

协作翻译 > 服务器端开发 > 正文

Spring MVC Flash Attribute 的讲解与使用示例

👍 顶

15

已翻译 100%

0595 投递于 2014/04/05 07:47 (共 5 段, 翻译完成于 04-10) 阅读 23087

☆ 收藏 160 💬 评论 24

Spring

英文原文: Spring MVC Flash Attribute Tutorial With Example

参与翻译 (3人): [戴仓薯](#), [Idiot_s_Sky](#), [LeoXu](#)

🖨 打印

仅中文

中英文对照

仅英文

Spring MVC 3.1版本加了一个很有用的特性, Flash属性, 它能解决一个长久以来缺少解决的问题, 一个POST/Redirect/GET模式问题。

正常的MVC Web应用程序在每次提交都会POST数据到服务器。一个正常的Controller (被注解@Controller标记)从请求获取数据和处理它 (保存或更新数据库)。一旦操作成功, 用户就会被带到 (forward) 一个操作成功的页面。传统上来说, 这样的POST/Forward/GET模式, 有时候会导致多次提交问题. 例如用户按F5刷新页面, 这时同样的数据会再提交一次。

为了解决这问题, POST/Redirect/GET 模式被用在MVC应用程序上. 一旦用户表单被提交成功, 我们重定向 (Redirect) 请求到另一个成功页面。这样能够令浏览器创建新的GET请求和加载新页面。这样用户按下F5, 是直接GET请求而不是再提交一次表单。

