

# The design and implementation of the framework for Spring+SpringMVC+MyBatis in the development of Web application

Yuxiang Hou

Nanchang Institute of Science & Technology, China

**Keywords:** Web application; SpringMVC; Spring; Mybatis; Framework

**Abstract:** In the development and design of Web application, the choice of framework development is very important. A good framework development can speed up the development of Web application, reduce development costs, and reduce the workload of developers. At the same time, Web application has good expansibility and portability. The framework based on SpringMVC + Spring + Mybatis (SSM) has good performance and rapid development efficiency, gradually become a mainstream development framework of Web application.

## 1. The Introduction of Spring, SpringMVC, MyBatis

### 1.1 Spring

Spring is an open source framework, and Spring is a lightweight Java development framework that emerged in 2003. It is created to solve the complexity of enterprise application development. One of the main strengths for the framework is its layered architecture, which allows users to choose which component to use, at the same time, provides integrated frameworks for J2EE application development. Spring uses the basic JavaBean to accomplish things that could have been done by EJB only before. However, the use of Spring is not limited to server-side development. From the point of view of simplicity, testability and loose coupling, any Java application can benefit from Spring. The core of Spring is Inversion of Control (IoC) and Aspect Oriented Programming (AOP). In simple terms, Spring is a layered JavaSE/EEfull-stack (one-stop) lightweight open source framework.

### 1.2 SpringMVC

SpringMVC belongs to follow-up products of Spring framework, has been integrated into Spring Web Flow. The Spring framework provides an full-function MVC module for building Web applications. SpringMVC decouples controllers, model objects, dispatchers, and the roles of handler objects, which made them easier to customize.

SpringMVC is a powerful and flexible web framework provided by Spring. With annotations, SpringMVC provides almost POJO development patterns which made the development and testing of the controller easier. This kind of controllers normally does not process the requests directly, instead, delegates these requests to other beans in the Spring context, these beans are injected into the controller through the dependency injection function of Spring.

SpringMVC is mainly composed of DispatcherServlet, processor mapping, processor (controller), view parser and view. It's two cores as below:

Processor mapping: choose which controllers to use to process the requests

View parser: choose the results how to be rendered

Through above two points, SpringMVC guarantees how to select, control and process the requests, and how to select views to show loose coupling between outputs.

### 1.3 MyBatis

MyBatis is an open source project iBatis of apache, this project was moved to Google code from apache software foundation in 2010, and renamed MyBatis.

MyBatis is an excellent persistence framework that supports general SQL queries, stored

procedures and advanced mapping. MyBatis eliminates almost all JDBC code, the manual setup of parameters and the retrieval of result sets. MyBatis uses simple XML or annotations for configuration and original mapping, to map the interface and Java's POJOs (Plain Old Java Objects, normal Java objects) into the records of the database.

Each MyBatis application program is mainly used with the SqlSessionFactory instance, and a SqlSessionFactory instance can be obtained through the SqlSessionFactoryBuilder. The SqlSessionFactoryBuilder can be obtained from an xml configuration file or a predefined configuration instance.

Building an SqlSessionFactory instance with an xml file is a very simple thing. It is recommended to use the classpath resource in this configuration, but you can use any Reader instance, including instances created with file paths or url at the beginning of file://. MyBatis has a utility class---Resources, which has a number of methods, can easily load Resources from class paths and other locations.

## 2. The integration of SSM framework

### 2.1 SSM framework

The SSM framework consists of three open source frameworks of Spring, SpringMVC and MyBatis, which are often used as a framework for web projects which the data source is simpler.

### 2.2 Relationship between the various layers of SSM

SpringMVC is a control layer, spring is used to manage the business logic layer. Mybatis is used for the dao layer. In the architecture MVC, m refers to the model, which contains the service, dao and javabean (pojo), v refers to the view.

