**ДНІПРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ   
ІМЕНІ ОЛЕСЯ ГОНЧАРА**

Факультет прикладної математики

Кафедра обчислювальної математики та математичної кібернетики

# ЗВІТ

до виконання контрольно-модульної роботи

з дисципліни «Сучасні середовища програмування»

Виконав:

Студент групи ПА-17-1

М.О. Кригін

2020

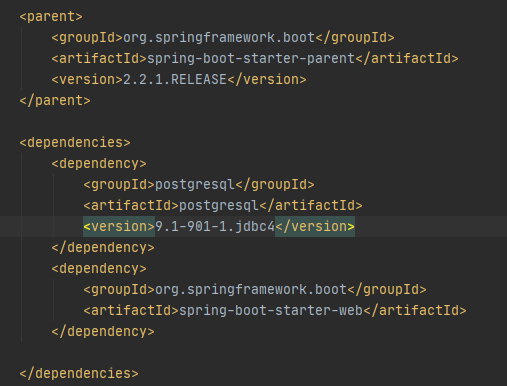
# Мета роботи

Мета роботи: Набути навички використання технологій Spring та Hibernate для розробки Rest Api інтерфейсу на прикладі онлайн магазину мототехніки.

# Постановка завдання

1. Створити проект з необхідними бібліотеками для Spring та Hibernate
2. Підключити та наповнити базу данних з необхідними сутностями.
3. Створити Rest Api, який повинен давати змогу виставляти на продаж транспортний засіб та переглядати створенні пропозиції
4. Написати графічну оболочку для тестування Api інтерфейсу

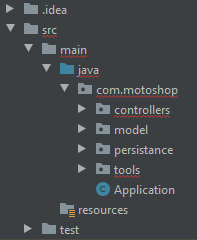
# Хід роботи

1. Створимо новий проект з наступними бібліотеками

Spring-boot-starter-web – бібліотека яка включає найнеобхідніші засоби для побудови web застосунків. В нашому випадку це spring-container, spring-web та hibernate

Postgresql – бібліотека необхідна hibernate для використання postgress бази даних

1. Розіб’ємо проект на частини як показано нижче



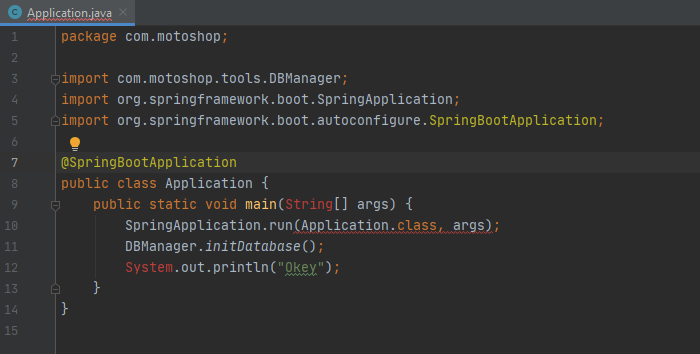
Controllers – відповідають за rest api points, які грфічна оболонка буде використовувати для отримання даних

Model – сукупність POJO (Plan Old Java Object) класів, які містять у собі дані

Persistance – класи необхідні для роботи с базою даних

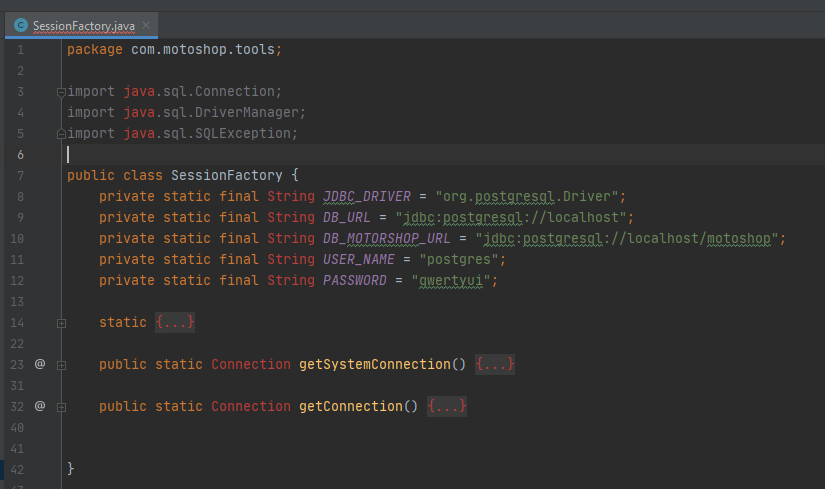
Tools - інші класи, які мають вузьке призначення

