

Default and Custom Event Handling

Step 1: Creating pom.xml for including the required components

- In the Project Explorer, expand SpringSetup and double click on pom.xml
- Add the following entries:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://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>SpringSetup Maven Webapp</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>

```

Step 2: Creating a bean class StartEventHandler

```
package com.ecommerce.beans;
```

```
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("ContextStartedEvent Received");
        }
    }
}

```

Step 3: Creating a bean class 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("ContextStoppedEvent Received");
    }
}

```

Step 4: Creating a bean class 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";
    }
}

```

```
}  
}
```

Step 5: Creating a bean class 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);  
    }  
}
```

Step 6: Creating a bean class CustomEventListener

```
package com.ecommerce;
```

```
import org.springframework.context.ApplicationListener;
```

```
public class CustomEventListener implements
```

```

ApplicationListener<CustomEvent> {

    public void onApplicationEvent(CustomEvent event) {
        System.out.println(event.toString());
    }
}

```

Step 7: Creating a Controller class 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.ClassPathXmlApplicationConte
xt;

@Controller
public class MainController {

```

```

    @RequestMapping(value = "/customevent", method =
RequestMethod.GET)
    public String customEvent(ModelMap map)
    {
        String confFile = "main-servlet.xml";
        ApplicationContext context = new
ClassPathXmlApplicationContext(confFile);
        CustomEventPublisher cvp =
            (CustomEventPublisher)
context.getBean("customEventPublisher");

        cvp.publish();
        cvp.publish();
        return "customEvent";
    }
}

```

Step 8: Creating the dispatcher servlet main-servlet.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"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:jee="http://www.springframework.org/schema/jee"
    xmlns:lang="http://www.springframework.org/schema/lang"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xmlns:util="http://www.springframework.org/schema/util"

```

```

xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-2.5.xsd

```


<http://www.springframework.org/schema/context>
<http://www.springframework.org/schema/context/spring-context-2.5.xsd>
<http://www.springframework.org/schema/tx>
[http://www.springframework.org/schema/tx/spring-tx-2.5.xsd">](http://www.springframework.org/schema/tx/spring-tx-2.5.xsd)

```
<context:annotation-config />
<context:component-scan base-
package="com.ecommerce.controller" />
<bean id="jspViewResolver"
class="org.springframework.web.servlet.view.InternalResourceViewR
esolver">
    <property name="viewClass"
        value="org.springframework.web.servlet.view.JstlView">
</property>
    <property name="prefix" value="/WEB-INF/view/"></property>
    <property name="suffix" value=".jsp"></property>
</bean>
<bean id="messageSource"
```

```
class="org.springframework.context.support.ReloadableResourceBu
ndleMessageSource">
    <property name="basename" value="classpath:messages">
</property>
    <property name="defaultEncoding" value="UTF-8"></property>
</bean>
```

```
<bean id = "startEventHandler" class =
"com.ecommerce.beans.StartEventHandler"/>
<bean id = "stopEventHandler" class =
"com.ecommerce.beans.StopEventHandler"/>
```

```
<bean id = "customEventListener" class =
"com.ecommerce.beans.CustomEventListener"/>
<bean id = "customEventPublisher" class =
"com.ecommerce.beans.CustomEventPublisher"/>
```

```
<tx:annotation-driven />

</beans>
```

Step 9: Creating a view index.jsp

```
<html>
<body>
<h2>Spring Application</h2>

<a href="customevent">Custom Event Handling</a>
</body>
</html>
```

Step 10: Creating a view customEvent.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Custom Event</title>
</head>
<body>
    Custom Event has been generated in the Java Console.
</body>
</html>
```

Step 11: Configuring web.xml

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

```

<web-app xmlns="http://java.sun.com/xml/ns/javaee"
version="2.5"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd">

  <display-name>Archetype Created Web Application</display-
name>

  <welcome-file-list>
    <welcome-file>/WEB-INF/view/index.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <servlet-name>main</servlet-name>
    <servlet-class>
      org.springframework.web.servlet.DispatcherServlet
    </servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>main</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/main-servlet.xml</param-value>
  </context-param>
  <listener>
    <listener-class>
      org.springframework.web.context.ContextLoaderListener</listener-
class>
    </listener>
  </web-app>

```