



Lesson



← → 🔍 Baku Engineering University(18.01.2024)

# Object Oriented Programming 2

**Nabi Shirinov**

**Lecturer : Yusif Yusifov**



Lesson

App Overview



← → 🔍 Group Members

# Rental App Overview



The rental car app lets users share their cars for rent, allowing others to easily find and learn about available rentals. Users can also manage their listings by editing or deleting information as necessary.



Lesson

App Overview

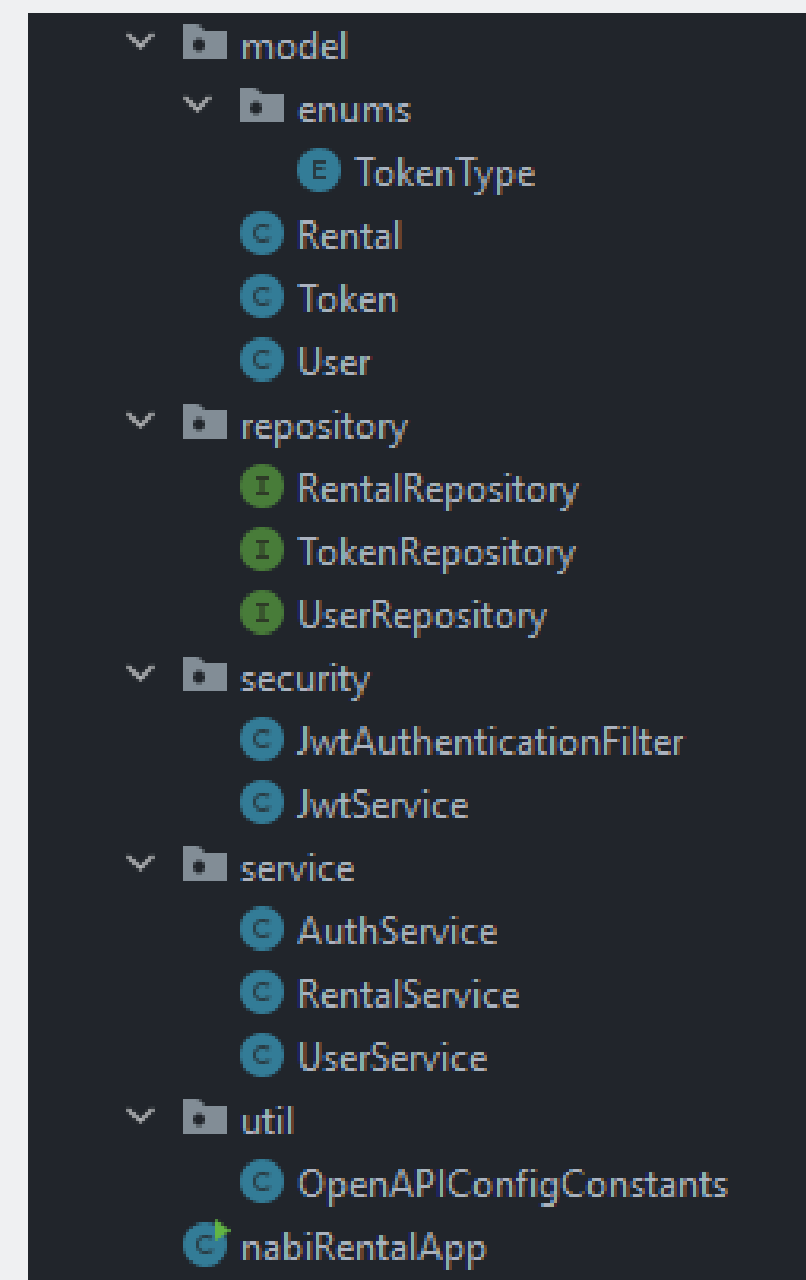
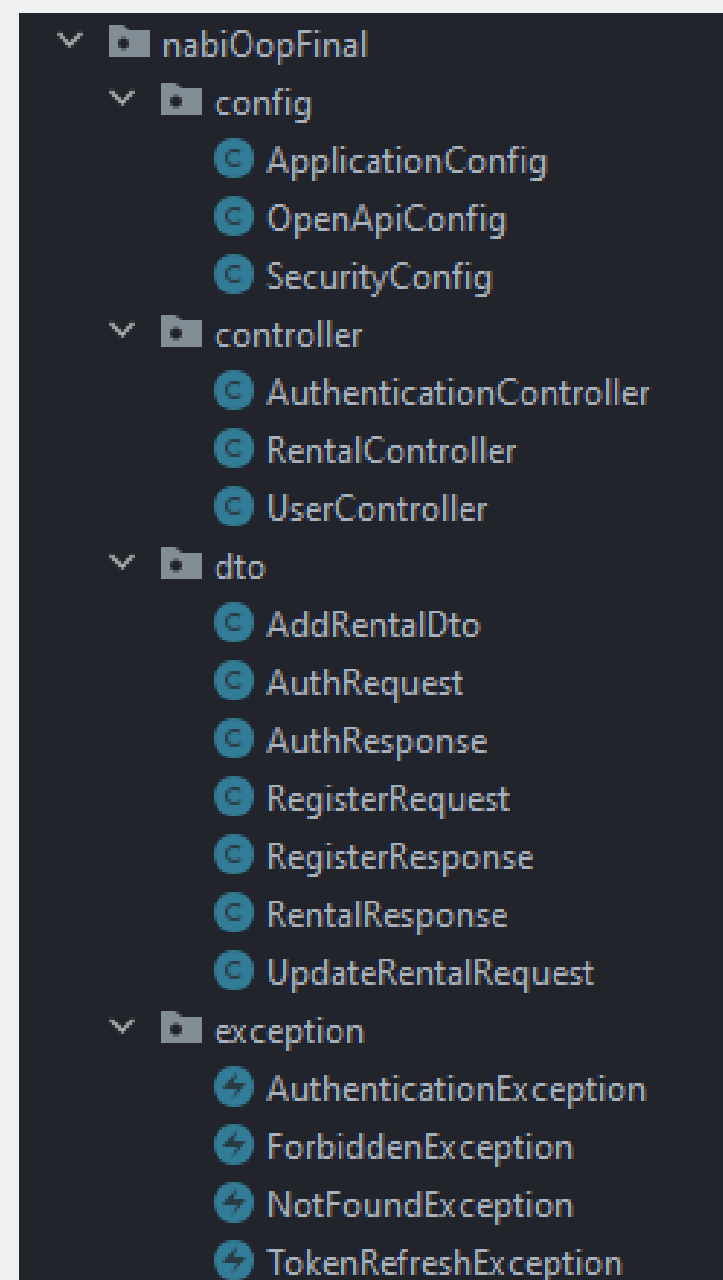
Clean Structure



