



V.B.G.

Virtual Business Gate

HOW TO di una interfaccia custom alle funzionalità di VBG2

V 2.10

Ed 1.0 01/12/2011



VBG è un sistema software per la gestione delle segnalazioni certificate di inizio attività e dei procedimenti di autorizzazione e concessione on-line.

Il sistema è stato realizzato da INIT srl su incarico del Consorzio SIR Umbria nell'ambito del progetto Ri_Umbria, il progetto di riuso della Regione Umbria.

Il sistema VBG è a disposizione in riuso a tutte le Pubbliche Amministrazioni che ne fanno richiesta al Consorzio SIR Umbria.

Maggiori informazioni sono reperibili all'indirizzo www.sir.umbria.it/riumbria



SOMMARIO

| | | |
|---|-------------------------------------|----|
| 1 | Introduzione | 4 |
| 2 | Descrizione | 5 |
| 3 | Creazione della classe Custom | 9 |
| 4 | Creazione del pacchetto | 10 |
| 5 | Configurazione in VBG2 | 12 |
| 6 | Esecuzione del progetto | 13 |



1 Introduzione

Con questo documento si spiega come rendere possibile creare delle classi che eseguano dei metodi di test che vengono invocati durante le operazioni di inserimento/modifica delle informazioni legate ai movimenti.



2 Descrizione

L'uso di AOP (programmazione orientata agli aspetti) permette di non vincolare al codice sviluppato aspetti non legati alla programmazione stessa (sicurezza, transazionalità delle operazioni database, auditing/log, ecc...);

Spring AOP, utilizzato in VBG, di fatto permette di accedere in maniera dichiarativa all'esecuzione di qualsiasi funzionalità del framework.

L'integratore in questo caso può creare delle classi che vengono intercettate durante l'esecuzione dell'applicativo e mandare in esecuzione del codice custom.

Nel documento che segue sarà creata una classe che stampa a log dei messaggi durante l'inserimento/modifica/cancellazione dei movimenti.

Descrizione di un caso d'uso:

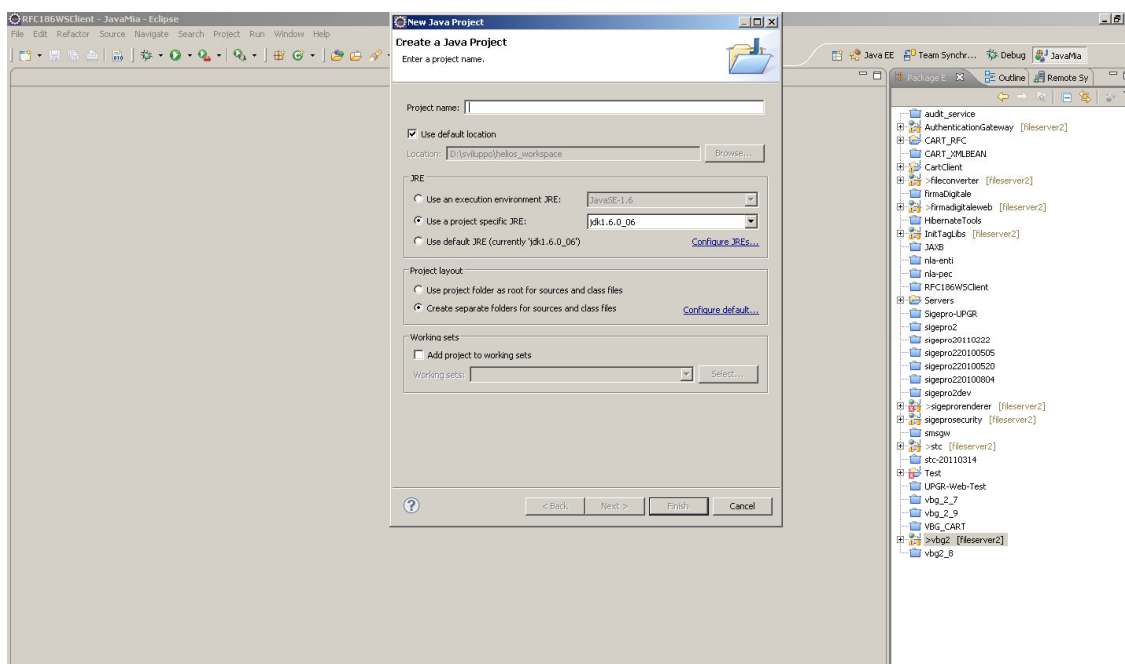
Un possibile caso d'uso potrebbe essere quello di scrivere del codice che all'inserimento di un determinato movimento (es. firma del responsabile) invia dei messaggi di posta elettronica al responsabile del procedimento del servizio.

