

Spring Best Practices





Speaker

Software Engineer and Architect

ProNetics

Founder

Spring Italian User Group

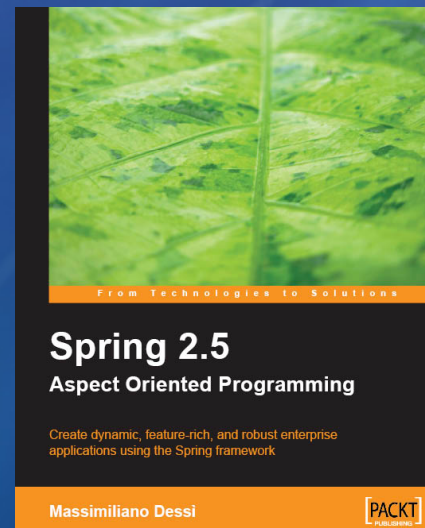
Presidente

JugSardegna Onlus

Committer/Contributor

OpenNMS – MongoDB

Autore Spring 2.5 Aspect Oriented programming



- Configurazione
 - Sicurezza
- Cache trasparente con l'AOP
 - Transazioni
 - SpringMVC
 - PortletMVC

Abbreviare l'xml con il namespace p

```
<?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:p="http://www.springframework.org/schema/p"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="due" class="it.jax.java5.Due" p:nome="Paperino"
p:cognome="Paolino"/>
```

Wildcard

```
<context-param>
```

```
    <param-name>contextConfigLocation</param-name>
```

```
    <param-value>/WEB-INF/conf/magicbox-*.xml</param-value>
```

```
</context-param>
```


Esternalizzare le parti modificabili

```
<?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:p="http://www.springframework.org/schema/p"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-2.5.xsd">

<context:property-placeholder location="WEB-INF/config.properties" />

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"
  p:url="${jdbc.url}" p:username="${jdbc.username}"
  p:password="${jdbc.password}" p:maxIdle="3"
  p:driverClassName="${jdbc.driver}" p:maxWait="50" />
```

...

Minimizzare l'XML necessario con le annotazioni

```
<?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"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-2.5.xsd">

<context:annotation-config />
<context:component-scan base-package="org.magicbox.ui" />
<context:component-scan base-package="org.magicbox.services" />
<context:component-scan base-package="org.magicbox.repositories" />

...
```

Annotazioni

@Controller @Component @Service @Repository
@Autowired @Qualifier @Configurable @Scope
@Transactional @Resource

Annotazioni test

@Test @RunWith @ContextConfiguration
@ExpectedException @IfProfileValue
@TestExecutionListeners @TransactionConfiguration

SpringSecurity

```
<filter>
  <filter-name>
    springSecurityFilterChain
  </filter-name>
  <filter-class>
    org.springframework.web.filter.DelegatingFilterProxy
  </filter-class>
</filter>
<filter-mapping>
  <filter-name>springSecurityFilterChain</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

```
<?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:sec="http://www.springframework.org/schema/security"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
http://www.springframework.org/schema/security
http://www.springframework.org/schema/security/spring-security-2.0.4.xsd">

<sec:authentication-provider user-service-ref="sffsUserDetailsService">
    <sec:password-encoder hash="sha" />
</sec:authentication-provider>

<bean id="sffsUserDetailsService"
    class="it.freshfruits.security.AuthenticationJdbcDaoImpl"
    p:rolePrefix="ROLE_" p:dataSource="dataSource"
    p:usersByUsernameQuery="SELECT username, password, enabled
        FROM authentication WHERE username = ?"
    p:authoritiesByUsernameQuery="SELECT username, authority
        FROM roles WHERE username = ?" />
```

Definire quali ruoli possono invocare gli URL

```
<sec:http>
  <sec:intercept-url pattern="/log*.jsp" filters="none" />
  <sec:intercept-url pattern="*.htm" access="ROLE_USER,ROLE_ANONYMOUS" />
  <sec:intercept-url pattern="*.page" access="ROLE_USER,ROLE_ADMIN" />
  <sec:intercept-url pattern="*.edit" access="ROLE_USER,ROLE_ADMIN" />
  <sec:intercept-url pattern="*.admin" access="ROLE_ADMIN" />
  <sec:form-login login-page="/login.jsp" default-target-url="/"
    login-processing-url="/j_security_check"
    authentication-failure-url="/loginError.jsp" />
  <sec:logout logout-url="/logout.jsp" logout-success url="/login.jsp" />
  <sec:remember-me />
</sec:http>
```

Definire quali ruoli possono eseguire i metodi

