

SPRING FRAMEWORK



OUTLINE

- Introduction to Spring Framework
- Spring modules



INTRODUCTION TO SPRING FRAMEWORK





WHAT IS SPRING?

“Most popular Application Development
Framework for Enterprise Java”

<http://www.springsource.org>



SPRING OVERVIEW

- The Spring Framework is an application framework and inversion of control container for Java platform
- It is an open source layered application framework
- Developed by Rod Johnson for Apache 2.0 in Jun 2013
- Philosophy: J2EE should be easier to use, based on “Lightweight Container” concept.

WHAT ARE LIGHTWEIGHT FRAMEWORKS?

- Non-intrusive
- No container requirements
- Simplify application development
 - Remove re-occurring pattern code
 - Productivity friendly
 - Unit test friendly
- Very pluggable
- Usually open source
- Examples:
 - Spring, Pico, Hivemind
 - Hibernate, IBatis, Castor
 - WebWork
 - Quartz
 - Sitemesh



SPRING IS LIGHTWEIGHT

- Lightweight in terms of size
 - Entire Spring framework distributable in a JAR file is in few MBs only
- Lightweight in terms of overhead
 - Processing overhead required by Spring is negligible
- Non-intrusive
 - Objects in Spring-enabled application have no dependencies on Spring-specific classes

SPRING IS A CONTAINER

- Container
 - Builds the environment in which all other software components live
- Why Spring is a container?
 - Because it creates the components of your application and the components are children of the container
 - Contains and manages life cycle and configuration of application objects
 - Manages creation and configuration of each bean and its association with other beans



SPRING IS A FRAMEWORK

- Framework
 - Collection of components that you can use to build your applications.
- Why Spring is a framework?
 - Because it provides components to build common parts of applications, such as data access support, MVC support, and many others.
 - Makes it possible to configure and compose complex applications from simpler components
 - Provides infrastructural functionality, leaving only development of application logic to programmer

ARCHITECTURAL BENEFITS OF SPRING

- Spring can effectively organize your middle tier objects
- Spring takes care of plumbing that would be left up to you if you use only Struts or other framework
- Spring is designed so that applications built with it depend on as few of its APIs as possible.
- Spring can facilitate good programming practice by reducing the cost of programming to interfaces, rather than classes, almost to zero.
- Applications built using Spring are very easy to unit test.
- Spring provides a consistent framework for data access
- Spring provides a consistent, simple programming model in many areas, making it an ideal architectural "glue."



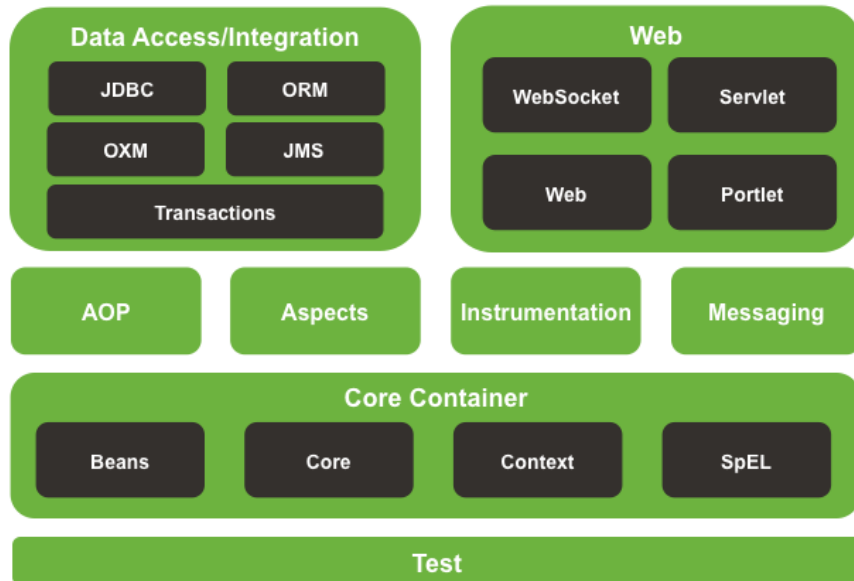
SPRING MODULES



SPRING MODULES



Spring Framework Runtime



- The Spring Framework consists of features organized into about 20 modules.
- These modules are grouped into Core Container, Data Access/Integration, Web, AOP (Aspect Oriented Programming), Instrumentation, Messaging, and Test, as shown in the diagram

CORE CONTAINER

- This is the base module of Spring.
- It consists of spring-core, spring-beans, spring-context, spring-context-support, and spring-expression
- **Spring-core and spring-beans** provide the fundamental parts of the framework, including IOC and Dependency Injection features
- The **context** module inherits its features from beans module and adds support for internationalization, event propagation and resource loading. It also supports Java EE features such as EJB, JMX, and basic remoting
- **Spring-context-support** provides support for integrating common third party libraries for caching, mailing, scheduling and template engines
- **Spring-expression** module provides a powerful Expression Language for querying and manipulating an object graph at runtime



AOP AND INSTRUMENTATION

- The spring-aop module provides an aspect-oriented programming implementation.
- AOP enables defining method-interceptors and pointcuts to decouple code that should be separated.
- The separate spring-aspects module provides integration with AspectJ
- The spring-instrument module provides class instrumentation support and classloader implementations to be used in certain application servers

DATA ACCESS/INTEGRATION

- This layer consists of the JDBC, ORM, OXM, JMS and transaction modules
- The **spring-jdbc module** provides a JDBC-abstraction layer that removes the need to do tedious JDBC coding and parsing of database-vendor specific error codes.
- The **spring-tx module** supports programmatic and declarative transaction management for classes that implement special interfaces and for *all your POJOs (Plain Old Java Objects)*.
- The **spring-orm module** provides integration layers for popular object-relational mapping APIs, including JPA, JDO, and Hibernate
- The **spring-oxm module** provides an abstraction layer that supports Object/XML mapping implementations such as JAXB, Castor, XMLBeans, JiBX and XStream.
- The **spring-jms module** (Java Messaging Service) contains features for producing and consuming messages



WEB

- The Web layer consists of the spring-web, spring-webmvc, spring-websocket and spring-webmvc-portlet modules
- The **spring-web** module provides basic web-oriented integration features such as multipart file upload functionality and the initialization of the IoC container using Servlet listeners and a web-oriented application context
- The **spring-webmvc** contains Spring's MVC and REST Web service implementation for web application
- The **spring-webmvc-portlet** module provides the MVC implementation to be used in a Portlet environment



TEST

- The spring-test module supports the unit testing and integration testing of Spring Components.
- It also provides mock objects that you can use to test the code in isolation