← → 🔍 Insert your topic here

# Clean Structure

My code features a clean structure with well-named classes and modular design, enhancing readability and maintainability.





Lesson

App Overview

Clean Structure

Database Connection



← → 🔍 Insert your topic here

# Database Connection

I have three tables in my postgres database, with a one-to-many relationship existing between "\_user" and "token," as well as between "\_user" and "rental."

	id [PK] integer	email character varying (255)	password character varying (255)	username character varying (255)
1	1	elariz@gmail.com	\$2a\$10\$OBZGgZiYioRUbl7OYpcrWuc5iucUoBQQcj2dxqtvDW4dp25BRwc...	elariz
2	2	nabi1@gmail.com	\$2a\$10\$ZzU3ESQPcViHnztLQWWmo.GMmvXob6mzi1ssS000u4sJt8lhfu...	Nabi1
3	3	nabi1@gmail.com	\$2a\$10\$ulrLcjETHuvibkQTEFVResETz/M10efyrwyhR3qX5TbUt4645vzy	Nabi1
4	4	nabi3@gmail.com	\$2a\$10\$BXLl.I5oZn7y.vCoBQuiZe/ei1eN7NsPGSV164vt5ZNVhkUbv00fa	Nabi3
5	5	nabi3@gmail.com	\$2a\$10\$Dy8eyxloXEAXe5hZc8zOROCORG7gKUIEmMnCdTYUNaVfgKcZg...	[null]
6	6	nabi3@gmail.com	\$2a\$10\$VAfGWbg/6cwlwCMzICxTseW9LqVx5eqh45RI8xaMbcppPvhN5c...	Nabi Shirinov3
7	7	murad@gmail.com	\$2a\$10\$LTBXi4CE5aFqTRShYc0I0.CQJI8v5GiasPDYCFbmSBCjszejwJsC	[null]
8	8	ayhan@gmail.com	\$2a\$10\$qhv29.EubSyzo29ocwEFDeXL/63L8VzOke11kBf1af11cZR2Gr3nS	Ayhan Alakbarov
9	10	cafar@gmail.com	\$2a\$10\$Sfu0PluRBX/2qAGyq3BDSueEwWSu32UsfcbpTLVwH/6DCatkb7...	Cafar Bakish
10	11	abid@gmail.com	\$2a\$10\$zQpEOdy9C9/Kuy1LY2Y5deG6kwD23KyCqy76DeTQKx8bB6s.Tf5...	Abid Qurbanov

