# Spring and SpringMVC

**Group** EDG

## Spring

Spring is an open source framework, which is created to solve the complexity of enterprise application development. One of the main advantages of the framework is its layered architecture, which allows you to choose which component to use, while providing an integrated framework for J2EE application development.
Spring framework is a layered architecture consisting of seven defined modules. Spring module is built on top of the core container, which defines how beans are created, configured, and managed.

## Advantages and Disadvantages

**Advantages**
1.It provides a method of managing objects, which can effectively organize the objects in the middle layer. A perfect frame "glue".
2.Use Spring's IOC container to transfer the dependency between objects to Spring, reduce the coupling between components, and let us focus on application logic.
3.The hierarchical structure is adopted, which can be added to the project incrementally.
4.AOP is well supported, and convenient face-to-face programming is conducive to the cultivation of interface oriented programming habits.
5.Spring is of low intrusion and low code pollution. The dependence of application on Spring API can be minimized.
6.Provide good integration support for mainstream frameworks, such as Hibernate, Struts 2, JPA, etc.
7.Spring DI mechanism reduces the complexity of business object replacement.

**Disadvantages**
1.Interrupt the logic of the application program, making the code incomplete and not intuitive. At this time, only from the Source can not fully grasp all the behaviors of the application.
2.Configure the logic that should have been coded, increasing the chance of error and additional burden.

## Comparison

The creation of Spring container in contextloaderlistener is mainly used for some components that the whole web application needs to share, such as Dao, connectionfactory of database, etc.; while the container of SpringMVC created by dispatcher servlet is mainly used for some components related to the servlet, such as controller, viewresolver, etc. The relationship between them is as follows:

1.Range The child container (SpringMVC container) can access the bean of the parent container (Spring container), and the parent container (Spring container) cannot access the bean of the child container (SpringMVC container). That is to say, when getBean is in SpringMVC container, if the corresponding bean cannot be found in its own container, it will go to the parent container, which also explains why the controller created by SpringMVC container can get the service component created by Spring container.
2.Specific implementation In the specific implementation of Spring, both the child container and the parent container are put into the ServletContext through the setAttribute method of ServletContext. However, contextloaderlistener will create ApplicationContext before dispatcher servlet. When dispatcher servlet creates ApplicationContext, it will first find ApplicationContext created by contextloaderlistener, and then pass the latter ApplicationContext as a parameter to the setparent() method of ApplicationContext of dispatcher servlet. In other words, the creation of a child container depends on the creation of the parent container, which is created before the child container.
In general, Spring is the container framework of IOC and AOP. SpringMVC is a web framework added based on Spring functions. If you want to use SpringMVC, you must first rely on Spring. In a simple way, you can compare SpringMVC to struts. Spring is the container framework of IOC and AOP. SpringMVC is a web framework added based on Spring functions. If you want to use SpringMVC, you must first rely on Spring.

## Summary

Spring is like an engine,and Spring MVC is an MVC framework based on Spring.Spring's two core AOP and IOC can be used independently for any application, including integration with MVC frameworks such as struts and ORM frameworks such as hibernate. At present, many companies use spring + struts (2) + Hibernate as their so-called lightweight development. Spring MVC is an MVC framework. Spring MVC annotation type development is more convenient than struts 2, and can directly replace the above struts (of course, as a very mature MVC, the function of struts is better than spring, but spring MVC is enough). Of course, the execution efficiency of spring MVC is higher than that of struts, because the value stack of struts affects the efficiency.

**Figure 1. The seven modules of the Spring framework**

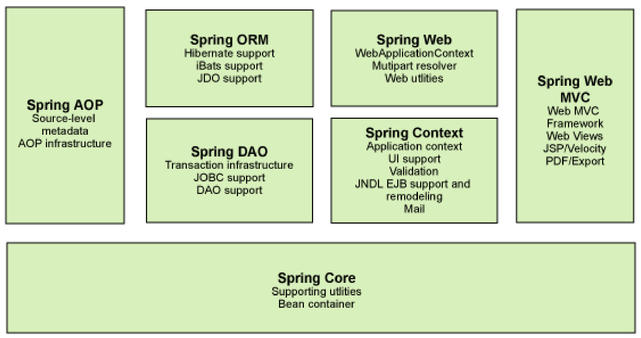
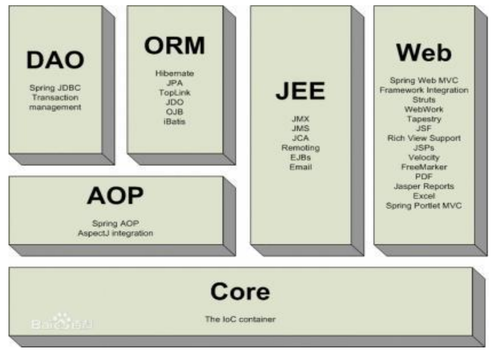



**Figure 2. The sixth modules of the SpringMVC framework**




## SpringMVC

Spring MVC is a module of spring framework. Spring MVC and spring do not need to be integrated through the middle integration layer. Spring MVC is a web framework based on MVC. It receives external requests and parses parameters to the service layer MVC refers to the design concept of C control layer, m module layer and V display layer. Spring  can also be understood as constraints) to develop web projects according to the design of MVC. Spring, the other two frameworks, is mainly used as IOC, For other design principles such as AOF, mybatis is used to operate the database conveniently, so they are all in the MV. As for V, it refers to the display part, and generally refers to JSP

## Advantages and Disadvantages

**Advantages**
Package code, low maintenance cost and low coupling In MVC mode, each layer performs its own functions, so once the requirements of one layer change, only the code in the corresponding layer needs to be changed without affecting the code in other layers. It is conducive to the division of labor in development and improves development efficiency In MVC mode, because the system is separated by layers, the division of labor in development can be better realized. Component reuse is conducive to code reuse and high reusability After layering, it is more conducive to the reuse of components.

**Disadvantages**
Not suitable for small, medium-sized applications Spending a lot of time Applying MVC to applications that are not very large often pays off. Increase the complexity of system structure and Implementation For a simple interface, strictly follow MVC to separate the model, view and controller, which will increase the complexity of the structure, and may produce too many update operations, reducing the operating efficiency. Too close connection between view and controller View and controller are separated from each other, but they are closely related components. View has no controller, and its application is very limited.

## Index



## References

https://www.cnblogs.com/rainbow70626/p/9784938.html
https://blog.csdn.net/qq_35251019/article/details/74963347
https://www.cnblogs.com/hafiz/p/5875740.html

## Division of labor

Chen Wang(@zhaoboy02)->Spring
Mengdian Xie(@Mengdian-Xie)->SpringMVC
Yimin Zhang(@xiaozhuzhu2333)->Comparison
Junyi Li(@lijunyi3399)->Summary
Yizhao Yang and Mengjie Liu(@liumengjie0119)->Poster
Zhao Xu and Rui Zhang->Presentation