@ModelAttribute注解

# @ModelAttribute

## 作用

**@ModelAttribute**可以应用在**方法参数上或方法上**，他的作用主要是**当注解在方法参数上时会将注解的参数对象添加到Model中**；当注解在请求处理方法Action上时会将该方法变成一个非请求处理的方法，但其它Action被调用时会首先调用该方法。

## @ModelAttribute注释一个方法

**被@ModelAttribute注释的方法表示这个方法的目的是增加一个或多个模型(model)属性**。这个方法和被@RequestMapping注释的方法一样也支持@RequestParam参数，但是它不能直接被请求映射。**实际上，控制器中的@ModelAttribute方法是在同一控制器中的@RequestMapping方法被调用之前调用的。**

**被@ModelAttribute注释的方法用于填充model属性**，例如，为下拉菜单填充内容，或检索一个command对象（如，Account），用它来表示一个HTML表单中的数据。

一个控制器可以有任意数量的**@ModelAttribute方法**。所有这些方法都在@RequestMapping方法被调用之前调用。

**有两种类型的@ModelAttribute方法**。一种是：只加入一个属性，用方法的返回类型隐含表示。另一种是：方法接受一个Model类型的参数，这个model可以加入任意多个model属性。

(1)@**ModelAttribute**注释void返回值的方法

复制代码

@Controller

@RequestMapping(value="/test")

public class TestController {

/\*\*

\* 1.@ModelAttribute注释void返回值的方法

\* @param abc

\* @param model

\*/

@ModelAttribute

public void populateModel(@RequestParam String abc, Model model) {

model.addAttribute("attributeName", abc);

}

@RequestMapping(value = "/helloWorld")

public String helloWorld() {

return "test/helloWorld";

}

}

复制代码

这个例子，在获得请求/helloWorld 后，populateModel方法在helloWorld方法之前先被调用，它把请求参数（/helloWorld?abc=text）加入到一个名为attributeName的model属性中，在它执行后helloWorld被调用，返回视图名helloWorld和model已由@ModelAttribute方法生产好了。

这个例子中model属性名称和model属性对象由model.addAttribute()实现，不过前提是要在方法中加入一个Model类型的参数。

## @ModelAttribute注释返回具体类的方法

复制代码

/\*\*

\* 2.@ModelAttribute注释返回具体类的方法

\* @param id

\* @return

\*/

@ModelAttribute

public User getUserInfo(String id){

if(id!=null && !id.equals("")){

return userService.getUserInfo(id);

}

return null;

}

复制代码

这种情况，model属性的名称没有指定，它由返回类型隐含表示，如这个方法返回User类型，那么这个model属性的名称是user。

这个例子中model属性名称有返回对象类型隐含表示，model属性对象就是方法的返回值。它无须要特定的参数。

## @ModelAttribute和@RequestMapping同时注释一个方法

复制代码

@Controller

@RequestMapping(value="/test")

public class TestController {

@RequestMapping(value = "/helloWorld")

@ModelAttribute("attributeName")

public String helloWorld() {

return "hi";

}

}

复制代码

这时这个方法的返回值并不是表示一个视图名称，而是model属性的值，视图名称由RequestToViewNameTranslator根据请求"/helloWorld"转换为helloWorld。Model属性名称由@ModelAttribute(value=””)指定，相当于在request中封装了key=attributeName，value=hi。

4.2 @ModelAttribute注释一个方法的参数

@ModelAttribute注释方法的一个参数表示应从模型model中取得。若在model中未找到，那么这个参数将先被实例化后加入到model中。若在model中找到，则请求参数名称和model属性字段若相匹配就会自动填充。这个机制对于表单提交数据绑定到对象属性上很有效。

当@ModelAttribute注解用于方法参数时，它有了双重功能，即“存/取”。首先，它从模型中取出数据并赋予对应的参数，如果模型中尚不存在，则实例化一个，并存放于模型中；其次，一旦模型中已存在此数据对象，接下来一个很重要的步骤便是将请求参数绑定到此对象上（请求参数名映射对象属性名），这是Spring MVC提供的一个非常便利的机制--数据绑定。

@RequestMapping(value = "/login.htm", method = RequestMethod.GET)

public String doLogin(@ModelAttribute("baseMember") BaseMember member) {

member.setLoginName("loginName");

return "home";

}

上述代码中，如果模型中尚不存在键名为“baseMember”的数据，则首先会调用BaseMember类的默认构造器创建一个对象，如果不存在默认构造器会抛出异常。因此，给实体类提供一个默认构造器是一个好的编程习惯。当请求路径的请求参数或提交的表单与BaseMember的属性名匹配时，将自动将其值绑定到baseMember对象中，非常的便利！这可能是我们使用@ModelAttribute最主要的原因之一。比如：请求路径为http://localhost:8080/spring-web/login.htm?loginName=myLoginName，baseMember对象中的loginName属性的值将被设置为myLoginName。

4.3 @ModelAttribute注解的使用场景

当@ModelAttribute注解用于方法时，与其处于同一个处理类的所有请求方法执行前都会执行一次此方法，这可能并不是我们想要的，因此，我们使用更多的是将其应用在请求方法的参数上，而它的一部分功能与@RequestParam注解是一致的，只不过@RequestParam用于绑定单个参数值，而@ModelAttribute注解可以绑定所有名称匹配的，此外它自动将绑定后的数据添加到模型中，无形中也给我们提供了便利，这也可能是它命名为ModelAttribute的原因。