Creazione del progetto

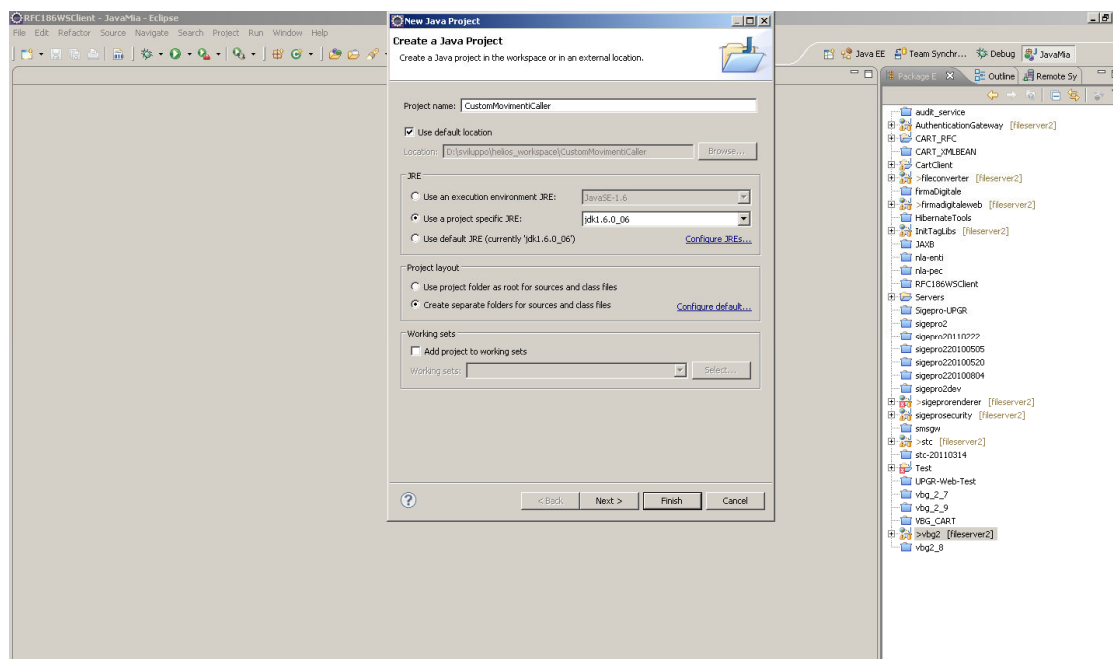
Il seguente esempio è stato creato utilizzando i seguenti software:

- IDE: Eclipse Helios
- JDK: 1.6
- Tomcat: 6.0.32

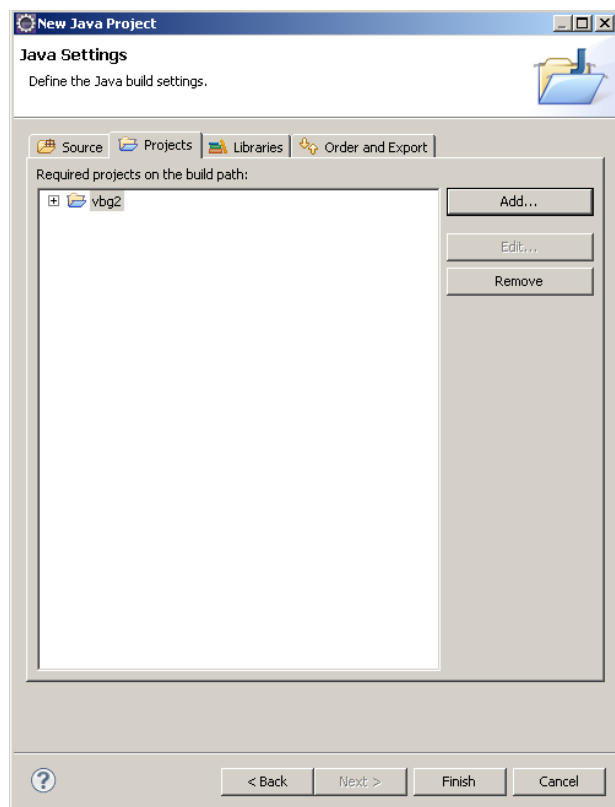
Aprire Eclipse e creare un nuovo java project



