

Spring Framework Overview & Tools

Sang Shin

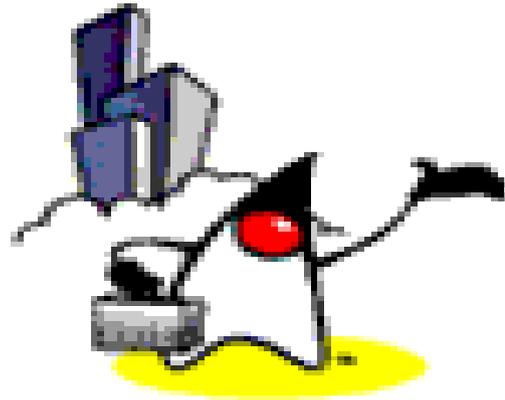
JPassion.com

“Code with JPassion!”



Topics

- What is Spring framework?
- Why Spring framework?
- Spring framework architecture
- Usage scenario
- Tools (we are going to use in this course)



What is Spring Framework?

What is Spring Framework?

- Light-weight yet comprehensive framework for building various types of Java applications
 - Web applications
 - Enterprise applications
 - Standalone applications
 - Batch application
 - Integration application
 - Mobile application

Things you can build with Spring

New Spring Starter Project

Boot Version: 1.2.5

Dependencies:

- Cloud
 - AWS
 - AWS JDBC
 - AWS Messaging
 - Cloud Bootstrap
 - Cloud Bus AMQP
 - Cloud Connectors
 - Cloud Security
 - Config Client
 - Config Server
 - Eureka
 - Eureka Server
 - Feign
 - Hystrix
 - Hystrix Dashboard
 - OAuth2
 - Ribbon
 - Turbine
 - Turbine AMQP
 - Zuul
- Core
 - AOP
 - Atomikos (JTA)
 - Bitronix (JTA)
 - Cache
 - DevTools
 - Security
- Data
 - Elasticsearch
 - Gemfire
 - JDBC
 - JPA
 - MongoDB
 - Redis
 - Solr
- Database
 - Apache Derby
 - H2
 - HSQLDB
 - MySQL
 - PostgreSQL
- I/O
 - AMQP
 - Batch
 - Integration
 - JMS
 - Mail
- Ops
 - Actuator
 - Remote Shell
- Social
 - Facebook
 - LinkedIn
 - Twitter
- Template Engines
 - Freemarker
 - Groovy Templates
 - Mustache
 - Thymeleaf
 - Velocity
- Web
 - HATEOAS
 - Jersey (JAX-RS)
 - Mobile
 - Rest Repositories
 - Vaadin
 - WS
 - Web
 - Websocket

< Back Next > Finish Cancel

Key Features - DI

- Bean wiring is done through the **Dependency Injection (DI)**
 - This aims to eliminate manual wiring of beans
- A core bean factory, which is usable globally
 - Spring MVC uses it internally

Key Features - Persistence

- Comprehensive RDBMS support
 - Generic abstraction layer for database transaction management
 - Higher abstraction over JDBC
 - Integration with persistence frameworks such as Hibernate, JPA
- NoSQL support
 - MongoDB

Key Features - Web-Tier

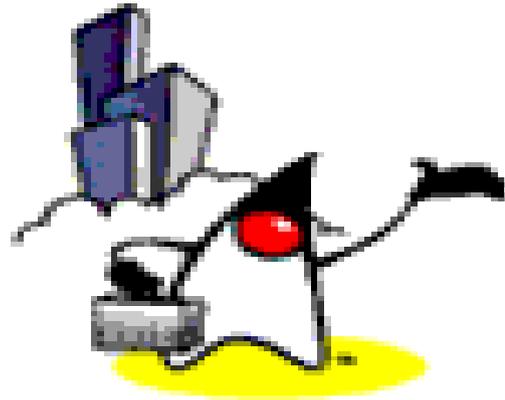
- Spring MVC web application framework
 - Built on core Spring functionality
 - Supports many technologies for generating views, including Thymeleaf, Velocity, Freemarker, and JSP, etc
- Spring Web Flow
 - Navigation logic is externalized
- REST support
 - Simple to create RESTful service
- Default over configuration
 - Everything is configurable and customizable

Key Features - AOP

- Extensive aspect-oriented programming (AOP) framework for providing services such as transaction management, security support
- As with DI, this aims to improve the modularity of systems created using the framework

Key Features - Test

- Supports Unit testing and Integration testing of Spring components
- Supports both JUnit and TestNG
- Provides consistent loading of Spring ApplicationContexts and caching of those contexts
- Provides mock objects that you can use to test your code in isolation



Why Use Spring Framework?