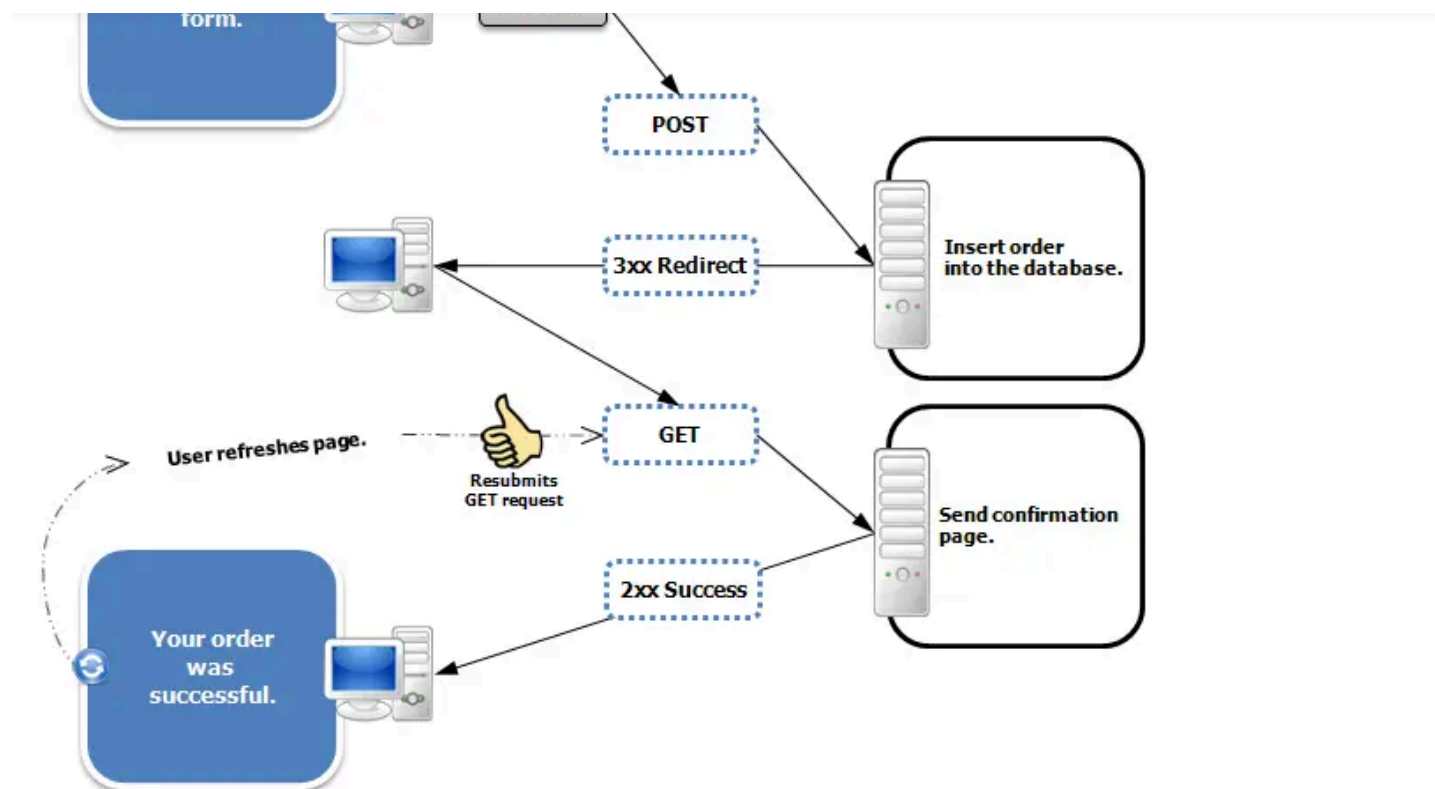


Image credit: Wikipedia



Idiot_s_Sky

翻译于 2014/04/09
17:31



顶

6

虽然这一方法看起来很完美，并且解决了表单多次提交的问题，但是它又引入了一个获取请求参数和属性的难题。通常当我们生成一次http重定向请求的时候，被存储到请求数据会丢失，使得下一次GET请求不可能访问到这次请求中的一些有用的信息。

Flash attributes 的到来就是为了处理这一情况。Flash attributes 为一个请求存储意图为另外一个请求所使用的属性提供了一条途径。Flash attributes 在对请求的重定向生效之前被临时存储（通常是在 session)中，并且在重定向之后被立即移除。



为了这样做, Flash 特性使用了两个集合. FlashMap 被用来管理 flash attributes 而 FlashMapManager 则被用来存储, 获取和管理 FlashMap 实体.

对于每一次请求一个 “input” flash map 会被创建, 来存储来自任何之前请求的 flash attribute 还有一个 “output” flash map 会被创建, 来存储任何我们存储在这个请求中的, 之后的请求参数.

**LeoXu**翻译于 2014/04/10
08:08

顶

3

使用

要想在你的 Spring MVC 应用中使用 Flash attribute, 要用 **3.1 版本或以上**. 并且要在 spring-servlet.xml 文件中加入 mvc:annotation-driven.

```
<mvc:annotation-driven />
```

这些都完成之后, Flash attribute 就会自动设为“开启”, 以供使用了. 只需在你的 Spring controller 方法中加入 RedirectAttributes redirectAttributes.

```
import org.springframework.web.servlet.mvc.support.RedirectAttributes;
//...

@RequestMapping(value="addcustomer", method=RequestMethod.POST)
public String addCustomer(@ModelAttribute("customer") Customer customer,
                           final RedirectAttributes redirectAttributes) {
    //...
    redirectAttributes.addFlashAttribute("message", "Successfully added..");
    //...
```

addFlashAttribute 方法会自动向 **output** flash map 中添加给定的参数，并将它传递给后续的请求。

我们来看看一个使用 Flash attribute 来完成 POST/Redirect/GET 并传递一些信息的完整实例吧。

Flash Attribute 实例

下面的应用向用户显示一个表单。当用户填完数据，并提交表单之后，页面会重定向到另一个显示成功信息的页面。在这个重定向的新页面中，会显示用户刚才输入的信息。



戴仓薯

翻译于 2014/04/09
15:52

👍 顶

4

第1步: 需要的 JAR 和项目结构

如果你用 Maven 来做依赖管理，用下面的 dependencies 来添加 Spring 3.1 MVC 的支持。

```
<dependencies>
  <!-- Spring 3.1 MVC -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>3.1.2.RELEASE</version>
  </dependency>
  <!-- JSTL for c: tag -->
  <dependency>
    <groupId>jstl</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
  </dependency>
</dependencies>
```