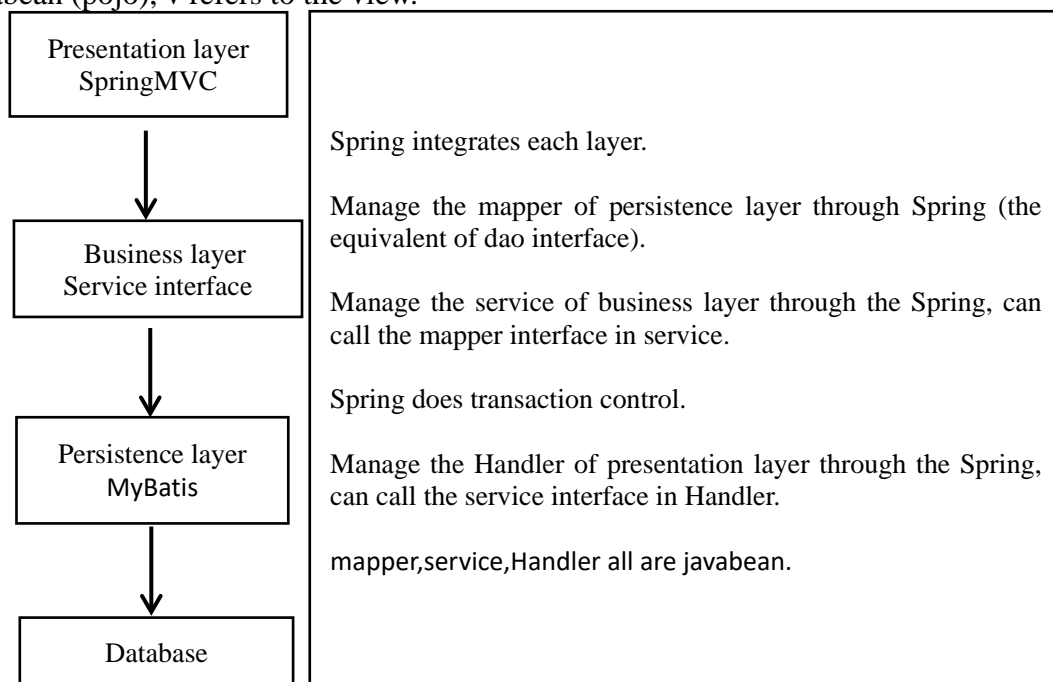


Figure 1. Structure of each layer relationship for SSM

The DAO layer (mapper) of persistent layer is invoked through MyBatis. DAO layer mainly do the work of data persistence layer, some of tasks responsible for contacting the database are encapsulated here, the design of the DAO layer first is to design the interface of DAO, and then to define implementation class of this interface in the configuration file of Spring, then can call this interface in the module to process the business of data, and don't need to care about the concrete implementation class of this interface is which class, structure seems to be very clear, the data source configuration of DAO layer and the parameters relating to database connections are configured in the configuration file of Spring.

The Service layer of Business layer: the Service layer is mainly responsible for the logical application design of business modules. First, design the interface, then design its implementation class, next, configure its implementation association in the configuration file of Spring. In this way, we can invoke the Service interface in the application to do business processing. Business implementation in the Service layer, in particular, need to call the defined interface in DAO layer, encapsulating the business logic of the Service layer is conducive to the independence and reuse of the general business logic, the procedure is shown very simple.

Controller layer (Handler layer): the Controller layer is responsible for the control of specific business module processes, in this layer, need to invoke the interface in Service layer to control the business process, the configuration of control is also processed in the configuration file of Spring, for the specific business process, there will be a different controller, in our specific design process, the process can be abstracted and summarized, to design the sub-unit process module which can be reused, this way not only makes the program structure clear, also greatly reduces the amount of code.

View layer: this layer is tightly coupled with the control layer, and requires the combination of the two to work together. The View layer is mainly responsible for the presentation of the foreground JSP page.

The links of each layer as follows:

The DAO layer and Service layer can be independently developed, their mutual coupling is very low, and can be carried out independently, such a pattern is especially advantageous in the development of large projects.

The Controller layer and View layer because of the high degree of coupling, so need to be developed together, but it can also be considered as a whole which is independent of the first two layers to be developed. In this way, we only need to know the definition of the interface before the layer and layer, and only need to call the interface, then can complete the necessary logic unit application, which is very clear and simple.

The design of Service logic layer is based on DAO layer, after set up the DAO layer, then can set up the Service layer, and Service layer is under the Controller layer, so Service layer not only need to invoke the interface of DAO layer, but also need to provide the interface for the class of Controller layer to make calls, it is just in a middle position. Each model has a Service interface, each interface encapsulates the respective business processing methods.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd">
  <display-name></display-name>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <servlet-name>springmvc</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>springmvc</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>
</web-app>
```

Figure 2. The configuration file code of Web.xml

## 2.3 The design and implementation of SSM framework

The design and implementation of Spring+SpringMVC+MyBatis in the development of Web application, the design as follows:

① In the development of web system, first to configure the file environment of web.xml, springmvc-servlet.xml.

The configuration file code of Web.xml is shown in Figure 2.

The configuration file of springmvc-servlet.xml is shown in Figure 3.

```
<mvc:annotation-driven/>
<context:component-scan base-package="com.ahead.controler"/>
<bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/"></property>
    <property name="suffix" value=""></property>
</bean>
```

Figure 3. The configuration file of springmvc-servlet.xml

② In the development of web system, to design the entity class, to define the properties of object, properties has the methods of get and set (can refer to the field of the table in the database to set it, the database should be designed before all the coding starts), to create construction methods.

Under the directory of SRC, the design of configuration file for mysql.properties database which has four properties of jdbc.driver, jdbc.url, jdbc.usernamejdbc, password.

③ Then, to design the Mapper.xml, first to customize it to return to the result set, the id properties in various labels must be the same as the methods in the interface, and the id property value must be unique, and cannot be reused. Take the student to query the web page design as an example, the result set code of student query function is shown in Figure 4.

