

ModelAndView详解

本项目用到的最多的类大概就是ModelAndView了，你基本可以在任何一个Controller里面发现它。为什么我那么喜欢用这个类呢？

先看看它的解释：

使用ModelAndView类用来存储处理完后的结果数据，以及显示该数据的视图。从名字上看ModelAndView中的Model代表模型，View代表视图，这个名字就很好地解释了该类的作用。业务处理器调用模型层处理完用户请求后，把结果数据存储在該类的model属性中，把要返回的视图信息存储在該类的view属性中，然后让该ModelAndView返回该Spring MVC框架。框架通过调用配置文件中定义的视图解析器，对该对象进行解析，最后把结果数据显示在指定的页面上。

原因应该已经了然了，它既能返回视图（html页面），还能携带该html页面所需要的数据，那我只要把html页面连同数据一起返回给前端，我们后端的工作就结束了。

知道了它强大的功能，下面解析一下该如何使用。

首先它是如何返回视图的？

一个ModelAndView只能返回一个html页面，你只需要把html页面的名字给ModelAndView就可以了。

例如我想打开home.html页面: ModelAndView.setViewName("home")

其次它如何携带数据？

它有很多函数，其中最好用的就是addObject(name,object)，把你想要传到前端的object起一个名字，在前端就可以根据你起的名字获得这个object。

它可以携带很多数据，但是注意起名不能重复。

可以参考下面的一段代码：

```
@RequestMapping("lookAll")
public ModelAndView lookAll(HttpServletRequest request,@RequestParam
(value = "page", defaultValue = "0") Integer page,
                                @RequestParam(value = "size", defaultValu
e = "15") Integer size){

    User user = getSessionUser(request);
    page+=1;
    int userId;
    if(user==null)
    {
        userId=0;
    }
    else{
        userId=user.getUserId();
    }
    Page<Topic> topics = topicServiceImp.getAllTopic(page,size);
    List<Topic> topics1 = new ArrayList<>();
    for(Topic topic:topics){
        topic.setUser(userServiceImp.getUserById(topic.getUserId()));
    }
}
```

```
        topic.setUrl("/userReadDetail/"+topic.getTopicId());
        setIsPraise(topic,user);
        topics1.add(topic);
    }
    ModelAndView mv = new ModelAndView();
    mv.setViewName("look2");
    mv.addObject("collects",topics1);
    mv.addObject("userId",userId);
    mv.addObject("type","look");
    mv.addObject("functio",userServiceImp);
    mv.addObject("size",topics1.size());
    return mv;
}
```