Indicare la jdk appropriata

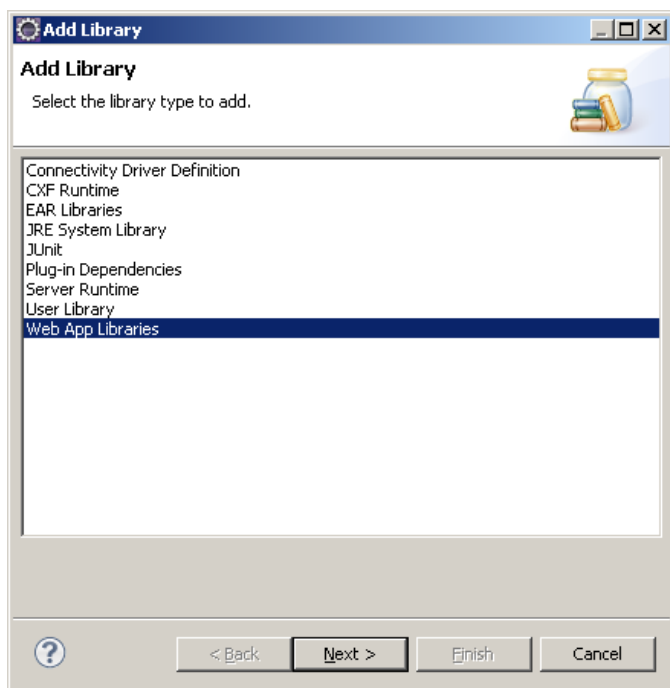


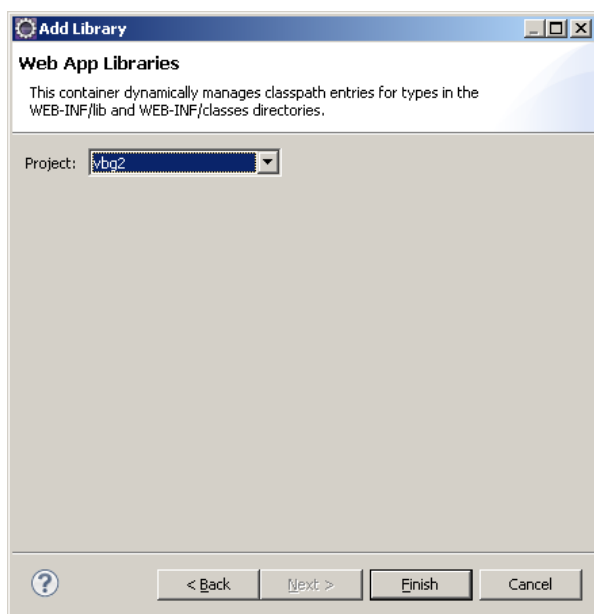
Immettere come riferimento di progetto lo war di vbg2 o le classi





Aggiungere come riferimenti le librerie di vb2 presenti nella directory WEB-INF dello war vb2