Why Use Spring?

- Wiring components (Beans) through Dependency Injection (DI)
 - Promotes de-coupling among the parts that make up an application
- Design to Java interfaces
 - Insulates a user of a functionality from implementation details
- Test-Driven Development (TDD)
 - POJO classes can be tested without being tied up with the framework

Why Use Spring? (Continued)

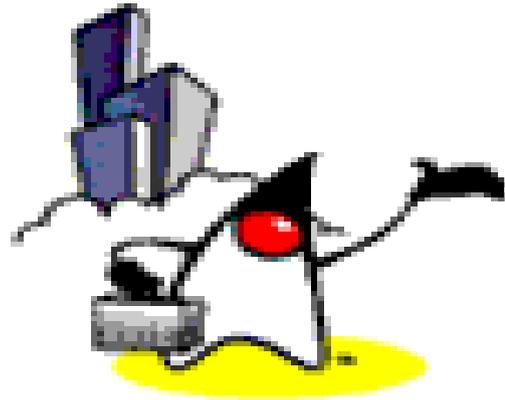
- Declarative programming through AOP
 - Transaction and security can be easily and declaratively configured
- Simplify use of popular technologies
 - Abstractions insulate application from specifics, eliminate redundant code
 - Underlying technology specifics still accessible
 - Handle common error conditions

Why Use Spring? (Continued)

- Conversion of checked exceptions to unchecked
 - (Or is this a reason not to use it?)
- Not an all-or-nothing solution
 - Extremely modular and flexible
- Well designed
 - Easy to extend
 - Many reusable classes

Why Use Spring? (Continued)