1. Створимо вхідну точку для програм java

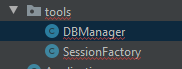


Анотація @SpringBootApplication необхідна умова для роботи бібліотеки spring-boot

1. Створимо класс, який буде підключатися до бази даних:

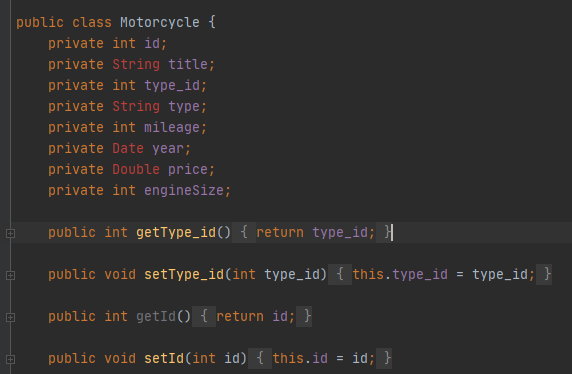


1. Створимо класс, який буде заповнювати даними нову базу – DBManager. Фінальна структура пакету tool буде наступна:



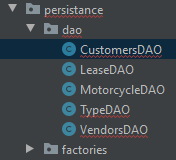
1. Наступним кроком розробимо класи даних у пакеті model

* Customer – Користувач програми
* Lease – договір продажі \ оренди
* Motorcycle – Клас мотоциклу
* Vendor – Компанія, яка створила транспортний засіб

Приклад классу Motorcycle Який складається лише з приватних полів та методів, які задають\повертають значення одного поля.

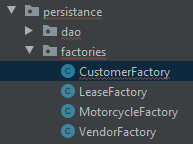
Підпакет dto э частинкою шаблону проектування dto (див <https://martinfowler.com/eaaCatalog/dataTransferObject.html>)

1. Почнемо працювати з базою, створивши декілька класів за шаблоном проектування DAO (див <https://en.wikipedia.org/wiki/Data_access_object>)



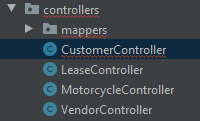
Де кожний клас відповідальний за отримання\збереження класа своєї сутності. Наприклад: класс MotorcycleDAO має методи для отримання\ збереження нового класу Motorcycle до бази данних.

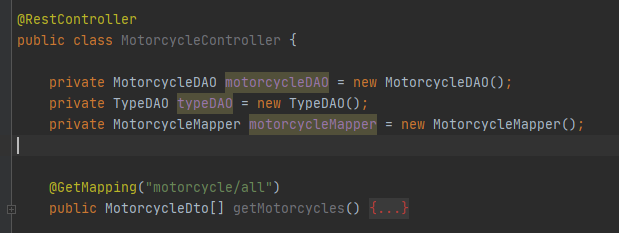
1. Поряд, у пакеті persistence, створимо підпакет factories



Який буде відповідати за перетворення результатів отриманих з бази даних в класи java.