Tables (3)
> _users
> rental
> token

	id [PK] integer	car_model character varying (255)	car_name character varying (255)	car_year character varying (255)	driven_km character varying (255)	price integer	user_id integer
1	1	x3	bmw	2000	123112	15000	4
2	3	x3	bmw	2000	1213	10	4
3	6	X7	BMW	2020	8000	6000	1
4	7	camry	toyota	2020	1000	2500	1



Lesson

App Overview

Clean Structure

Database Connection

CRUD operations



← → 🔍 Insert your topic here

# CRUD operations(Rentals)

```
@PostMapping("/add")
public ResponseEntity<?> addRental(@RequestBody() AddRentalDto dto) {
    rentalService.addRental(dto);
    return ResponseEntity.ok().build();
}

no usages  🧑 NabiShirinov
@GetMapping("/get/{id}")
public ResponseEntity<RentalResponse> getRentalsById(@PathVariable("id") Integer id) {
    RentalResponse rentalResponse = rentalService.getRentalsById(id);
    return new ResponseEntity<>(rentalResponse, HttpStatus.OK);
}

no usages  🧑 NabiShirinov
@DeleteMapping("/delete/{id}")
public ResponseEntity<?> deleteRentalById(@PathVariable("id") Integer id) {
    try {
        rentalService.deleteRentalById(id);
        return new ResponseEntity<>({ body: "Rental deleted successfully", HttpStatus.OK);
    } catch (NotFoundException e) {
        return new ResponseEntity<>(e.getMessage(), HttpStatus.NOT_FOUND);
    } catch (Error e) {
        return new ResponseEntity<>(e.getMessage(), HttpStatus.FORBIDDEN);
    }
}
```

```
@PutMapping("update/{id}")
public ResponseEntity<Void> updateRental(@PathVariable("id") Integer id, @RequestBody UpdateRentalRequest newRental) {
    rentalService.updateRental(id, newRental);
    return ResponseEntity.ok().build();
}

no usages  🧑 NabiShirinov
@PatchMapping("partialUpdate/{id}")
public ResponseEntity<Void> updateRentalPartially(@PathVariable("id") Integer id, @RequestBody UpdateRentalRequest newRental) {
    rentalService.updateRentalPartially(id, newRental);
    return ResponseEntity.ok().build();
}

no usages  🧑 NabiShirinov
@GetMapping("/myRentals")
public ResponseEntity<List<RentalResponse>> getMyRentals() {
    List<RentalResponse> myRentals = rentalService.getMyRentals();
    return new ResponseEntity<>(myRentals, HttpStatus.OK);
}

no usages  🧑 NabiShirinov
@GetMapping("/all")
public ResponseEntity<List<RentalResponse>> getAllRentals() {
    List<RentalResponse> allRentals = rentalService.getAllRentals();
    return new ResponseEntity<>(allRentals, HttpStatus.OK);
}
```



Lesson

App Overview

Clean Structure

Database Connection

CRUD

Swagger



← → 🔍 Insert your topic here

# Swagger

In this Swagger documentation, you can access and utilize my APIs by exploring the listed endpoints and their corresponding functionalities.