或者，你可以下载以下 JAR 文件，然后把它们放在 **/WEB-INF/lib** 文件夹下。



资讯

软件

博客

专区

问答

活动

工具

培训



第2步: Spring 配置

要为 web 项目添加 Spring 支持，需要在 web.xml 中添加 DispatcherServlet。

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  id="WebApp_ID" version="2.5">

  <display-name>Spring MVC Flash attribute example</display-name>
  <servlet>
    <servlet-name>spring</servlet-name>
    <servlet-class>
      org.springframework.web.servlet.DispatcherServlet
    </servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>default</servlet-name>
    <url-pattern>/index.html</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>spring</servlet-name>
    <url-pattern>*.html</url-pattern>
  </servlet-mapping>
</web-app>
```

然后，spring-servlet 使用 mvc:annotation-driven 来支持 mvc，并且会扫描项目中的 context:component-scan 标签。

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:mvc="http://www.springframework.org/schema/mvc"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans
                           http://www.springframework.org/schema/context http://www.springframework.org/schema/context
                           http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc" >

    <bean id="jspViewResolver"
          class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="viewClass"
                  value="org.springframework.web.servlet.view.JstlView" />
        <property name="prefix" value="/WEB-INF/jsp/" />
        <property name="suffix" value=".jsp" />
    </bean>

    <context:component-scan base-package="net.viralpatel.controller" />
    <mvc:annotation-driven />

</beans>
```

第3步: Spring Controller – RedirectAttributes

Controller 的代码使用 Customer.java 对象作为 bean 来保存客户信息。

Customer.java

```
package net.viralpatel.spring;

public class Customer {
    private String firstname;
    private String lastname;
    private int age;
    private String email;

    //getter, setter methods
}
```

CustomerController 类有3个方法。showForm 方法对应 URL **/form**，用来显示 Add New Customer 表单。addCustomer 方法对应 URL **/addcustomer**，用来处理 POST 请求。

CustomerController.java

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.mvc.support.RedirectAttributes;

@Controller
public class CustomerController {

    @RequestMapping(value="showform", method=RequestMethod.GET)
    public String showForm(@ModelAttribute("customer") Customer customer) {
        return "add_customer";
    }

    @RequestMapping(value="addcustomer", method=RequestMethod.POST)
    public String addCustomer(@ModelAttribute("customer") Customer customer,
        final RedirectAttributes redirectAttributes) {

        redirectAttributes.addFlashAttribute("customer", customer);
        redirectAttributes.addFlashAttribute("message","Added successfully.");

        return "redirect:showcustomer.html";
    }

    @RequestMapping(value="showcustomer", method=RequestMethod.GET)
    public String showCustomer(@ModelAttribute("customer") Customer customer) {
        System.out.println("cust:" + customer.getFirstname());
        return "show_customer";
    }
}
```

注意我们在 addCustomer 方法中是如何使用 redirectAttributes 参数来添加 flash attribute 的。并且，我们是用 addFlashAttribute 方法来设置新的参数为 flash attribute。

**戴仓薯**翻译于 2014/04/09
16:09

👍 顶

4

第4步: View 层

```
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<html>
<body>
    <h1>Add New Customer</h1>
    <form:form action="addcustomer.html" method="post" commandName="customer">
    <table>
        <tr>
            <td><form:label path="firstname">Firstname</form:label></td>
            <td><form:input path="firstname" /> </td>
        </tr>
        <tr>
            <td><form:label path="lastname">Lastname</form:label></td>
            <td><form:input path="lastname" /> </td>
        </tr>
        <tr>
            <td><form:label path="age">Age</form:label></td>
            <td><form:input path="age" /> </td>
        </tr>
        <tr>
            <td><form:label path="email">Email</form:label>
            <td><form:input path="email" /> </td>
        </tr>
        <tr>
            <td colspan="2"><input type="submit" value="Add Customer" />
        </td>
        </tr>
    </table>
    </form:form>
</body>
</html>
```

show_customer.jsp 简单地显示客户的名和姓，以及用 flash attributes 设置的成功信息。

show_customer.jsp