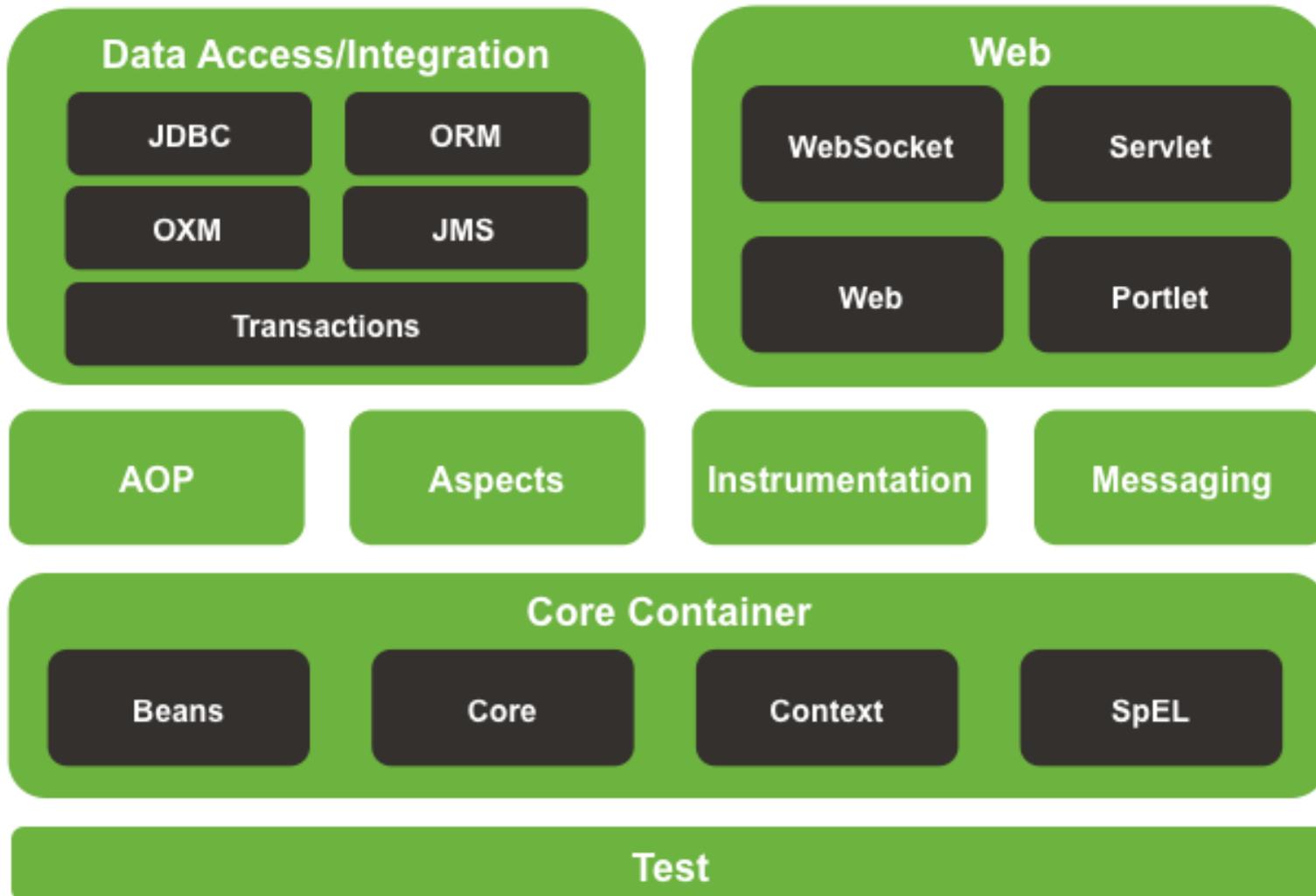
- Integration with other technologies
 - JPA, Hibernate, JDBC (for data access)
 - Thymeleaf, Velocity, etc. (for presentation)
 - JSF, Wicket, Struts, etc (For web)
 - AngularJS, jQuery, JavaScript (for front-end UI)
 - ActiveMQ, AMQP (for messaging)
 - ...