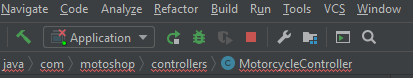
1. Наступним кроком буде налаштування можливості зв'язку програми з зовнішнім світом, а саме створення контролерів



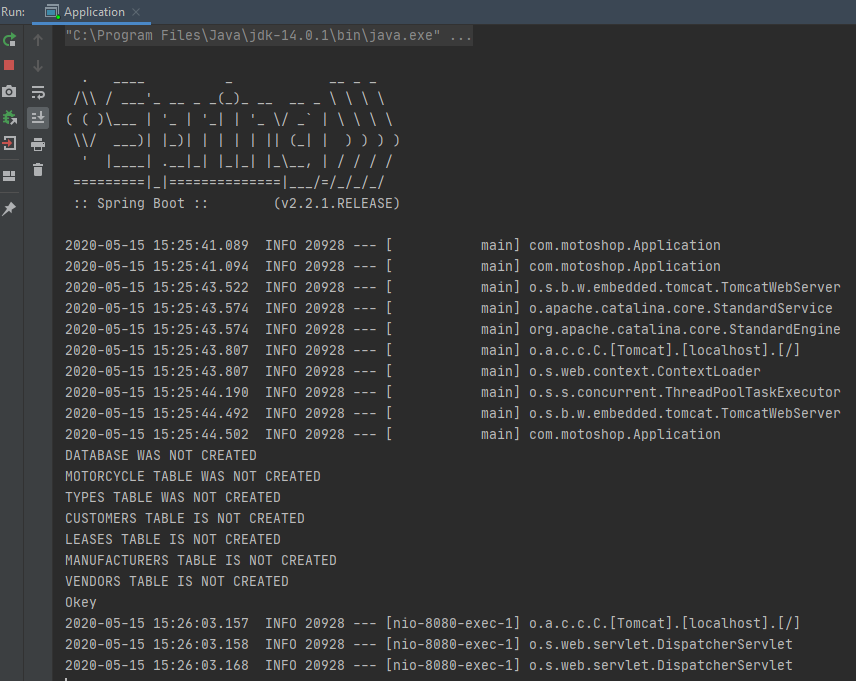
Де кожен контролер має деякий url шлях, який пов'язаний з методом java. Наприклад MotorcycleController 

Анотація @GetMapping(“motorcycle/all”) говорить, якщо у браузері буде введено <serveraddress>/motorcycle/all, то програма виконає метод public MotorcycleDto[] getMotorcycles() {…}

1. Перевіремо роботу цього контролера. Для цього запустимо програму за допомогою Intellij IDEA

