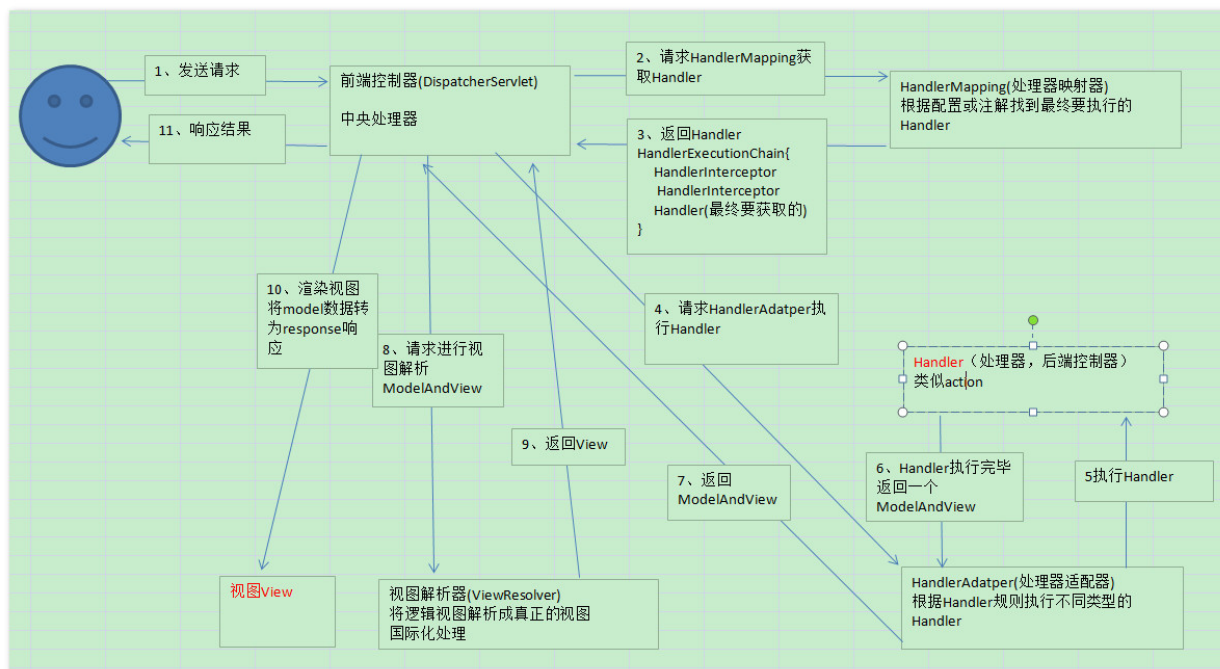


## 一.SpringMvc的工作原理:



用户发送请求到**前端控制器**,前端控制器对**请求分发,映射**,HandlerMapping基于**selvlet**做的,需要配置selvlet,路径对应的那个类对应响应.然后返回个**中央处理器**,返回Miderlandview**数据和视图**返回,然后渲染页面,反馈给用户

8,9找的是视图,2.3找的是映射关系,(路径和方法的关系)

## 二.SpringMvc的学习:

### 1.导包:

1. `spring-webmvc`
2. `spring-test`(test测试类注解对容器的初始化)
3. `javax.servlet-api`(相当于加上servlet,在本地使用的)

### 2.配置:

- 1 在recourse中新建:Spring-config和SpringMVC-config进行配置
- 2 通过前面的学习,我们知道,spring主要针对serise层,
- 3 springMVC主要是针对controller,mybatis主要是针对dao层的操作.

在web.xml中配置:首先,需要**初始化容器**,我们知道,在spring的学习中,初始化容器可以通过三种方式初始化:**xml,注解,测试类中通过注解初始**

化.

其次,spring提供**监听器**,监听Tomcat服务器的启动,自动初始化Spring容器,

我们需要配置spring的xml容器.找到spring的配置.

```
1 <!--Spring容器初始化方式:xml,注解,test测试类_spring注解-->
2
3 <!--Spring提供监听器,监听Tomcat服务器的启动,自动初始化Spring容器-->
4 <listener>
5   <listener-class>org.springframework.web.context.ContextLoaderLi
   stener</listener-class>
6 </listener>
7
8 <!--配置Spring的xml文件路径-->
9 <context-param>
10   <param-name>contextConfigLocation</param-name>
11   <param-value>classpath:Spring-config.xml</param-value>
12 </context-param>
13
14 <!--配置springMVC的中央处理器,用于做请求的分发处理,
15   是基于servlet实现的-->
16 <servlet>
17   <servlet-name>dispatcherServlet</servlet-name>
18   <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
19   <!--配置Springmvc的xml文件路径-->
20   <init-param>
21     <param-name>contextConfigLocation</param-name>
22     <param-value>classpath:SpringMVC-config.xml</param-value>
23   </init-param>
24   <!--
25   加载的优先级:
26   优先级为>=0时,随Tomcat启动加载
27   优先级为<0时,当这个servlet被调用时加载
28   -->
```

```

1  <!-- 设为1,当tomcat启动时,就对请求分发-->
2  <load-on-startup>1</load-on-startup>
3  </servlet>
4
5  <servlet-mapping>
6    <servlet-name>dispatcherServlet</servlet-name>
7    <url-pattern>*.do</url-pattern>
8  </servlet-mapping>

```

需要在SpringMvc-config中配置:配置的请求路径.<bean>使用了**注解**的方式,直接查找controller层的文件.运行javaee启动服务器的时,跳转到主页面,然后在url中输入/**hello.do**则显示,HelloController的内容.

```

1  <bean id="helloController" class="com.lanou.controller.HelloController"/>
2
3  <!-- 配置请求路径 -->
4  <bean id="handlerMapping" class="org.springframework.
5    web.servlet.handler.SimpleUrlHandlerMapping">
6    <property name="mappings">
7      <props>
8        <prop key="/hello.do">
9          helloController
10         </prop>
11       </props>
12     </property>
13   </bean>

```

在HelloController中实现了**controller**的接口,跳转到了一个JSP页面

```

1  public class HelloController implements Controller {
2    @Override
3    public ModelAndView handleRequest
4      (HttpServletRequest Request, HttpServletResponse Response) throws Exception {
5      ModelAndView modelAndView = new
6        ModelAndView("Hello.jsp");

```

```
7  return modelAndView;  
8  }  
9  }
```