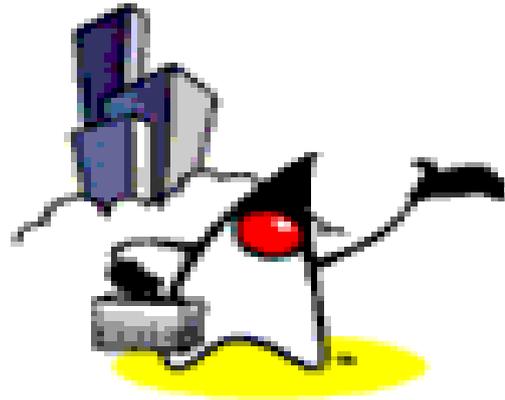


Spring Framework Architecture



Spring Framework Runtime



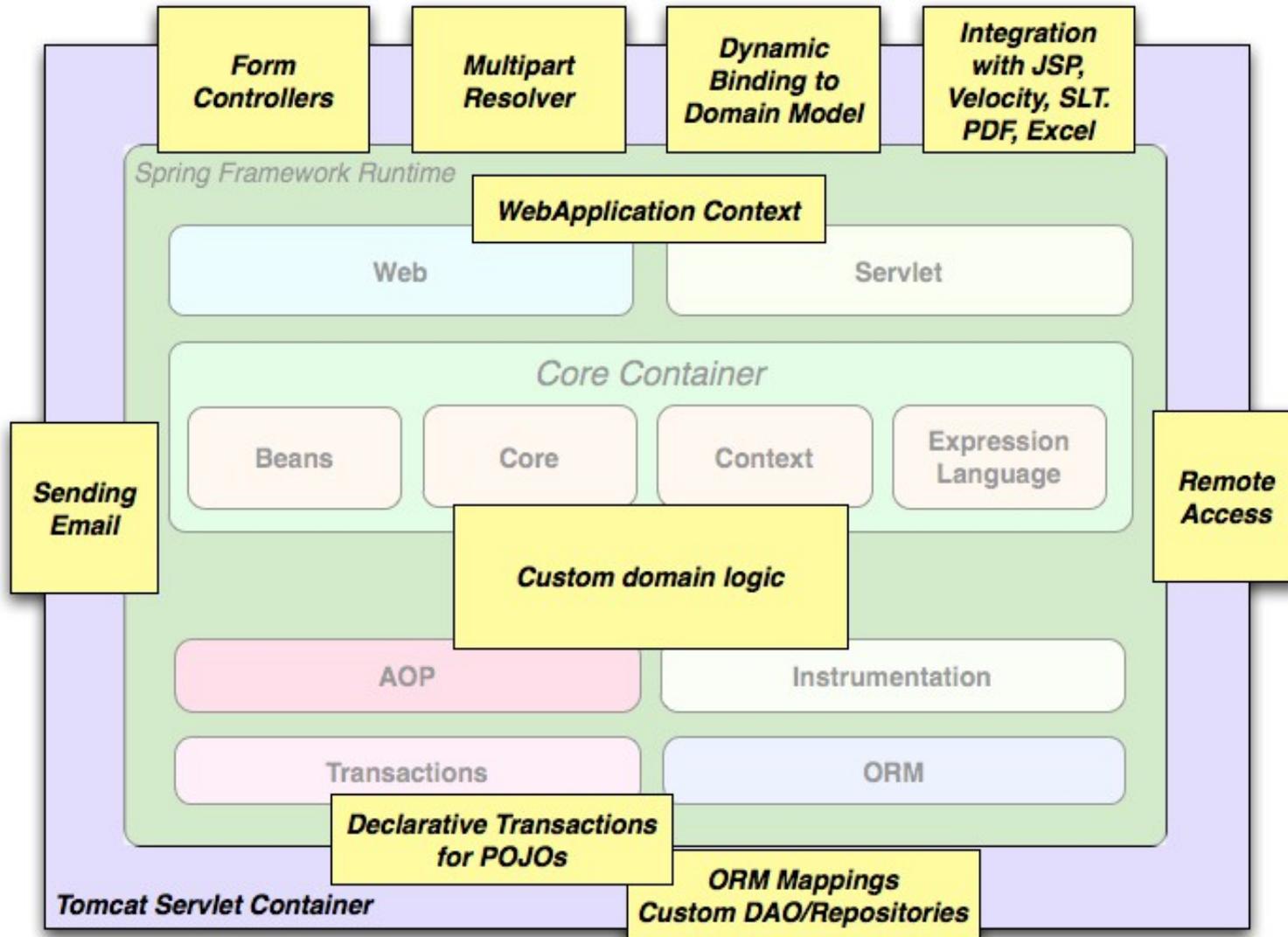


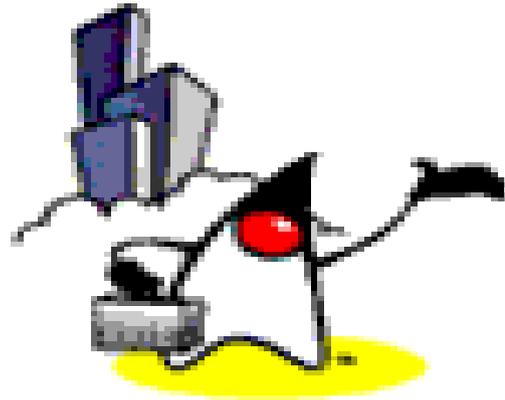
Usage Scenarios

Usage Scenarios

- You can use Spring in all sorts of scenarios, from applets up to fully-fledged enterprise applications using Spring's transaction management functionality and web framework integration

Typical Full-fledged Spring Web Application





Tools

SpringSource Tool Suite (STS)

- Eclipse-powered development environment for building Spring-powered enterprise applications
 - Supports Java, Spring, Groovy and Grails
- Comes with developer edition of vFabric tc Server
 - Pivotal tc Server is Tomcat that is optimized for Spring
 - You can use Tomcat and other servers with STS

Lab:

Exercise 1: Download & Install STS

**Exercise 2: Build and run a simple SpringMVC
Application using STS
4931_spring_sts.zip**



Code with Passion!
JPassion.com