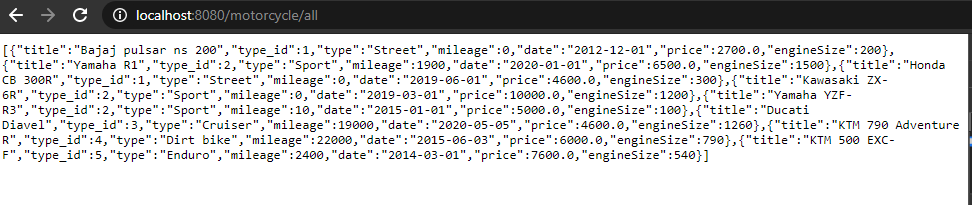


Отримаємо наступний лог у консолі

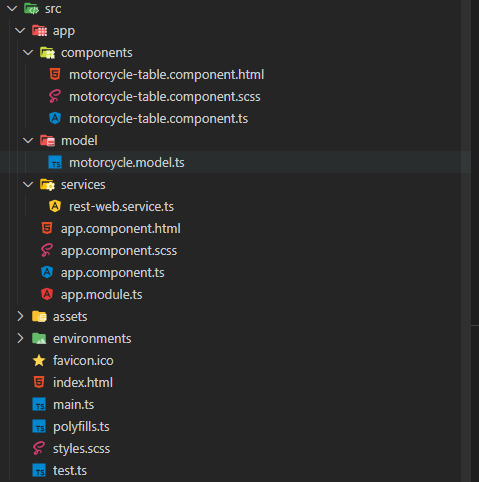


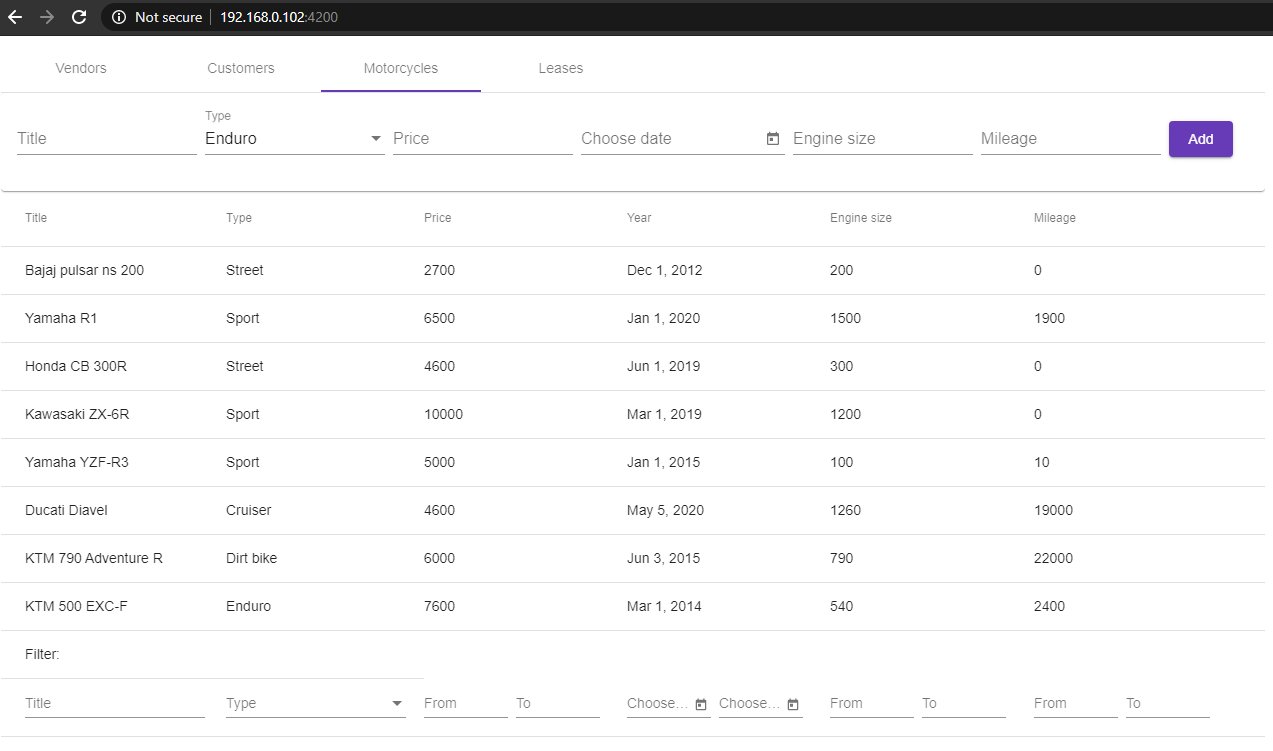
Слід зауважити, що ми повинні побачити лого spring у консолі, якщо все зроблено правильно

Перейдемо за посиланням <http://localhost:8080/motorcycle/all> та побачимо деяку відповідь від серверу у форматі JSON



1. Наступним кроком буде створення графічної оболочки за допомогою фреймворка angular 8. Більшу частину процесу створення ми опустимо так як це виходить за межі java. Нижче зазначений скріншот зі структурою проекту



Що виглядає наступним чином ****

# Висновки

Виконавши усі кроки зазначені в постановці задачі, ми оволоділи основними навичками написання Web застосунків з використанням таких засобів як Spring, Hibernate, PostgressDB, Angular 8. У процесі виконання завдання, ми практично отримали два проекти: Rest api та графічний інтерфейс на angular 8. Отримані результати будуть опубліковані на сервісі github за посиланням у додатках. Технології, зазначені при виконанні роботи були вибрані через свою популярність у колі ентерпрайз розробки ПЗ, вони користуються довірою багатьох розробників та використовуються майже у всіх проектах написаних на мові java. Тип проекту (Rest api) був обраний через свою розповсюдженість у сучасному підході до розробки веб застосунків.

# Додаток

<https://github.com/Noderoid64/JavaUniversityWork> - посилання на сторінку github