Rental Application 1.0 OAS3

[/v3/api-docs](#)

Rental Application

Servers

http://localhost:8090 - Generated server url

Authorize

rental-controller

PUT

/api/v1/rental/update/{id}

POST

/api/v1/rental/add

PATCH

/api/v1/rental/partialUpdate/{id}

GET

/api/v1/rental/myRentals

GET

/api/v1/rental/get/{id}

GET

/api/v1/rental/all

DELETE

/api/v1/rental/delete/{id}

authentication-controller

POST

/api/v1/auth/register

POST

/api/v1/auth/login

user-controller

DELETE

/api/v1/user/delete

Active  
Go to S



[Lesson](#)[App Overview](#)[Clean Structure](#)[Database Connection](#)[CRUD](#)[Logging](#)

← → 🔍 Insert your topic here

# Logging

```
logger.info("Rental {} Partially Updated.", rental.getId());
```

```
logger.info("Rental {} {} Updated.", newRental.getCarName(), newRental.getCarModel());
```

All processes are logged to the "RentalApp.log" file, providing a record of activity within the Rental App.

```
logger.info("Rental Deleted: {} {}", rental.getCarName(), rental.getCarModel());
```

```
logger.info("Registration successful for username: {}", request.getUsername());
```

```
02:06:15.673 [http-nio-8090-exec-3] INFO  nabi0opFinal.service.RentalService - New Rental Added: BMW X7
```

```
19:28:35.834 [http-nio-8090-exec-7] INFO  nabi0opFinal.service.RentalService - Rental toyota camry Updated.
```

```
19:29:02.695 [http-nio-8090-exec-9] INFO  nabi0opFinal.service.RentalService - Rental 7 Partially Updated.
```



# Error Handling

Effectively managing errors and providing custom exceptions enhances the robustness of the application, ensuring a more graceful handling of unexpected situations and improving the user experience.

```
public void deleteRentalById(Integer rentalId) {
    UserDetails userDetails = (UserDetails) SecurityContextHolder.getContext().getAuthentication().getPrincipal();
    Optional<User> userOptional = userRepository.findByEmail(userDetails.getUsername());

    if (userOptional.isPresent()) {
        User user = userOptional.get();

        // Check if the rental belongs to the logged-in user
        Optional<Rental> rentalOptional = rentalRepository.findById(rentalId);

        if (rentalOptional.isPresent()) {
            Rental rental = rentalOptional.get();

            if (rental.getUser().getId().equals(user.getId())) {
                // Delete the rental if it belongs to the logged-in user
                rentalRepository.deleteById(rentalId);
                logger.info("Rental Deleted: {} {}", rental.getCarName(), rental.getCarModel());
            } else {
                throw new Error("You do not have permission to delete this rental.");
            }
        } else {
            throw new NotFoundException("Rental not found.");
        }
    } else {
        throw new NotFoundException("User not found.");
    }
}
```



[Lesson](#)[App Overview](#)[Clean Structure](#)[Database Connection](#)[CRUD](#)[Methods usage](#)

← → 🔍 Insert your topic here

# Methods usage

```
public void updateRental(Integer id, UpdateRentalRequest newRental) {
    UserDetails userDetails = (UserDetails) SecurityContextHolder.getContext().getAuthentication().getPrincipal();
    Optional<User> userOptional = userRepository.findByEmail(userDetails.getUsername());

    if (userOptional.isPresent()) {
        User user = userOptional.get();

        // Check if the rental belongs to the logged-in user
        Optional<Rental> rentalOptional = rentalRepository.findById(id);

        if (rentalOptional.isPresent()) {
            Rental rental = rentalOptional.get();

            if (rental.getUser().getId().equals(user.getId())) {

                var requestedRental = rentalRepository.findById(id).orElseThrow(() -> new RuntimeException("Rental not found"));
                requestedRental.setCarName(newRental.getCarName());
                requestedRental.setCarModel(newRental.getCarModel());
                requestedRental.setCarYear(newRental.getCarYear());
                requestedRental.setDrivenKm(newRental.getDrivenKm());
                requestedRental.setPrice(newRental.getPrice());
                logger.info("Rental {} {} Updated.", newRental.getCarName(), newRental.getCarModel());
                rentalRepository.save(requestedRental);

            } else {
                throw new Error("You do not have permission to delete this rental.");
            }
        } else {
            throw new NotFoundException("Rental not found.");
        }
    } else {
        throw new NotFoundException("User not found.");
    }
}
```

```
public void addRental(AddRentalDto dto) {
    UserDetails userDetails = (UserDetails) SecurityContextHolder.getContext().getAuthentication().getPrincipal();
    System.out.println(userDetails.getUsername());
    Optional<User> user = userRepository.findByEmail(userDetails.getUsername());

    if (user.isPresent()) {
        Rental rental = new Rental();
        rental.setCarName(dto.getCarName());
        rental.setCarModel(dto.getCarModel());
        rental.setCarYear(dto.getCarYear());
        rental.setDrivenKm(dto.getDrivenKm());
        rental.setPrice(dto.getPrice());
        rental.setUser(user.get());

        rentalRepository.save(rental);
        logger.info("New Rental Added: {} {}", rental.getCarName(), rental.getCarModel());
    }
}
```