```
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<html>
<body>
<h1>${message}</h1>
    ${customer.lastname}, ${customer.firstname} added successfully..
</body>
</html>
```

Demo:



localhost:8080/SpringMVC_F x

localhost:8080/SpringMVC_Flash_Attribute_Maven_example/form.html

Add New Customer

Firstname

Lastname

Age

Email



localhost:8080/SpringMVC_F x

localhost:8080/SpringMVC_Flash_Attribute_Maven_example/showcustomer.html

Success Added successfully.

Patel, Viral added successfully...

下载源代码

SpringMVC_Flash_Attribute_example.zip (3.5 MB)



戴仓薯

翻译于 2014/04/09
15:56

👍 顶

2

本文中的所有译文仅用于学习和交流目的，转载请务必注明文章译者、出处、和本文链接。
我们的翻译工作遵照 [CC 协议](#)，如果我们的工作有侵犯到您的权益，请及时联系我们。

**qwiy5** 2017/07/13 09:27

不错

 举报**徐林** 2014/04/14 09:59`org.springframework.web.servlet.support.SessionFlashMapManager`

如果系统没有配置，默认保持在Session中。

在单机部署中，这个没有问题；但在集群环境中，数据可能会串掉。

 举报**沉城** 2014/04/13 22:26**引用来自“小杨阿哥哥”的评论**

压根就不用这种方式，重定向的话ModelAndView可以接受map，如果是redirectView也就将参数自动转换到URL中

引用来自“xmut”的评论

什么叫“压根”，佩服啊！先不说是否加密之类的东西，凭什么我们要跟你一样都把参数都放在URL中！

没错，有些人就是比较自以为是

 举报**XzhiF** 2014/04/13 20:36

赞

 举报**james_you** 2014/04/12 23:19

提交后，点击浏览器返回后，再提交呢？一般这种事情，都是用token做嘛

 举报**pauli** 2014/04/12 20:15里面好东西不少了 <https://code.google.com/p/viralpatel-net-tutorials/downloads/list> 举报**vio** 2014/04/12 19:54

个人认为用cookies比session存储这些数据好一点

 举报**leon_lau** 2014/04/12 19:00

源码

```
synchronized (writeLock) {  
    List<FlashMap> allMaps = retrieveFlashMaps(request);  
    allMaps = (allMaps == null) ? new CopyOnWriteArrayList<FlashMap>() : allMaps;  
    allMaps.add(flashMap);  
    updateFlashMaps(allMaps, request, response);  
}
```

```
protected void updateFlashMaps(List<FlashMap> flashMaps, HttpServletRequest request,
    HttpServletResponse response) {
    request.getSession().setAttribute(FLASH_MAPS_SESSION_ATTRIBUTE, flashMaps);
}
```

🚩 举报




unnamed

2014/04/12 15:45

spring mvc 也加了flash 啊。。

🚩 举报



Xiao_f

2014/04/12 14:58

如果存入session redirect后即删除，再刷新不就找不到了，感觉放到url中才是解决之道

🚩 举报

在这里发表你对此文的观点

 插入表情

插入软件

0/1000

 发表评论

OSCHINA 社区

关于我们

帮助中心

合作伙伴

Open API

在线工具

Gitee.com

企业研发管理

CopyCat-代码克隆检测

实用在线工具

攻略

项目运营

Awesome 软件（持续更新中）

QQ群



229767317

公众号



视频号



[资讯](#)

[软件](#)

[博客](#)

[专区](#)

[问答](#)

[活动](#)

[工具](#)

[培训](#)

©OSCHINA(OSChina.NET) [工信部](#) [开源软件推进联盟](#) [指定官方社区](#) [社区规范](#)

深圳市奥思网络科技有限公司版权所有 [粤ICP备12009483号](#)