链滴

SpringMVC 回顾笔记 (七)----@ModelAttribute

作者：pride

原文链接：https://ld246.com/article/1542291428746

来源网站：链滴

● 概念

1.有@ModelAttribute标记的方法，会在每个目标方法执行之前被SpringMVC调用

在方法定义上使用 @ModelAttribute 注解;

Spring MVC在调用目标处理方法前，会先逐个调用在方法级上标注了@ModelAttribute 的方法。

在方法的入参前使用 @ModelAttribute 注解:

– 可以从隐含对象中获取隐含的模型数据中获取对象，再将请求参数绑定对象中，再传入入参

– 将方法入参对象添加到模型中

2.@MethodAttribute 注解也可以来修饰目标方法 POJO 类型的入参，其 Value 属性值有如下作用:

1).SpringMVC 会使用 value 属性值在 implicitModel 中查找对应的对象，存在则会直接传入到目标方法的入参中

2).SpringMVC 会以 value 为 key ， POJO 类型的对象为 value， 存入到 request 中.

```
    /*
     *@ModelAttribute运行流程
     *1.执行@ModelAttribute 注解修饰的方法: 从数据库中取出对象 ， 把对象放入到Map中，键为:usr
     *2.SpringMVC 从Map中取出User对象，并把表单的请求参数赋予给该 User 对象的对应的属性。
     *3.SpringMVC 把上述对象传入目标方法的参数
     *
     *注意: 在 @MethodAttribute 修饰的方法中，放入到 Map 时的键需要和目标方法入参类型的第个字母小写的字符串一致!
     * (这里指TestMethodAttribute方法中的User类型需要和Map提交的键对应，如果在TestMethoAttribute方法参数中使用了
     * @ModelAttribute("字符串")那么Map中对应的键值就是该字符串)
     * */
    @ModelAttribute
    public void getUser(@RequestParam(value="id",required=false) Integer id , Map<String, Oject> maps){
        if(id!=null){
            User user = new User("xiaoming", "123456", 12, 1);
            System.out.println("从User抽取对象:"+user);
            maps.put("abc", user);
        }
    }

    @RequestMapping("/TestMethodAttribute")
    public String TestMethodAttribute(@ModelAttribute("abc") User user){
        System.out.println("修改:"+user);

        return "success";
    }
```

● 源码分析流程

1.调用@ModelAttribute注解修饰的方法。实际上把@ModelAttribute方法中Map中的数据放在了iplicitModel中

2.解析请求处理器的目标参数，实际上该目标参数来自于WebDataBinder对象的target 属性

1）.创建 WebDataBinder对象:

a.确定objectName属性：若传入的attrName属性值为""，则objectName为类名第一个字母小写。

注意：attrName.若目标方法的POJO属性使用了@ModelAttribute来修饰，则attrName值即为@MdelAttribute的value属性值

b.确定 target 属性：

/>在implicitMode1中查找attrName对应的属性值。若存在，ok

/>若不存在：则验证当前Handler是否使用了@SessionAttributes 进行修饰，若使用了，则尝试从Session中获取attrName所对应的属性值。若session中没有对应的属性值，则抛出了异常。

/> 若 Handler 没有使用@SessionAttributes 进行修饰，或@SessionAttributes 中没有使用value 指定的key和attrName相匹配，则通过反射创建了POJO对象

2）.SpringMVC 把表单的请求参数赋给了WebDataBinder的 target 对应的属性。

3）.*SpringMVC 会把WebDataBinder的attrName和target 给到implicitModel.

近而传到request域对象中。

4）.把WebDataBinder的target作为参数传递给目标方法的入参。

● SpringMVC 确定目标方法 POJO 类型入参的过程

1.确定一个key:

1).若目标方法的 POJO 类型的参数没有使用 @ModelAttribute 作为修饰，则 key 为 POJO 类名第个字母的小写

2).若使用了 @MethodAttribute 来修饰,则 key 为 @MethodAttribute 注解的 value 属性值

2.在 implicitModel 中查找 key 对应的对象，若存在，则作为入参传入

1).若在 @ModelAttribute 标记的方法中在 Map 中保存过，且 key 和 1 确定的key一致，则会获取。

3.若 implicitModel 中不存在 key 对应的对象，则检查当前的 Handler 是否使用 @SessionAttribute 注解修饰，

若使用了该注解，且 @SessionAttributes 注解的 value属性值中包含了 key，则会从 HttpSession 来获取key所对应的 value 值，

若存在则直接传入到目标方法的入参中。若不存在则将抛出异常。

4.若 Handler 没有标识 @SessionAttributes 注解或 @SessionAttributes 注解的 value 值中不包含ky，

则会通过反射来创建 POJO 类型的参数，传入为目标方法的参数

5.SpringMVC 会把 key 和 POJ0 类型的对象保存到 implicitModel 中，进而会保存到 request 中。