Lesson

App Overview

Clean Structure

Database Connection

CRUD

Git Commits



← → 🔍 Insert your topic here

# Git Commits (15 (from 2 Jan to 18 Jan)

Commits on Jan 17, 2024		
Delete User	NabiShirinov committed 6 hours ago	59656d9
Delete User	NabiShirinov committed 6 hours ago	
Exceptions	NabiShirinov committed 7 hours ago	
Update	NabiShirinov committed yesterday	
Update	NabiShirinov committed yesterday	
CRUD update	NabiShirinov committed yesterday	
Transfer	NabiShirinov committed yesterday	
RentalAppFinal	NabiShirinov committed yesterday	
Commits on Jan 16, 2024		
one to many relation	NabiShirinov committed 2 days ago	4f337d9
Commits on Jan 2, 2024		
update	NabiShirinov committed 2 weeks ago	4c3a5cf

Commits on Jan 18, 2024		
User Structure Response + Register Multiple Mail Issue solved	NabiShirinov committed 2 minutes ago	9d4e860
Structured Response For Rentals	NabiShirinov committed 24 minutes ago	e63b156
ApiResponse	NabiShirinov committed 1 hour ago	c0f7818
Structured response	NabiShirinov committed 1 hour ago	c9c3d9d
Structured Response	NabiShirinov committed 1 hour ago	3a9f8f1



Lesson

App Overview

Clean Structure

Database Connection

CRUD

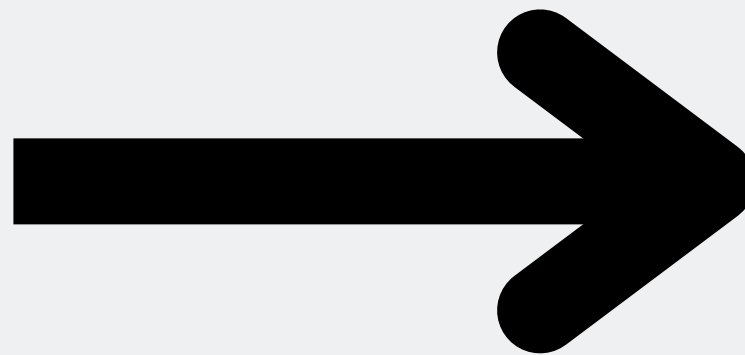
Structured Response ✕



← → 🔍 Insert your topic here

# Structured Responses

```
{
  "message": "string",
  "code": "string",
  "data": {
    "id": 0,
    "carName": "string",
    "carYear": "string",
    "carModel": "string",
    "drivenKm": "string",
    "price": 0
  }
}
```



```
{
  "message": "My rentals retrieved successfully",
  "code": "RETRIEVED",
  "data": [
    {
      "id": 15,
      "carName": "string",
      "carYear": "string",
      "carModel": "string",
      "drivenKm": "string",
      "price": 0
    },
    {
      "id": 9,
      "carName": "Mercedes",
      "carYear": "2022",
      "carModel": "GLA 330",
      "drivenKm": "10000",
      "price": 1400
    }
  ]
}
```

**Thank you!**