# App for fictious Car Showroom

## Base params of application

* Spring Boot 2.7.6
* Java 11
* DB: Oracle 19
* Template engine: Thymeleaf

## Description

This document describes my training application in Spring Boot on which I have been working for last days. At this time the applications is consist of about 30 entities and similar number of tables in db. For now is in progress stuff around car configuration. Entities - CarBrand, CarModel, CarEquipmentPack, CarEngine, etc. And the other branch where I am working on is around human resources - entity Employee, and many relational entities. I was created 3 ER diagrams. Which could be usefull for understanding the logic.

* ER-human-resources
* ER-car-variant
* ER-rest-of

## Documentation

ER diagrams: https://github.com/mirosimo/car\_showroom/tree/main/xxx-documentation/ER-diagrams

Some View screenshots Examples: https://github.com/mirosimo/car\_showroom/tree/main/xxx-documentation/app-screenshots

## Implemented Features

For now there are implemented things like: Inserting, Updating, Deleting records, Relation between entities, One To Many, Many To Many, Many To Many with extra columns in connection table, Exception management, Multilanguage, Views Validation, Just now in progress - JWT - user authorization will be followed by implementing User Roles, Privileges management. Images are saved as blob in db, Thymeleaf, Css.

## Multilanguage

* messages.properties
* messages\_en.properties

## Error, Exception management

When occur an exception, than is user informed ( In concise form via Error page ).

When will be done Priviledge, Role management, than the users with needed priviledges could also display print stack Trace of an exception.

## Thymeleaf

Template files are places in folder: **Resources/templates**

Templates are divided into 4 base groups. Postfix

* -new - view for adding new entity into system
* -list - view where are displayed data in table form
* -detail - view where is displayed detail for particular entity
* -update - view for updating entity

Attribute menuItem (used for menu generating) is pass into templates through MenuInterceptor - Not from controllers. Defining in each controller had been little bit annoying.

## Fragments

Fragment files are placed in folder: **Resources/templates/fragments**

Just now are using 3 fragments.

* **top\_part** - the very top part of application - contains logo and two design images
* **navbar** - contains manu
* **input\_warning\_board** - contains alert board, which is displayed, when user inserts and confirm wrong data.

## CSS

* Top Strip – logo, two images – used FLEX.
* Views for adding new records (label, input box) - used Grid.
* Views for table form data output – used Table
* Top Menu – resources/static/css/top\_navbar.css

Main CSS file – resources/static/css/app\_style.css

## TODO

In the near future will be done :

* User management - Privileges, Roles
* Authentication – JWT
* Error logging