3 Creazione della classe Custom

Creare una nuova java class con il nome di package e classe voluti

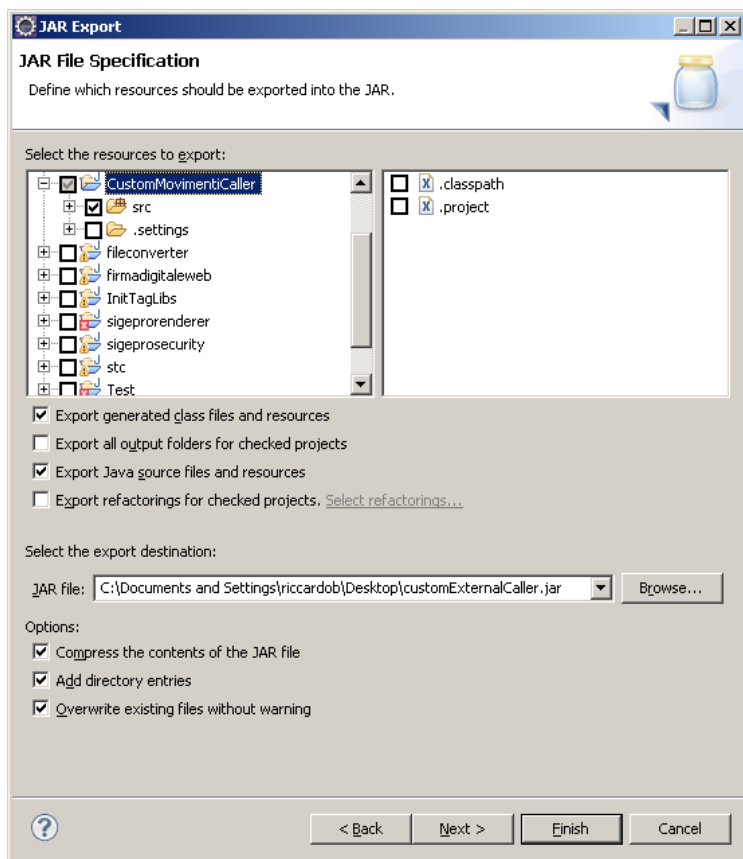
Esempio di codice che verrà invocato all'esecuzione dei metodi:

```
MovimentiCaller.java
CustomMovimentiCaller > src > it.gruppoinit.vbg.services.aop > MovimentiCaller > doCustomCall(JoinPoint) : void
1 package it.gruppoinit.vbg.services.aop;
2
3 import it.gruppoinit.sigepro.backoffice.domain.Movimenti;
4
5 import org.apache.commons.lang.StringUtils;
6 import org.aspectj.lang.JoinPoint;
7
8 public class MovimentiCaller {
9
10     public MovimentiCaller() {
11
12         System.out.println("Inizializzazione della classe " + getClass().getCanonicalName());
13     }
14
15     public void doCustomCall(JoinPoint jp) {
16
17         String metodo = jp.getSignature().getName();
18         Object[] args = jp.getArgs();
19         String codiceMovimento = "";
20         for (Object object : args) {
21             if (object instanceof Movimenti) {
22                 if (((Movimenti) object).getId().getCodice() != null) {
23                     codiceMovimento = ((Movimenti) object).getId().toString();
24                 }
25             }
26         }
27         String output = "Intercettato il metodo " + metodo;
28         if (StringUtils.isNotBlank(codiceMovimento)) {
29             output += " del movimento: " + codiceMovimento;
30         }
31         System.out.println(output);
32     }
33 }
```



4 Creazione del pacchetto

Esportare la/le classi come file jar ed includerle nella directory WEB-INF/lib di vbg2 o nel classpath del Tomcat es: \${catalina.home}/lib







5 Configurazione in VBG2

Modificare il file WEB-INF/applicationContext.xml aggiungendo:

1. la configurazione della classe che deve essere istanziata da Spring IOC

```
<bean id="customMovimentiCaller" class="it.gruppoinit.vbg.services.aop.MovimentiCaller" />
```

2. configurazione dell'aspetto che deve essere intercettato

```
<aop:aspect id="customMovimentiCallerId" ref="customMovimentiCaller" >  
<aop:pointcut id="movimentiUpdateDataMethodsId" expression="execution(*  
it.gruppoinit.sigepro.backoffice.service.MovimentiService.insert*(..)) || execution(*  
it.gruppoinit.sigepro.backoffice.service.MovimentiService.update*(..)) || execution(*  
it.gruppoinit.sigepro.backoffice.service.MovimentiService.delete*(..))" />  
<aop:before method="doCustomCall" pointcut-ref="movimentiUpdateDataMethodsId" />  
</aop:aspect>
```

6 Esecuzione del progetto

Quando viene lanciato il progetto vengono eseguiti ed intercettati l'inizializzazione della classe e l'esecuzione del metodo custom

1. Inizializzazione della classe tramite Spring

```
2-set-2011 14.48.20 org.apache.catalina.core.AprLifecycleListener init
INFO: The APR based Apache Tomcat Native library which allows optimal performance
2-set-2011 14.48.20 org.apache.tomcat.util.digester.SetPropertiesRule begin
AVVERTENZA: [SetPropertiesRule]{Server/Service/Engine/Host/Context} Setting prop
2-set-2011 14.48.20 org.apache.coyote.http11.Http11Protocol init
INFO: Initializing Coyote HTTP/1.1 on http-8080
2-set-2011 14.48.20 org.apache.catalina.startup.Catalina load
INFO: Initialization processed in 1341 ms
2-set-2011 14.48.21 org.apache.catalina.core.StandardService start
INFO: Starting service Catalina
2-set-2011 14.48.21 org.apache.catalina.core.StandardEngine start
INFO: Starting Servlet Engine: Apache Tomcat/6.0.26
2-set-2011 14.48.23 org.apache.catalina.core.ApplicationContext log
INFO: Initializing Spring root WebApplicationContext
Inizializzazione della classe it.gruppoinit.vbg.services.aop.MovimentiCaller
2-set-2011 14.51.14 org.apache.catalina.core.ApplicationContext log
```

2. Invocazione del metodo della classe custom durante l'esecuzione dei metodi individuati dal PointCUT.

