

## 一.springmvc的配置:

### 1.xml中配置

```
1 <bean id="hellocontroller" class="com.lanou.controller.Hellocontroller">
2 </bean>
3
4 <bean id="handlerMapping" class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
5 <!-- 路径映射 -->
6 <property name="mappings">
7 <props>
8 <prop key="/hello.do">
9 hellocontroller
10 </prop>
11 </props>
12 </property>
13 </bean>
```

### 2.注解的方式配置:

```
1 <!-- 注解 -->
2 <context:component-scan base-package="com.lanou.controller"/>
3
4 <!-- 开启springmvc注解 -->
5 <mvc:annotation-driven/>
6
7 <!-- 配置视图解析器 -->
8 <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
9 <!-- 前缀, 后缀 -->
10 <property name="prefix" value="/WEB-INF/jsp/" />
11 <property name="suffix" value=".jsp" />
12 </bean>
```

### 3.controller层的配置:所有的controller都需要放到容器中配置对象.

注解的形式Hollercontroller的写法:实现controller的接口,重写,数据和视图映射到jsp文档.

```
1 public class Hellocontroller implements Controller {
2
```

```

3  @Override
4  public ModelAndView handleRequest(HttpServletRequest Request, HttpServletResponse Response) throws Exception {
5
6  ModelAndView modelAndView = new ModelAndView("/WEB-INF/jsp/hello.jsp");
7
8  return modelAndView;
9  }
10 }

```

1 hicontroller文件中:

```

2 @Controller
3 @RequestMapping("/user")
4 public class hiController {
5
6  //转发,可以访问WEB-INF
7  @RequestMapping("/hi.do")
8  public ModelAndView sayhi(){
9  ModelAndView modelAndView = new ModelAndView("/WEB-INF/jsp/hi.jsp");
10 return modelAndView;
11 }
12
13 //重定向:不能访问WEB-INF
14 @RequestMapping("/world.do")
15 public ModelAndView sayhi1(){
16 ModelAndView modelAndView = new ModelAndView("redirect:/index.jsp");
17 return modelAndView;
18 }
19 也可以在springmvc-config.xml文件中配置
20 // 前缀,后缀
21 @RequestMapping("/ll.do")
22 public ModelAndView sayhi2(){
23 ModelAndView modelAndView = new ModelAndView("hi");
24 return modelAndView;

```

```

25  }
26
27  //post.get都支持
28  //@PostMapping("/l11.do")
29  @GetMapping("/l11.do")
30  public ModelAndView sayhi3(){
31      ModelAndView modelAndView = new ModelAndView("hi");
32      return modelAndView;
33  }
34  //简化
35  @GetMapping("/l111.do")
36  public String sayhi4(){
37      return "hi";
38  }
39
40  }

```

```

1  Dogcontroller文件中:
2  @Controller
3  @RequestMapping("/dog")
4  public class DogController {
5      @RequestMapping("/form.do")
6      public String getParam(HttpServletRequest request) throws UnsupportedEncodingException {
7          request.setCharacterEncoding("utf-8");
8          String name = request.getParameter("name");
9          String age = request.getParameter("age");
10         System.out.println("age = " + age);
11         System.out.println("name = " + name);
12         return "dog";
13     }
14
15     @RequestMapping("/form1.do")
16     public String getParam1(@RequestParam("name") String name1, in
t age) {

```

```

17 System.out.println("age = " + age);
18 System.out.println("name = " + name1);
19
20 Dog dog=new Dog();
21 dog.setName(name1);
22 dog.setAge(age);
23 return "dog";
24 }
25
26 @RequestMapping("/form2.do")
27 public String getParam2(Dog dog) {
28 System.out.println(dog);
29 return "dog";
30 }
31
32
33 }

```

1 Girlcontroller文件中:

```

2 @Controller
3 @RequestMapping("/girl")
4 public class GirlController {
5     @RequestMapping("/send1.do")
6     public String send(HttpServletRequest request){
7         request.setAttribute("a","123");
8         return "girl";
9     }
10    @RequestMapping("/send2.do")
11    public String send1(Model model){
12        // model的数据会存放到request中
13        model.addAttribute("b","321");
14        return "girl";
15    }
16    @RequestMapping("/send3.do")
17    public void send2(HttpServletResponse response) throws IOException {

```

```

18 Dog dog=new Dog();
19 dog.setAge(12);
20 dog.setName("小黄");
21
22 response.setContentType("application/json;charset=utf-8");
23
24 String jsonString = JSON.toJSONString(dog);
25 response.getWriter().write(jsonString);
26 }
27 //返回的内容
28 //返回格式
29 //转成json数据,架包jackson
30 @ResponseBody
31 @RequestMapping(value = "/send4.do",produces = "application/json;charset=utf-8")
32 public Dog send3(HttpServletResponse response) {
33     Dog dog=new Dog();
34     dog.setAge(12);
35     dog.setName("小黄");
36     return dog;
37 }
38 }

```

1 在jsp文件中:

```

2 ${requestScope.a}
3 ${requestScope.b}

```

3.post请求会有乱码:在web.xml文件中设置.

```

1 <filter>
2 <filter-name>characterEncodingFilter</filter-name>
3 <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
4 <init-param>
5 <param-name>encoding</param-name>
6 <param-value>utf-8</param-value>
7 </init-param>
8 </filter>

```

```
9 <filter-mapping>
10   <filter-name>characterEncodingFilter</filter-name>
11   <url-pattern>/*</url-pattern>
12 </filter-mapping>
```