

新车险系统学习第2天

投保单系统中的edit页面

@RequestMapping和@RequestParam

@RequestMapping

- **@RequestMapping** 是 Spring Web 应用程序中最常被用到的注解之一。这个注解会将 HTTP 请求映射到 MVC 和 REST 控制器的处理方法上。
- 要配置 Web 请求的映射，就需要你用上 @RequestMapping 注解。
- @RequestMapping 注解可以在控制器类的级别和/或其中的方法的级别上使用。在类的级别上的注解会将一个特定请求或者请求模式映射到一个控制器之上。之后你还可以另外添加方法级别的注解来进一步指定到处理方法的映射关系。
- 你可以将多个请求映射到一个方法上去，只需要添加一个带有请求路径值列表的 @RequestMapping 注解就行了。

```
@RestController
@RequestMapping("/home")
public class IndexController {

    @RequestMapping(value = {
        "",
        "/page",
        "page*",
        "view/*,**/msg"
    })
    String indexMultipleMapping() {
        return "Hello from index multiple mapping.";
    }
}
```

- @RequestMapping 支持通配符以及ANT风格的路径。前面这段代码中，如下的这些 URL 都会由 indexMultipleMapping() 来处理：

```
localhost:8080/home
localhost:8080/home/
localhost:8080/home/page
localhost:8080/home/pageabc
localhost:8080/home/view/
localhost:8080/home/view/view
```

带有 @RequestParam 的 @RequestMapping

- @RequestParam 注解配合 @RequestMapping 一起使用，可以将请求的参数同处理方法的参数绑定在一起。
- @RequestParam 注解使用的时候可以有一个值，也可以没有值。这个值指定了需要被映射到处理方法参数的请求参数，代码如下所示：

```

@RestController
@RequestMapping("/home")
public class IndexController {

    @RequestMapping(value = "/id")
    //id 这个请求参数被映射到了 thegetIdByValue() 这个处理方法的参数 personId
    上
    String getIdByValue(@RequestParam("id") String personId) {
        System.out.println("ID is " + personId);
        return "Get ID from query string of URL with value element";
    }

    @RequestMapping(value = "/personId")
    //如果请求参数和处理方法参数的名称一样的话, @RequestParam 注解的 value 这个
    参数就可省掉了
    String getId(@RequestParam String personId) {
        System.out.println("ID is " + personId);
        return "Get ID from query string of URL without value element";
    }

}

```

- @RequestParam 注解的 required 这个参数定义了参数值是否是必须要传的。

```

@RestController
@RequestMapping("/home")
public class IndexController {
    @RequestMapping(value = "/name")
    String getName(@RequestParam(value = "person", required = false) St
    ring personName) {
        return "Required element of request param";
    }
}

```

ViewName的构成

```

//1、机动车, 2、特种车, 3、摩托车, 4、提车险
String viewPrefix = "prpt";
if("3".equals(carType)){
    viewPrefix = "prpt/motor";
}
String newRateComCode = AppConfig.get(PubCons.NewRateComCode);
if(newRateComCode.indexOf(comCode.substring(0,2))> -1){
    viewPrefix = viewPrefix+"/poc";
}
String viewName = viewPrefix+"/PrpTEdit";// 默认为非摩托车
if("3".equals(carType)){
    viewName = viewPrefix+"/PrpTEdit_Motor";
    useAutoprice=false;
}
if("4".equals(carType)||Risk.isDBT(riskCode)){
    // 提车险
    viewName = "prpt/PrpTEdit_DAT";
    tmainVo.setCiRisk(PubCons.NO);
    tmainVo.setBiRisk(PubCons.Y1);
}

```

```

        useAutoprice=false;
    }
    if("3".equals(carType)&& !PubCons.Y1.equals(tmainVo.getBIRisk())){
        tmainVo.setJfeeFlag(PubCons.N0);
    }else{
        tmainVo.setJfeeFlag(PubCons.Y1);
    }
    if(!ObjectUtils.isEmpty(tmainVo.getTmain().getCoinsFlag())){

        if("2,4".indexOf(tmainVo.getTmain().getCoinsFlag())>-1){
            tmainVo.setJfeeFlag(PubCons.N0);
        }
    }
}

```

- carType = 3, 摩托车, prpt/motor/PrpTEdit_Motor
- carType = 4, 提车险, prpt/PrpTEdit_DAT
- 其它, prpt/PrpTEdit

edit的页面结构

- PrpTEdit.jsp为主页面。页面太大, exceeding the 65535 bytes limit 无法编译, 拆分页面。

```

<div class="tabs">
    <!-- 页面太大, exceeding the 65535 bytes limit 无法编译, 拆分页面 --%>
    <jsp:include page="/WEB-INF/view/prpt/poc/PrpTEdit_Edit01.jsp">
</jsp:include>
    <jsp:include page="/WEB-INF/view/prpt/poc/PrpTEdit_Edit02.jsp">
</jsp:include>
    <jsp:include page="/WEB-INF/view/prpt/poc/PrpTEdit_Edit03.jsp">
</jsp:include>
</div>

```

jspf

- jspf扩展名文件可以看作是jsp文件include进其他需要的jsp文件中。
- 网上有的说的jspf (Java Simple Plugin Framework) 是一个插件框架, 集成了很多 IoC 框架的概念在里面。此jspf不是彼jspf。
- jspf里可以直接读取XML。

总结

- 今天继续学习新车险业务管理系统中的投保单系统。主要从昨天没解决的edit方法入手, 经过对 **ModelAndView** 的学习理解, 以及对@RequestMapping和@RequestParam两个注解的资料学习, 我剥离出来方法中关于跳转和传参的部分, 找到了下级页面PrpTEdit系列页面。这一系列页面采用页面拆分和jspf的方法编写, 在寻找页面和阅读时花费了一定时间。我今天主要学习了PrpTEdit一系列页面的jspf引用和跳转部分, 粗略地浏览了一下对应的js代码。
- 仔细阅读学习PrpTEdit系列页面功能的js实现和java实现。

遇到的问题

1. js中跳转地址拼凑字段中有 contextPath 这一变量, 是从何处获得的。
2. 业务信息页面找不到, 找到的Edit_YW页面和实际不匹配。

