

## Create a Java program to demonstrate default and custom event handling

### Pom.xml:

```
project
xmlns="http://maven.apache.org/POM/4.0.0"xmlns:xsi="h
ttp://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>SpringSetup</groupId>
  <artifactId>SpringSetup</artifactId>
  <packaging>war</packaging>
  <version>0.0.1-SNAPSHOT</version>
  <name>SpringSetupMavenWebapp</name>
  <url>http://maven.apache.org</url>

  <properties>

<org.springframework.version>5.1.6.RELEASE</org.springframework.version>
  </properties>

  <dependencies>
  <dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>3.8.1</version>
  <scope>test</scope>
  </dependency>

  <dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
  <version>${org.springframework.version}</version>
  </dependency>
  <dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-expression</artifactId>
  <version>${org.springframework.version}</version>
  </dependency>
  <dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-beans</artifactId>
  <version>${org.springframework.version}</version>
  </dependency>
  <dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
  <version>${org.springframework.version}</version>
  </dependency>
  <dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context-support</artifactId>
  <version>${org.springframework.version}</version>
  </dependency>

  <dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-jdbc</artifactId>
```

```

<version>${org.springframework.version}</version>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-orm</artifactId>
<version>${org.springframework.version}</version>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-web</artifactId>
<version>${org.springframework.version}</version>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-webmvc</artifactId>
<version>${org.springframework.version}</version>
</dependency>

<dependency>
<groupId>javassist</groupId>
<artifactId>javassist</artifactId>
<version>3.12.1.GA</version>
</dependency>

<dependency>
<groupId>>taglibs</groupId>
<artifactId>standard</artifactId>
<version>1.1.2</version>
<scope>runtime</scope>
</dependency>

<dependency>
<groupId>commons-dbcp</groupId>
<artifactId>commons-dbcp</artifactId>
<version>1.4</version>
</dependency>

<dependency>
<groupId>javax.servlet</groupId>
<artifactId>jstl</artifactId>
<version>1.2</version>
</dependency>
<dependency>
<groupId>javax.servlet</groupId>
<artifactId>jstl</artifactId>
<version>1.2</version>
</dependency>
<dependency>
<groupId>javax.persistence</groupId>
<artifactId>persistence-api</artifactId>
<version>1.0.2</version>
<scope>provided</scope>
</dependency>
</dependencies>
<build>
<finalName>SpringSetup</finalName>
</build>
</project>

```

### **STARTEVENTHANDLER. JAVA:**

```
package com.ecommerce;  
import org.springframework.context.ApplicationListener;  
import org.springframework.context.event.ContextStartedEvent;  
public class StartEventHandler  
implements ApplicationListener<ContextStartedEvent>{  
    public void onApplicationEvent (ContextStartedEvent event) {  
        System.out.println("ContextStartedEventReceived");  
    }  
}
```

### **STOPEVENTHANDLER:**

```
package com.ecommerce.beans;  
import org.springframework.context.ApplicationListener;  
import org.springframework.context.event.ContextStoppedEvent;  
public class StopEventHandler  
implements ApplicationListener<ContextStoppedEvent>{  
    public void onApplicationEvent (ContextStoppedEvent event) {  
        System.out.println("ContextStoppedEventReceived");  
    }  
}
```

### **CUSTOMEVENT:**

```
package com.ecommerce.beans;  
import org.springframework.context.ApplicationEvent;  
public class CustomEvent extends ApplicationEvent{  
  
    public CustomEvent (Object source) {  
        super (source);  
    }  
    public String toString() {  
        return "This is a custom event";  
    }  
}
```

### **CUSTOMEVENTPUBLISHER:**

```
package com.ecommerce.beans;  
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);  
    }  
}
```

### CUSTOMEVENTLIST:

```
package com.ecommerce;  
import org.springframework.context.ApplicationListener;  
public class CustomEventListener  
implements ApplicationListener<CustomEvent> {  
  
    public void onApplicationEvent (CustomEvent event) {  
        System.out.println(event.toString());  
    }  
}
```

### MAINCONTROLLER:

```
package com.ecommerce.controller;  
import org.apache.log4j.Logger;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.ModelMap;  
import org.springframework.validation.BindingResult;  
import org.springframework.web.bind.annotation.ModelAttribute;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.context.ApplicationContext;  
import  
org.springframework.context.support.ClassPathXmlApplicationContext;  
@Controller  
public class Main Controller{  
  
    @RequestMapping (value="/customevent",  
  
        method= RequestMethod.GET)  
    public String customEvent (ModelMap map)  
    {  
        String confFile="main-servlet.xml";  
        ApplicationContext context = new  
        Class PathXmlApplicationContext (confFile);  
        CustomEventPublisher cvp =  
        (CustomEventPublisher)  
        context.getBean ("customEventPublisher");  
  
        cvp.publish();  
        cvp.publish();  
        return "customEvent";  
    }  
}
```