```
<bean id="accessDecisionManager" class="org.springframework.security.vote.AffirmativeBased">
  <property name="decisionVoters">
    <list>
      <bean class="org.springframework.security.vote.RoleVoter" />
      <bean class="org.springframework.security.vote.AuthenticatedVoter" />
    </list>
  </property>
</bean>

<sec:global-method-security access-decision-manager-ref="accessDecisionManager">
  <sec:protect-pointcut expression="execution(* it.myapp.domain.entity.*.*(..))"
    access="ROLE_USER,ROLE_ADMIN" />
</sec:global-method-security>
```


@Aspect

```
public class CacheAspect {  
  
    public Object cacheObject(ProceedingJoinPoint pjp) throws Throwable {  
  
        Object result;  
        String cacheKey = getCacheKey(pjp);  
        Element element = (Element) cache.get(cacheKey);  
  
        if (element == null) {  
            result = pjp.proceed();  
            element = new Element(cacheKey, result);  
            cache.put(element);  
        }  
        return element.getValue();  
    }  
  
    public void flush() {  
        cache.flush();  
    }  
}
```

```
<?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:p="http://www.springframework.org/schema/p"
xmlns:aop="http://www.springframework.org/schema/aop"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop.xsd">

<bean id="cache" abstract="true"
      class="org.springframework.cache.ehcache.EhCacheFactoryBean">
  p:cacheManager-ref="cacheManager" />

<bean id="cacheManager"
      class="org.springframework.cache.ehcache.EhCacheManagerFactoryBean"
      p:configLocation-value="classpath:it/ehcache.xml"/>
```

```
<aop:config>
  <aop:pointcut id="read" expression="execution(* it.MyClass.get*(..))" />
  <aop:pointcut id="exit" expression="execution(void it.MyClass.exit())" />

  <aop:aspect id="dummyCacheAspect" ref="rocketCacheAspect">
    <aop:around pointcut-ref="readOperation" method="cacheObject" />
    <aop:after pointcut-ref="exitOperation" method="flush" />
  </aop:aspect>
</aop:config>

<bean id="rocketCacheAspect"
  class="it.pronetics.cache.CacheAspect" >
  <property name="cache">
    <bean id="bandCache" parent="cache">
      <property name="cacheName" value="methodCache" />
    </bean>
  </property>
</bean>
```

```
<?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:p="http://www.springframework.org/schema/p"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:tx="http://www.springframework.org/schema/tx"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop.xsd">

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"
  p:url="${jdbc.url}" p:username="${jdbc.username}"
  p:password="${jdbc.password}" p:driverClassName="${jdbc.driver}" />

<bean id="transactionManager"
  class="org.springframework.jdbc.datasource.DataSourceTransactionManager"
  p:dataSource-ref="dataSource"/>
```



```
<tx:advice id="txAdvice" transaction-manager="transactionManager">
  <tx:attributes>
    <tx:method name="save*" propagation="REQUIRED"
      no-rollback-for="it.exception.FooException"
      rollback-for="it.exception.ItemException"/>
    <tx:method name="delete*" propagation="REQUIRED"
      rollback-for="it.exception.MYException"/>
    <tx:method name="disable*" propagation="REQUIRED" />
    <tx:method name="*" read-only="true" />
  </tx:attributes>
</tx:advice>

