



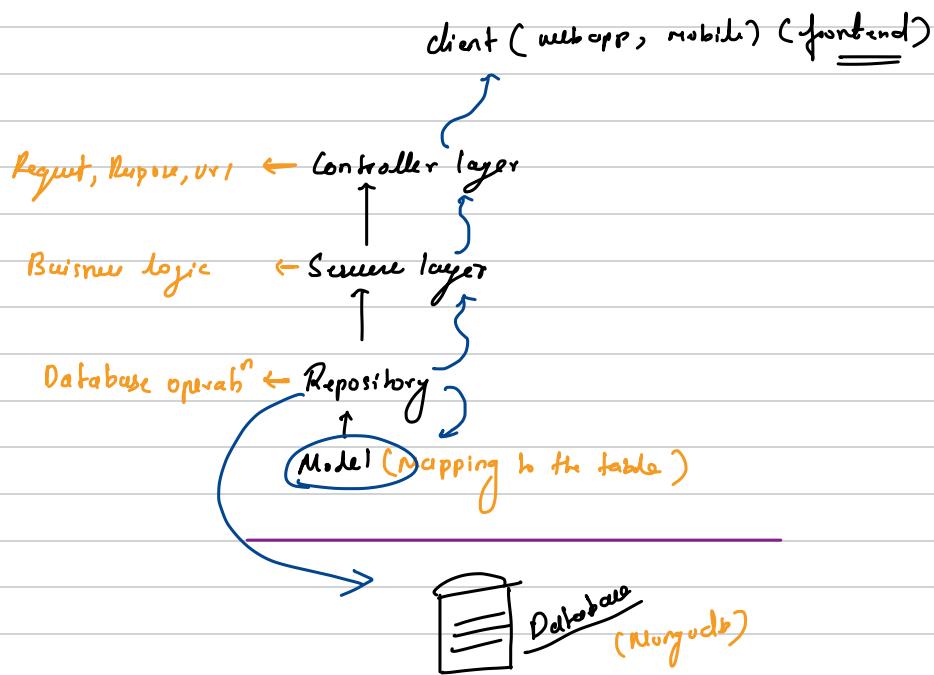
Agenda

- API REST
- DTO Layer
- |

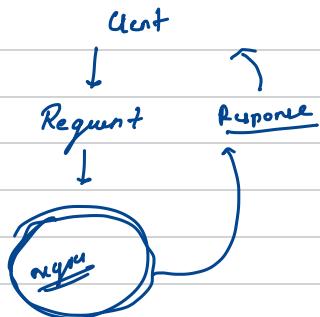
↳ created Spring boot (spring startkr.io) + IntelliJ

↳ Controller = Restcontroller.

3) . . .

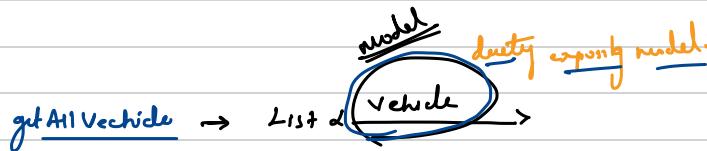


Controller



- ① @RestController → class is Rest -
- ② @RequestMapping → base api paths

GET → @GetMapping
 POST → @PostMapping
 PUT → @PutMapping
 DELETE → @DeleteMapping.



get VehicleBy id → Response Entity <Vehicle> " /vehicle/{id}"
(@PathVariable) ↗ ↘ argument.

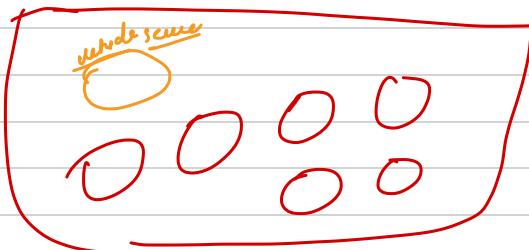
create Vehicle → " " " "

(@RequestBody) ↗ ← argument.

③ Auto wired.

private VehicleService service;

IOC Container



Component scans

Service days Bureau type

@Service.

public class VehicleService {



Repository layer : Database operations using help of Model class.

dependency → spring boot static data
non db.

@Repository

public class

VehiclesRepository {

private MongoTemplate tmp;

- ↳ find() → Vehicle.class
- ↳ findByid (id, Vehicle.class)
- ↳ save (Vehicle)
- ↳ remove()
- ⋮

API Design principle

open api

Nouns

GET

/api/v2/Vehicles.

/api/v2/Vehicles/1

/api/v2/Vehicles?brand=bmw

/api/v2/Vehicles?brand=bmw&year=2015.

x Verb x

+ /api/v2/getAllVehicles

+ /api/v2/getVehicleById

+ /api/v2/getVehicleByBrand

⇒ /getVehicleByBrandAndYear

success

status code

Spring way of writing it

GET ✓

200

Response Entity.ok().body(object)

CREATE ✓

201

RF status(201).body(object)

GET / update

404

RE. not found(404).body();

POST : ↴

400

RF. Bad request.body();

|||

s

500

503

→ DTO layer

Data Transfer object

User Model

id

name

email

username

↑

password

address

pan

create Vehicle (Vehicle route)

→ Request

Vehicle Request DTO

Vehicle Response DTO

return Response

