Spring

* Java Framework
  + Java applications – Standalone app
  + Web MVC application – ModelViewController
  + RestFul application -- SpringBoot

Spring – IOC Container – InversionOfControl

DI - DependencyInjection

Benefits of Spring

Focus on Business logic

Configuration is maintained separately

Loosely Coupled Components

Business layer

Operator

Configuration

AddOperator

MultiplyOperator

looksup

MainClient

Configuration

Xml

Java classes

HelloWorld

1. Maven/Gradle - Projectmanagement tool - libraries—pom.xml
2. Quick start
3. Add spring dependency
4. Beans.xml in resources
5. Add entries for bean

Flow

1. Application context loaded
2. Xml file is loaded into memory
3. Xml file is read
4. HelloWorld Bean class is loaded (Singleton)
5. Bean object is created (getBean)
6. Method is called

OperatorDemo

AddOperator

operate(int x,int y)

Client

injectes

Operator (I)

operate(int,int)

MultiplyOperator

operate(int x,int y)

DispatcherServlet

Controller

(Java class)

path

MVC App/web app

View

Html

Jsp

Java Classes

(Model)

Spring MVC

1. Create maven -webapp
2. Add dependency for spring mvc
3. Configure DispatcherServlet in web.xml
4. Spring configuration xml -- <servlet-name>-servlet.xml Eg: spring-servlet.xml in WEB-INF