<aop:config>
  <aop:pointcut id="repoOperations"
    expression="execution(*it.pronetics.service.*.*(..))" />
  <aop:advisor advice-ref="txAdvice" pointcut-ref="repoOperations"/>
</aop:config>
```

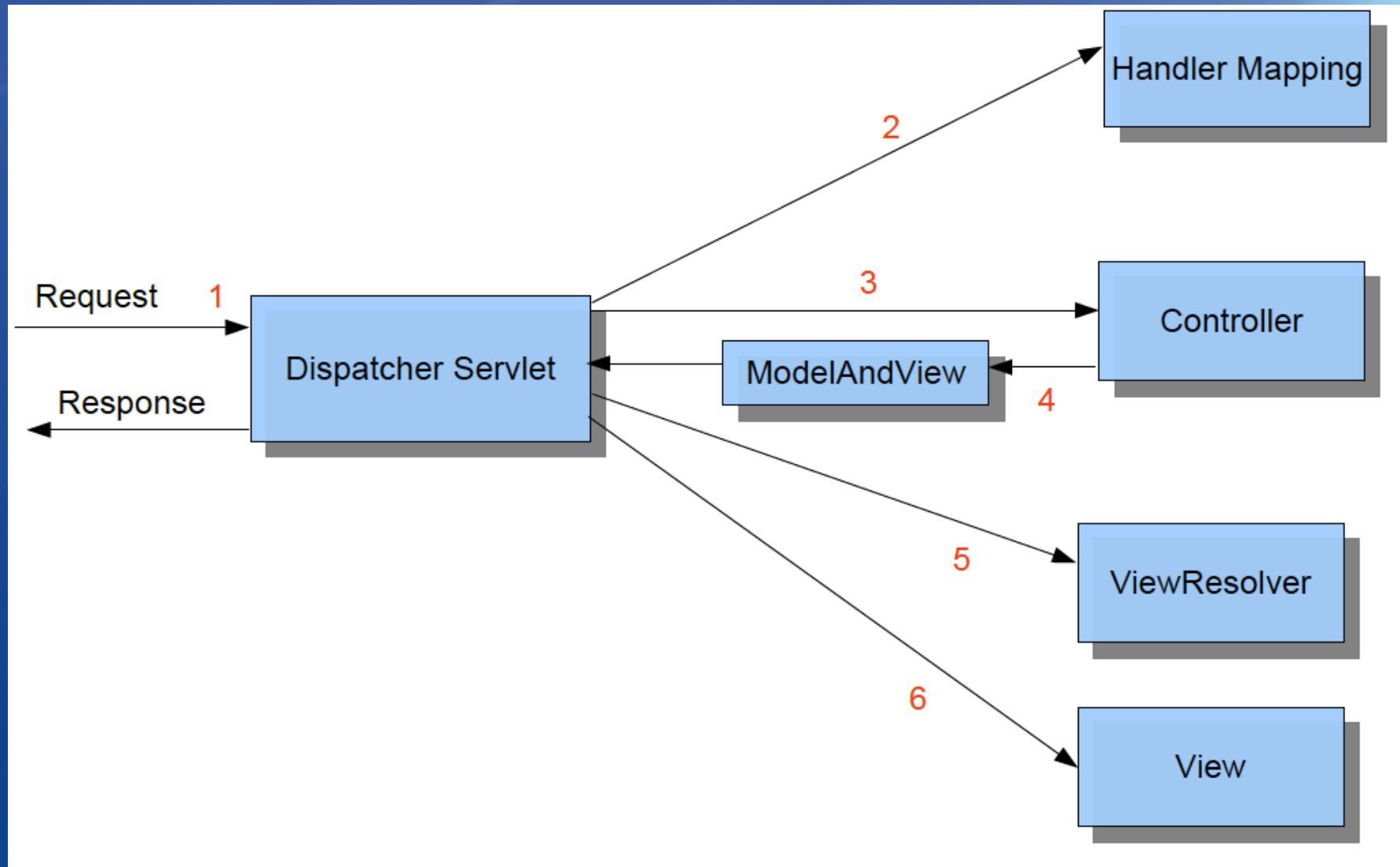
```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:p="http://www.springframework.org/schema/p"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:tx="http://www.springframework.org/schema/tx"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

<tx:annotation-driven transaction-manager="transactionManager" />

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"
    p:url="${jdbc.url}" p:username="${jdbc.username}"
    p:password="${jdbc.password}" p:driverClassName="${jdbc.driver}" />

<bean id="transactionManager" p:dataSource-ref="dataSource"
    class="org.springframework.jdbc.datasource.DataSourceTransactionManager"/>
```

```
import org.springframework.dao.DataAccessException;  
  
import org.springframework.transaction.annotation.Propagation;  
  
import org.springframework.transaction.annotation.Transactional;  
  
public class MyService implements Service{  
    ...  
  
