMessage.generate(MessageStatus.POS, ListingSuccessMessage.LISTING_CREATED, request.getTitle()));
        return GenericResponse.success(ListingSaveResponse.of(request));
    }

    // Get all listings from database if there is no data on database then throw an exception,
    // else convert listings to ListingGetResponse list then return it.
    public GenericResponse<List<ListingGetResponse>> getAll() { 1 usage new *
        List<Listing> listings = listingRepository.findAll();
        if (listings.isEmpty()) {
            log.error(LogMessage.generate(MessageStatus.NEG, ListingExceptionMessage.NO_DATA_ON_DATABASE));
            throw new ListingException.NoDataOnDatabaseException(ListingExceptionMessage.NO_DATA_ON_DATABASE);
        }
        log.info(LogMessage.generate(MessageStatus.POS, ListingSuccessMessage.ALL_LISTINGS_FETCHED));
        return GenericResponse.success(ListingConverter.toListingGetResponseList(listings));
    }
}
```

Listing controller and service unit tests added.

Coverage ListingControllerTest x				
Element ^				
Class, %				
Method, %				
Line, %				
Branch, %				
org.erkam.propertyapp.controll	33% (1/3)	50% (5/10)	50% (5/10)	100% (0/0)
AuthController	0% (0/1)	100% (0/0)	100% (0/0)	100% (0/0)
ListingController	100% (1/1)	100% (5/5)	100% (5/5)	100% (0/0)

Coverage ListingServiceTest x				
Element ^				
Class, %				
Method, %				
Line, %				
Branch, %				
org.erkam.propertyapp.service	33% (1/3)	52% (9/17)	57% (34/59)	50% (4/8)
AuthService	0% (0/1)	100% (0/0)	100% (0/0)	100% (0/0)
ListingService	100% (1/1)	100% (9/9)	100% (34/34)	100% (4/4)

```
✓ Tests passed: 8 of 8 tests – 623 ms

/Library/Java/JavaVirtualMachines/temurin-17.jdk/Contents/Home/bin/java ...
16:21:11.827 [main] INFO org.erkam.propertyapp.service.ListingService -- Listing deleted with id = 6532
16:21:11.840 [main] ERROR org.erkam.propertyapp.service.ListingService -- Listing not found with id = 1
16:21:11.845 [main] ERROR org.erkam.propertyapp.service.ListingService -- Duplicate listing found = VRKIDZEQUO
16:21:11.848 [main] ERROR org.erkam.propertyapp.service.ListingService -- Listing not found with id = 1
16:21:11.850 [main] ERROR org.erkam.propertyapp.service.ListingService -- No data found on database
16:21:11.852 [main] INFO org.erkam.propertyapp.service.ListingService -- All listings fetched
16:21:11.856 [main] INFO org.erkam.propertyapp.service.ListingService -- Listing created = DFPCU0G0
16:21:11.860 [main] INFO org.erkam.propertyapp.service.ListingService -- Listing fetched with id = 508

Process finished with exit code 0
```

Gateway and Registry added.

Services	
	Console
Spring Boot	
PropertyServiceApplication :8081/	2024-07-08T01:46:04.564+03:00 INFO 30712 --- [property-service] [
PropertyGatewayApplication :8080/	2024-07-08T01:46:04.578+03:00 INFO 30712 --- [property-service] [
PropertyRegistryApplication :8761/	2024-07-08T01:46:04.580+03:00 INFO 30712 --- [property-service] [
	2024-07-08T01:46:04.581+03:00 INFO 30712 --- [property-service] [
	2024-07-08T01:46:04.582+03:00 INFO 30712 --- [property-service] [nfoRepl
	2024-07-08T01:46:04.593+03:00 INFO 30712 --- [property-service] [
	2024-07-08T01:46:04.593+03:00 INFO 30712 --- [property-service] [