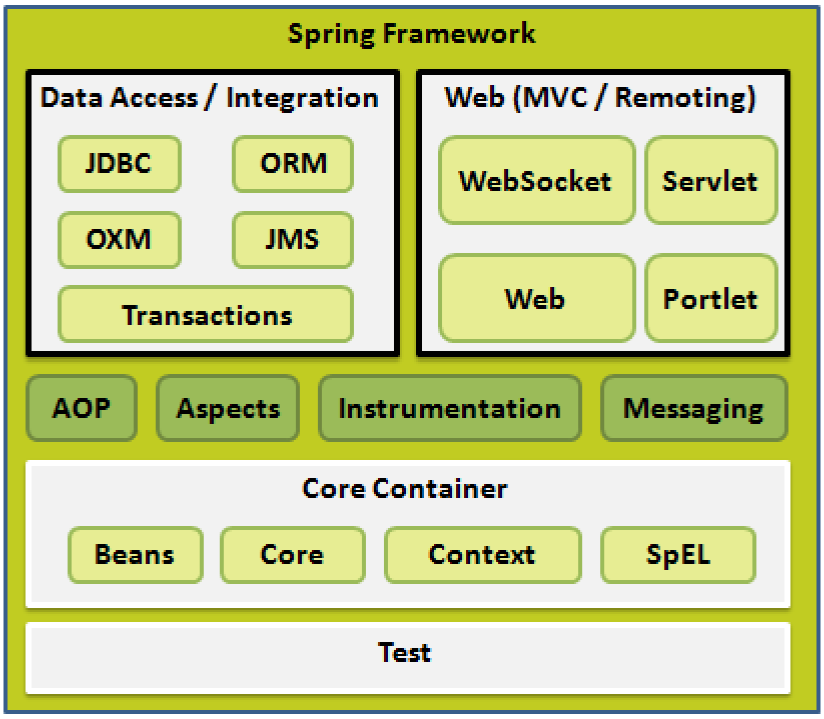
SPRING FRAMEWORK - ARCHITECTURE

Spring could potentially be a one-stop shop for all your enterprise applications, however, Spring is modular, allowing you to pick and choose which modules are applicable to you, without having to bring in the rest. Following section gives detail about all the modules available in Spring Framework.

The Spring Framework provides about 20 modules which can be used based on an application requirement.



Core Container:

The Core Container consists of the Core, Beans, Context, and Expression Language modules whose detail is as follows:

- The **Core** module provides the fundamental parts of the framework, including the IoC and Dependency Injection features.
- The **Bean** module provides BeanFactory which is a sophisticated implementation of the factory pattern.
- The **Context** module builds on the solid base provided by the Core and Beans modules and it is a medium to access any objects defined and configured. The ApplicationContext interface is the focal point of the Context module.
- The **SpEL** module provides a powerful expression language for querying and manipulating an object graph at runtime.

Data Access/Integration:

The Data Access/Integration layer consists of the JDBC, ORM, OXM, JMS and Transaction modules whose detail is as follows:

- The **JDBC** module provides a JDBC-abstraction layer that removes the need to do tedious JDBC related coding.
- The **ORM** module provides integration layers for popular object-relational mapping APIs, including JPA, JDO, Hibernate, and iBatis.
- The **OXM** module provides an abstraction layer that supports Object/XML mapping

implementations for JAXB, Castor, XMLBeans, JiBX and XStream.

- The Java Messaging Service **JMS** module contains features for producing and consuming messages.

- The **Transaction** module supports programmatic and declarative transaction management for classes that implement special interfaces and for all your POJOs.

Web:

The Web layer consists of the Web, Web-MVC, Web-Socket, and Web-Portlet modules whose detail is as follows:

- The **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 **Web-MVC** module contains Spring's model-view-controller MVC implementation for web applications.

- The **Web-Socket** module provides support for WebSocket-based, two-way communication between client and server in web applications.

- The **Web-Portlet** module provides the MVC implementation to be used in a portlet environment and mirrors the functionality of Web-Servlet module.

Miscellaneous:

There are few other important modules like AOP, Aspects, Instrumentation, Web and Test modules whose detail is as follows:

- The **AOP** module provides aspect-oriented programming implementation allowing you to define method-interceptors and pointcuts to cleanly decouple code that implements functionality that should be separated.

- The **Aspects** module provides integration with AspectJ which is again a powerful and mature aspect oriented programming AOP framework.

- The **Instrumentation** module provides class instrumentation support and class loader implementations to be used in certain application servers.

- The **Messaging** module provides support for STOMP as the WebSocket sub-protocol to use in applications. It also supports an annotation programming model for routing and processing STOMP messages from WebSocket clients.

- The **Test** module supports the testing of Spring components with JUnit or TestNG frameworks

Loading [MathJax]/jax/output/HTML-CSS/jax.js