    @Transactional(propagation=Propagation.REQUIRED,  
        rollbackFor=DataAccessException.class)  
  
    public Boolean release(String idOrder, String idItem) {  
        ...  
    }  
}
```



SpringMVC WEB.XML

```
<servlet>
  <servlet-name>spring</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <param-name>contextConfigLocation</param-name>
    <param-value>classpath:applicationContext.xml</param-value>
  <load-on-startup>0</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>spring</servlet-name>
  <url-pattern>*.page</url-pattern> <!-- estensione preferita -->
  <url-pattern>*.html</url-pattern> <!-- estensione preferita -->
  <url-pattern>*.admin</url-pattern> <!-- estensione preferita -->
</servlet-mapping>
```

```
<?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:p="http://www.springframework.org/schema/p"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">

    <context:annotation-config/>
    <context:component-scan base-package="it.pronetics.web.ui" />

    <bean id="messageSource" p:basename="it.pronetics.messages.msg"
        class="org.springframework.context.support.ResourceBundleMessageSource"/>

    <bean name="viewResolver"
        class="org.springframework.web.servlet.view.InternalResourceViewResolver"
        p:viewClass="org.springframework.web.servlet.view.JstlView"
        p:prefix="/WEB-INF/jsp/" p:suffix=".jsp"/>
```

```
@Controller("articleController")
public class ArticleController {

    private ArticleFacade facade;

    @Autowired
    public ArticleController(@Qualifier("articleFacade") ArticleFacade facade) {
        facade = articleFacade;
    }

    @RequestMapping(value = "/articles/{id}", method = RequestMethod.GET)
    public ModelAndView getArticle(@PathVariable("id") String id) {
        return new ModelAndView("articleView", "article", facade.getArticle(id));
    }

    @RequestMapping("/article/delete/{id}")
    public ModelAndView delete(@PathVariable("id") String id) {
        facade.deleteArticle(id);
        return new ModelAndView("redirect:articles");
    }
}
```



SpringMVC Form Controller

```
@Controller("articleFormController")
@SessionAttributes("article")
public class ArticleFormController {

    private ArticleFacade articleFacade;
    private ArticleValidator validator;

    @Autowired
    public ArticleFormController(@Qualifier("articleFacade") ArticleFacade facade,
        @Qualifier("articleValidator") ArticleValidator validator) {
        this.articleFacade = facade;
        this.validator = validator;
    }

    @InitBinder()
    public void initBinder(WebDataBinder binder) throws Exception {
        binder.registerCustomEditor(String.class, new StringTrimmerEditor(false));
    }
}
```


SpringMVC Form Controller

```
@RequestMapping(value= "/article/edit/{id}", method = RequestMethod.POST)
public String processSubmit(@ModelAttribute("article") ArticleCommand article,
                           BindingResult result, SessionStatus status) {
    validator.validate(article, result);
    if (result.hasErrors()) {
        return "articleForm";
    } else {
        articleFacade.saveArticle(article);
        status.setComplete();
        return Constants.REDIRECT_LIST_ARTICLES_DEFAULT;
    }
}
```

```
@RequestMapping(value= "/article/edit/{id}", method = RequestMethod.GET)
public String setupForm(@PathVariable("id") String id, ModelMap model) {
    ArticleCommand article = articleFacade.getArticle(id);
    model.addAttribute(Constants.ARTICLE,
                       article != null ? article : new ArticleCommand(););
    return "articleForm";
}
```

SpringMVC Interceptor

```
public class MyInterceptor extends HandlerInterceptorAdapter {

    @Override
    public boolean preHandle(HttpServletRequest req, HttpServletResponse res,
                             Object handler) throws Exception {

        ...
    }

    @Override
    public void postHandle(HttpServletRequest req, HttpServletResponse res,
                           Object handler, ModelAndView modelAndView) throws Exception {

        ...
    }

    @Override
    public void afterCompletion(HttpServletRequest request,
                               HttpServletResponse response,
                               Object handler, Exception ex) throws Exception {

        ..
    }
}
```

SpringMVC Interceptor

```
<bean name="urlMapping"  
  class="org.springframework.web.servlet.mvc.annotation.DefaultAnnotationHandlerMapping">  
  <property name="interceptors">  
    <list>  
      <ref bean="myInterceptor"/>  
    </list>  
  </property>  
</bean>  
  
