



Database project with JDBC & Spring

Slides by
Fahad Shaon



Objective

Show how to build a
web application with
database in **Java**



Assumptions

◎ Familiar with SQL

- create table customer (
 id int NOT NULL AUTO_INCREMENT,
 name varchar(255) NOT NULL
)
- insert into customer (id, name) values (1, 'customer')
- select * from customer;

◎ Familiar with Java

Topics

- ◎ JDBC
- ◎ JPA & Hibernate
- ◎ Spring Framework
- ◎ Web Application Basics
 - Servlet
- ◎ Spring MVC & Spring Boot



Softwares Required

- ◎ Database: MySQL
- ◎ Java Development Kit (JDK)
- ◎ Build tool: Gradle
 - Manage dependency
- ◎ IDE: IntelliJ Idea





JDBC

Java Database Connectivity



JDBC - Java Database Connectivity

- ◎ Part of JDK
- ◎ Bridges the gap between Java and DB
- ◎ Project - `tasks-jdbc`
 - `InitialExample`
 - `PreparedStatementExample`
 - `TransactionExample`
- ◎ Need to repeat **a lot** of boilerplate code





JPA & Hibernate

Java Persistence API



JPA - Java Persistence API




◎ Java specification

- For accessing, persisting, and managing data between Java objects / classes and a RDBMS

◎ We represent database in OOP

- Table → Class
- Rows → Object of the class

- Relationships are represented with annotation
- 

Hibernate

- ◎ Hibernate is a framework - implements JPA

- ◎ Provides EntityManager that manages entity lifecycle

- ◎ Example: tasks-jpa

 - Entities: Item, Task
 - Management: Main



Spring

Spring Framework Intro

Background

- ◎ Initially written by Rod Johnson
 - Released Book "*Expert One-on-One J2EE Design and Development*"
- ◎ Original authors built a company
 - Interface21 → SpringSource
- ◎ SpringSource was purchased for **\$420M** by VMware in August 2009.
- ◎ Now part of Pivotal (Dell + VMWare spin-off)



Core feature - Dependency Injection

◎ Software design pattern

- choice of component to be made at runtime rather compile time

◎ Further Reading

- <http://www.martinfowler.com/articles/injection.html>
- http://en.wikipedia.org/wiki/Dependency_injection





Core feature - Bean Management

- ◎ Spring manages life cycle of beans
- ◎ Injects proper dependencies of class

Bean

- ◎ is a POJO
- ◎ is serializable
- ◎ has a no-argument constructor
- ◎ allows access to properties using getter and setter



Spring Data

- ◎ One of many Spring sub-projects
- ◎ Repository *[mostly Spring provided]*
 - Reusable classes to perform db queries
- ◎ Service *[Programmer defined]*
 - Contains business logic
- ◎ Example - tasks-spring
 - SpringApp
 - TaskRepository
 - TaskService

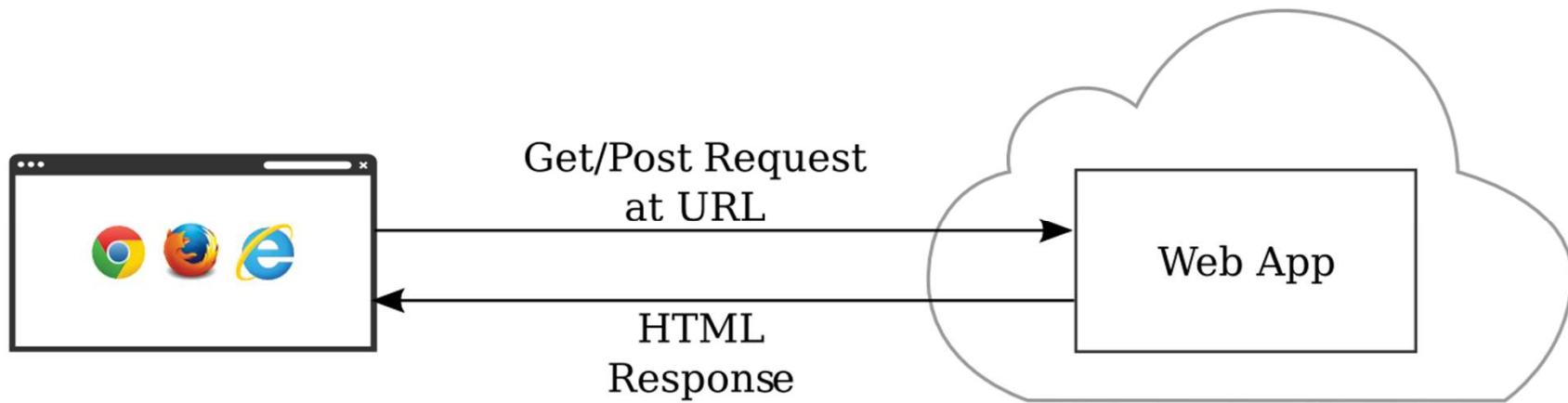


Web application

HTTP requests (Get/Post)
workflow



HTTP Request



Servlet

- ◎ Java Servlet API for responding to requests
- ◎ Can fit any client-server protocol
- ◎ Primarily used to map
 - GET and POST request in HTTP
- ◎ Example - `javax.servlet-jsp-hello-world`
 - `HelloServlet`



Spring MVC

Model-View-Controller
application using *Spring MVC*



MVC

Model

- ⦿ Communicate with database
- ⦿ Represent data in OOP
- ⦿ Encapsulate Business Logic

View

- ⦿ Provide user to input data
- ⦿ Display data to user

Controller

- ⦿ Map user request to methods
- ⦿ Control execution of business logic

Example Project `tasks-spring-mvc`

Thanks!

Questions?

🌐 <https://www.shaon.dev>

✉️ fs@shaon.dev

