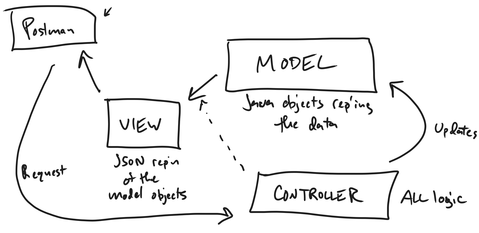
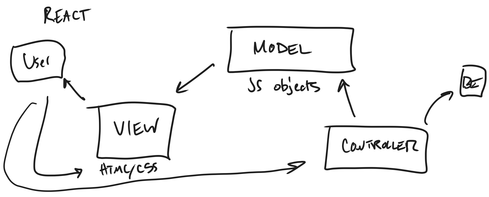
SPRING (BOOT) FRAMEWORK

- Backend framework for REST APIs
- MVC Framework
  - ↳ Model - current representation of data
    View - what the user sees/receives
    Controller - any logic that updates the model





(BACK TO SPRING)

- Maven to build/launch
- Dependencies (Spring by default)

basic REST API:
- MySQL driver
- Spring Data JPA (Java Persistence API)
  - incl. Hibernate ORM
    - ↳ Object-Relational Mapping
  - also things like @Transactional, @Entity
- Spring Web
  - comes with a default embedded Tomcat server

- devtools — auto-restart on save, etc.
- Spring Security — basic auth, cors, csrf, OAuth, etc.
- Spring Actuator — metrics about a running server
- Spring Cloud — microservices

FILES

- pom.xml — same function
- application.properties / application.yml
  - configuration file for app
- resources folder in use!

Application Context

- overall environment in which the app is running
- anything made available in the context can be used
- Beans
  - methods/classes made available to the context
  - you can inject Beans w/o instantiating them directly
- Dependency Injection
  - if Class A depends on an instance of Class B
  - I can "inject" it if B is a Bean
  - In Spring, DI uses IOC
    - Inversion Of Control
  - Context can only have ONE Bean of each type

Spring does a LOT of auto configuration
- searches for Beans within the package structure and makes them available