<bean name="myInterceptor" class="it.app.MyInterceptor"/>
```

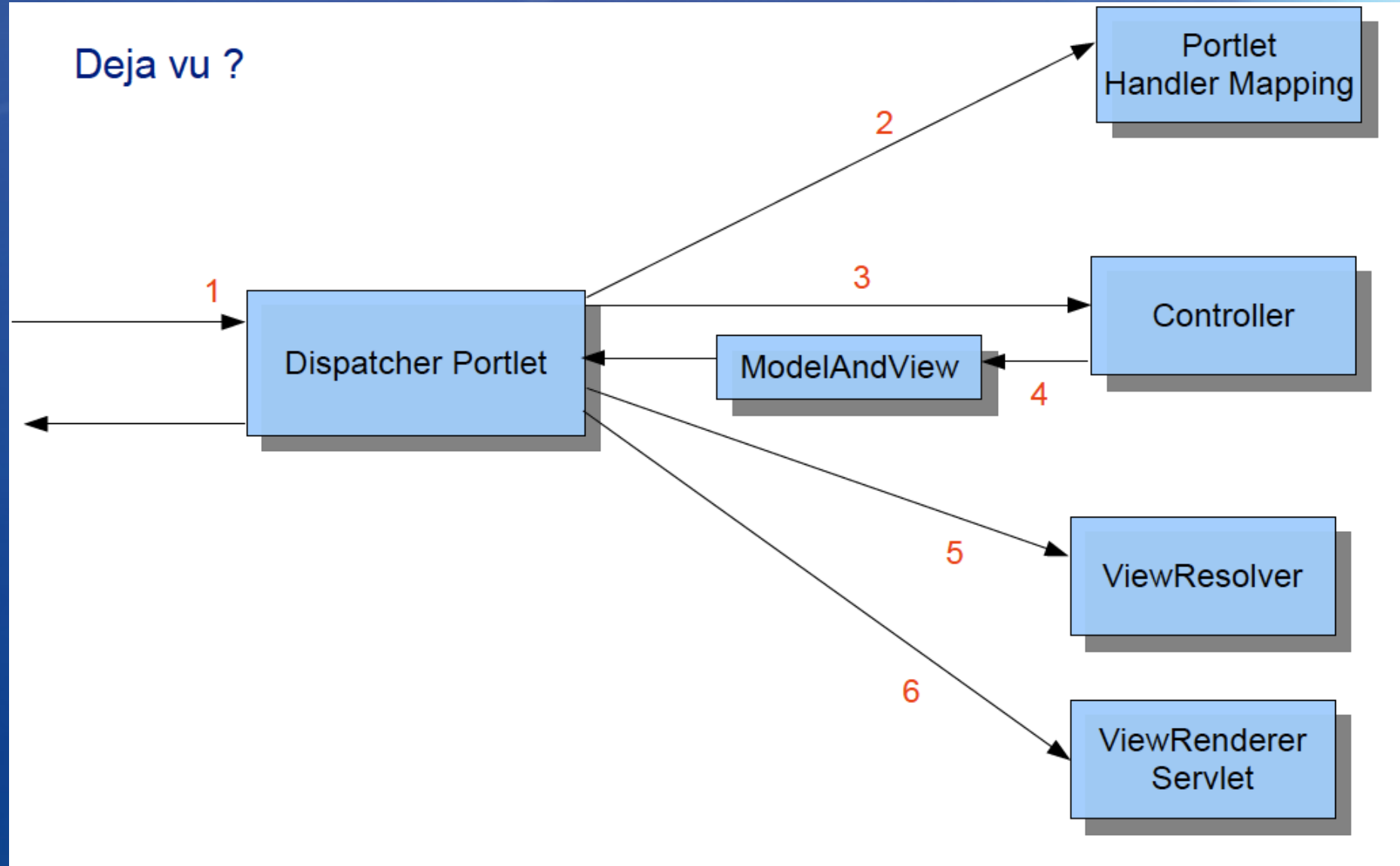
Native WebRequest

Non definisco se è in un servlet o portlet environment

```
@RequestMapping("/list.page")  
public ModelAndView list(NativeWebRequest req)
```


SpringPortletMVC

Deja vu ?



SpringPortletMVC

```
<?xml version="1.0" encoding="UTF-8"?>
<portlet-app xmlns="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd
http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd" version="2.0">
<portlet>
<portlet-name>aclPortlet</portlet-name>
<portlet-class>org.springframework.web.portlet.DispatcherPortlet</portlet-class>
  <init-param>
    <name>contextConfigLocation</name>
    <value>/WEB-INF/aclPortlet-portlet.xml</value>
  </init-param>
  <supports>
    <mime-type>text/html</mime-type>
    <portlet-mode>VIEW</portlet-mode>
  </supports>
  <portlet-info><title>Acl</title></portlet-info>
</portlet>
```

...



SpringPortletMVC

