**Spring 中的自定义事件**

**Spring 中的自定义事件**

编写和发布自己的自定义事件有许多步骤。按照在这一章给出的说明来编写，发布和处理自定义 Spring 事件。

|  |  |
| --- | --- |
| **步骤** | **描述** |
| 1 | 创建一个名称为 SpringExample 的项目，并且在创建项目的 **src** 文件夹中创建一个包 com.tutorialspoint。 |
| 2 | 使用 Add External JARs 选项，添加所需的 Spring 库，解释见 Spring Hello World Example 章节。 |
| 3 | 通过扩展 **ApplicationEvent**,创建一个事件类 CustomEvent。这个类必须定义一个默认的构造函数，它应该从 ApplicationEvent 类中继承的构造函数。 |
| 4 | 一旦定义事件类，你可以从任何类中发布它，假定 EventClassPublisher 实现了 ApplicationEventPublisherAware。你还需要在 XML 配置文件中声明这个类作为一个 bean，之所以容器可以识别 bean 作为事件发布者，是因为它实现了 ApplicationEventPublisherAware 接口。 |
| 5 | 发布的事件可以在一个类中被处理，假定 EventClassHandler 实现了 ApplicationListener 接口，而且实现了自定义事件的 onApplicationEvent 方法。 |
| 6 | 在 **src** 文件夹中创建 bean 的配置文件 Beans.xml 和 MainApp 类，它可以作为一个 Spring 应用程序来运行。 |
| 7 | 最后一步是创建的所有 Java 文件和 Bean 配置文件的内容，并运行应用程序，解释如下所示。 |

这个是 **CustomEvent.java** 文件的内容：

package com.tutorialspoint;

import org.springframework.context.ApplicationEvent;

public class CustomEvent extends ApplicationEvent{

public CustomEvent(Object source) {

super(source);

}

public String toString(){

return "My Custom Event";

}

}

下面是 **CustomEventPublisher.java** 文件的内容：

package com.tutorialspoint;

import org.springframework.context.ApplicationEventPublisher;

import org.springframework.context.ApplicationEventPublisherAware;

public class CustomEventPublisher

implements ApplicationEventPublisherAware {

private ApplicationEventPublisher publisher;

public void setApplicationEventPublisher

(ApplicationEventPublisher publisher){

this.publisher = publisher;

}

public void publish() {

CustomEvent ce = new CustomEvent(this);

publisher.publishEvent(ce);

}

}

下面是 **CustomEventHandler.java** 文件的内容：

package com.tutorialspoint;

import org.springframework.context.ApplicationListener;

public class CustomEventHandler

implements ApplicationListener<CustomEvent>{

public void onApplicationEvent(CustomEvent event) {

System.out.println(event.toString());

}

}

下面是 **MainApp.java** 文件的内容：

package com.tutorialspoint;

import org.springframework.context.ConfigurableApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

public static void main(String[] args) {

ConfigurableApplicationContext context =

new ClassPathXmlApplicationContext("Beans.xml");

CustomEventPublisher cvp =

(CustomEventPublisher) context.getBean("customEventPublisher");

cvp.publish();

cvp.publish();

}

}

下面是配置文件 **Beans.xml**：

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

<bean id="customEventHandler"

class="com.tutorialspoint.CustomEventHandler"/>

<bean id="customEventPublisher"

class="com.tutorialspoint.CustomEventPublisher"/>

</beans>

一旦你完成了创建源和 bean 的配置文件后，我们就可以运行该应用程序。如果你的应用程序一切都正常，将输出以下信息：

My Custom Event

My Custom Event