

Spring MVC Controller 参数的匹配机制

作者: [kevin](#)

原文链接: <https://ld246.com/article/1397034100442>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Controller 如何将参数注入 (或匹配) 对象?

>举一个具体的栗子：表单提交时，如何将众多参数自动注入到 Controller 的对象参数中，而不使用 @RequestParam 注解强制标注参数来进行注入。

这个问题我最初的理解是：**表单中元素的 name 值与持久层对象的属性名一致，Controller 即可自扫描参数对象的属性，如果 name 值一致，则注入到参数对象中。**

下面进行实验：

基于 MVC 的思想，程序中必定有持久层，无论是叫做 JavaBean，还是叫做 POJO，他们都应当尽与数据库中表的结构保持一致。并且，这些属性都有 Setter 和 Getter 方法。

####实验一：

下面是基于下划线分割命名形式生成的 User 类（为什么要用这样奇怪的形式呢？为了和数据库中结构对应）：

```
public class User {  
  
    private int user_id;  
    private String user_name;  
    private String user_pass;  
  
    public void setUser_Id(int user_id) {  
        this.user_id = user_id;  
    }  
  
    public void setUser_Name(String user_name) {  
        this.user_name = user_name;  
    }  
  
    public void setUser_Pass(String user_pass) {  
        this.user_pass = user_pass;  
    }  
  
    // getter 省略  
  
}
```

紧接着，需要一个有对应 name 值的前台表单：

```
<h2>请登录</h2>
```

```
登录
```

最后，编写 Controller 类控制请求：

```
@Controller  
public class SignController {  
    @RequestMapping(value = "/signin", method = RequestMethod.POST)  
    public ModelAndView Login(User user) {
```

```

ModelAndView mv = new ModelAndView();
String mvName = "index";
if (user != null) {
    if (user.getUser_Name() != null && user.getUser_Pass() != null) {
        System.out.println(user.getUserName() + " " + user.getUserPassword())

    } else {
        System.out.println("username and password == null");
    }
} else {
    System.out.println("user == null");
}
}
}
}

```

编写完代码，要开始验证啦。

悲剧的是，打印出来的居然是 username and password == null!

debugging 进去，Login 方法的参数 User 居然是一个空壳：User 本身不为空，而 User 中全部的性都没有值，也就是说，没有属性名自动注入到对象中!!!

问题出在哪里？难道需要显式的写明属性注解才能识别？

将 Login 方法的参数改为 @RequestAttribute("ser")，问题依旧。

等等，是不是下划线分割命名形式中的下划线 _ 阻碍了注入？

来验证一下猜想：将程序稍作修改，替换为驼峰形式。

####实验二

```

public class User {
    private int userId;
    private String userName;
    private String userPassw;

    public void setUserId(int userId) {
        this.userId = userId;
    }

    public void setUserName(String userName) {
        this.userName = userName;
    }

    public void setUserPass(String userPass) {
        this.userPass = userPass;
    }

    // getter 省略
}

```

<h2>请登录</h2>

登录

果然自动注入了!!!

什么? 强大的 Spring 会傻傻的被下划线击败吗? 这似乎与“表单中元素的 name 值与持久层对象的姓名一致, Controller 即可自动扫描参数对象的属性, 如果 name 值一致, 则注入到参数对象中”这的表述相悖, 也与第二次实验相悖。

仔细观察实验一和实验二, 两者最大的区别, 除了使用下划线分割单词以外, 实验二中属性的第二个之后的单词首字母大写, Setter 无论是否有下划线, 均为第二个及之后的单词首字母大写, 而 Setter 自动把第一个单词的首字母做了大写的处理。

实验继续, 将实验一表单中的name值由 **user_name** 变为 **user_Name** 。

####实验三

<h2>请登录</h2>

登录

实验三同实验二结果相同, 可以自动注入。

总结一下实验: 可以发现属性值均为 private, 所以应当依靠 Setter 注入, 表单中 user_name 这样的属性, 在 User 类中没有 Setter 对应, 应为 Spring 会找 setUser_ame() 方法注入啊亲!

写了这么多, 过程有些繁琐, 结论很简单。再次开头的问题:

>Controller 如何将参数注入 (或匹配) 对象?

>表单元素的name值必须与 Setter 名保证一致, 首字母除外。