```
<?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:p="http://www.springframework.org/schema/p"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-2.5.xsd">

<context:component-scan base-package="it.pronetics.acl.ui.portlet" />

<bean name="annotationMapper"
  class="org.springframework.web.portlet.mvc.annotation.DefaultAnnotationHandlerMapping">
  <property name="interceptors">
    <list>
      <bean class="org.springframework.web.portlet.handler.ParameterMappingInterceptor"/>
      <ref bean="userFactoryInterceptor"/>
      <ref bean="authoritiesFactoryInterceptor"/>
    </list>
  </property>
</bean>
```



SpringPortletMVC

```
@Controller @RequestMapping("VIEW") @SessionAttributes("role")
public class AuthorityController {

    /*WELCOME method*/
    @RequestMapping
    public ModelAndView handleRenderRequest(RenderRequest req, RenderResponse res)
    throws Exception {

        ...
    }

    @RequestMapping(params = "action=authority.delete")
    public void deleteAuthority(ActionRequest req, ActionResponse res) {

        ...
    }

    @RequestMapping(params = "action=authority.list")
    public ModelAndView authoritiesList(RenderRequest req) {

        ...
    }
}
```


SpringPortletMVC

```
@RequestMapping(params = "action=authority.detail")  
public ModelAndView detailAuthority(RenderRequest req) {  
    ...  
}
```

```
@RequestMapping(params = "action=authority.confirm")  
public ModelAndView confirmDeleteAuthority(RenderRequest req) {  
    ...  
}
```

```
@RequestMapping(params = "action=authority.items")  
public ModelAndView authorityItems(RenderRequest req) {  
    ...  
}
```

```
@RequestMapping(params = "action=authority.selection")  
public void selectionAuthorities(@RequestParam("includedHidden") String ids,  
                                ActionRequest req, ActionResponse res) {  
    ...  
}
```

SpringPortletMVC

```
/* FORM methods*/
@InitBinder()
public void initBinder(WebDataBinder binder) throws Exception {
    binder.registerCustomEditor(String.class, new StringTrimmerEditor(false));
}

@ModelAttribute("strategies")
public List<PermissionStrategy> populatePermissionStrategy() {
    return roleStrategyService.getPermissions();
}
```

SpringPortletMVC

```
// render phase
@RequestMapping(params = "action=authority.edit")
public String showForm(Model model,
    @RequestParam(required = false, value = "aid") Integer id) {
    ...
}

// action phase
@RequestMapping(params = "action=authority.edit")
public void processSubmit(ActionResponse res,
    @ModelAttribute("authority") AuthorityDTO authority,
    BindingResult result, SessionStatus status, Model model) {
    ...
}
```

Gestione Ruoli

http://localhost:8080/acl-web/role.list.page

Access Control List

Home Ruoli Utenti logout

Ruoli
Elenco

Nome			
Lettura nodi			
dieci			
dodici			
due			
goo			
nove			
otto			
quattordici			
quattro			
quindici			
sei			
sette			
tre			
tredici			
undici			
uno			

Author's Blog

JBoss Portal 2.7.0-GA

http://localhost:8080/portal/portal/default/SpringExamplePortlet

JBoss Portal Login

Home Novita' Meteo Esempi SpringExamplePortlet

Access Control List

Home Ruoli Utenti

Ruoli
Elenco

Nome			
Lettura nodi			
dieci			
dodici			
due			
goo			
nove			
otto			
quattordici			
quattro			
quindici			
sei			
sette			
tre			
tredici			
undici			
uno			

Author's Blog

Domande ?



Grazie per l'attenzione !

Massimiliano Dessì

desmax74 at yahoo.it

massimiliano.dessi at pronetics.it

<http://twitter.com/desmax74>

<http://jroller.com/desmax>

<http://www.linkedin.com/in/desmax74>

<http://wiki.java.net/bin/view/People/MassimilianoDessi>

<http://www.jugsardegna.org/vqwiki/jsp/Wiki?MassimilianoDessi>

Spring Framework Italian User Group

<http://it.groups.yahoo.com/group/SpringFramework-it>