```
<resultMap id="studentMap" type="com.ahead.beans.StudentInfoBean">
    <id property="studentId" column="studentId" javaType="java.lang.Integer"></id>
    <result property="studentCode1" column="studentCode1" javaType="java.lang.String">
</result>
</resultMap>
```

Figure 4. The result set code of student query function

④ Then to define the function of the web system, to correspond to the operations that are performed on the database, such as insert, selectAll, selectByKey, delete, update, and so on. The student inquiry code is shown in Figure 5.

```
<select id="selectAllStudent" resultMap="studentMap">
    select s.* from student_info s
</select>
```

Figure 5. The student inquiry code

⑤ The design of Mapper.java, to map the operations in Mapper.xml to Java functions according to id. The code of the student's query work interface is shown in Figure 6.

```
public interface StudentInfoMapper {
    public List<StudentInfoBean> selectAllStudent() throws Exception;
}
```

Figure 6. The code of the student's query work interface

⑥ The design of service.java, to provide services for the control layer, accept the parameters of

the control layer, complete the corresponding functions, and return to the control layer. The implementation code of student's inquiry method is shown in Figure 7.

```
public class StudentInfoService {
    public static List<StudentInfoBean> selectAllStudent(){
        SqlSession session=DBTools.getSession();
        StudentInfoMapper mapper=session.getMapper(StudentInfoMapper.class);
        List<StudentInfoBean> student=null;
        try {
            student=mapper.selectAllStudent();
            session.commit();
        } catch (Exception e) {
            e.printStackTrace();
            session.rollback();
        }finally{
            session.close();
        }
        return student;
    }
}
```

Figure 7. The implementation code of student's inquiry method

⑦ The design of Controller.java, to connect page requests and service layer, obtain the parameters of the page request, through the automatic assembly, to map a different URL to the corresponding processing function, and obtain parameters, process the parameters, and then pass it to the service layer. The code of student's query function in control layer is shown in Figure 8.

```
@Controller
@RequestMapping("/systemset/")
public class StudentInfoController {
    @RequestMapping("/studentlist")
    public String studentList(Model model) {
        model.addAttribute("studentList", StudentInfoService.selectAllStudent());
        return "systemset/StudentList.jsp";
    }
}
```

Figure 8. The code of student's query function in control layer

⑧ Finally, to design the JSP page invoking, what parameters are requested and what data is required.

In a word, the process for the design and implementation of SSM is shown in Figure 9.

```
DataBase ==> Entity==> Mapper.xml ==> Mapper.Java ==> Service.java Service layer ==>
Controller.java Control layer ==> jsp page
```

Figure 9. The process for the design and implementation of SSM

## 2.4 Operation process of SSM

Step 1: Jsp (view) sends the request.

Step 2: Through the invoking of core Controller DispatcherServlet to request parser: HandlendMapping parses the request, and matches to the Controller layer through the mapping relationship.

Step 3: To call business logic layer (service) in the control layer, data persistence layer (DAO) returns to the control layer, to request the completion to obtain a result, to set up a view which will be jumped, (ModelAndView loads and transmits the data, sets the view).

Step 4: The invoking of core controller, the view parser: ViewResolver parses view, to match the corresponding page to achieve page jumping.

## 3. Conclusion

This paper mainly introduces the design and implementation of the framework for Spring, SpringMVC and MyBatis in the development of Web application, simplifies the development process and workload of the system, to improve the expansion of the system and the convenience of deployment.

## References

- [1] Xiaolan Yang, Ming Luo. The design and implementation of online forum based on Spring+SpringMVC+MyBatis. Heilongjiang science and technology information, 2016 (36).
- [2] Yang Li. The design and implementation of SSM framework in the development of Web application. Computer technology and development, 2016 (12).
- [3] Hongting Zou. The research and application of Web system based on SSM framework. The journal of Hunan Institute of technology (Natural science edition), 2017 (01).
- [4] Qi Yin, Lisha Xu, Shaocheng Di. The implementation of second-hand book transaction system in University based on SSM. Computer knowledge and technology. 2017 (01).
- [5] Xiujuan Huang. Research on the application of Web system based on the frame of SpringMVC+Hibernate+Extjs. Computer knowledge and technology, 2015 (16).
- [6] Wenjia Xu, Yingkai Zhao. The application of SpringMVC and Hibernate for WEB development. Computer application and software, 2008 (02).