

SPRING MVC & SPRING SECURITY

CRAIG WALLS

ABOUT YOU...

- By show of hands...
 - Java 6? Java 5? Java 1.4? Java 1.3? Java 1.2-?
 - C#? Ruby? Groovy? Scala? Erlang? Python?
 - Spring 1.x? Spring 2.0.x? Spring 2.5.x?

ABOUT ME...

- Professionally developing software for almost 14 years
 - Java developer for most of that time
- Telecom, finance, retail, education, software
- Now at Semantra, Inc. developing natural language business intelligence solution
- Author of Spring in Action and XDoclet in Action
- Spring fanatic

E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://WWW.SPRINGLOADED.INFO) SOURCE CODE: [SVN://SVN.GEEKISP.COM/SIA](http://SVN.GEEKISP.COM/SIA) [SVN://SVN.GEEKISP.COM/HABUMA](http://SVN.GEEKISP.COM/HABUMA)

AGENDA

- Spring MVC
 - Review of Spring MVC pre-2.5
 - What's new in Spring 2.5's MVC framework
- Spring Security
 - Review of Acegi Security 1.0.x
 - Introducing Spring Security 2.0
- Summary
- Q&A

E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://WWW.SPRINGLOADED.INFO) SOURCE CODE: [SVN://SVN.GEEKISP.COM/SIA](http://SVN.GEEKISP.COM/SIA) [SVN://SVN.GEEKISP.COM/HABUMA](http://SVN.GEEKISP.COM/HABUMA)

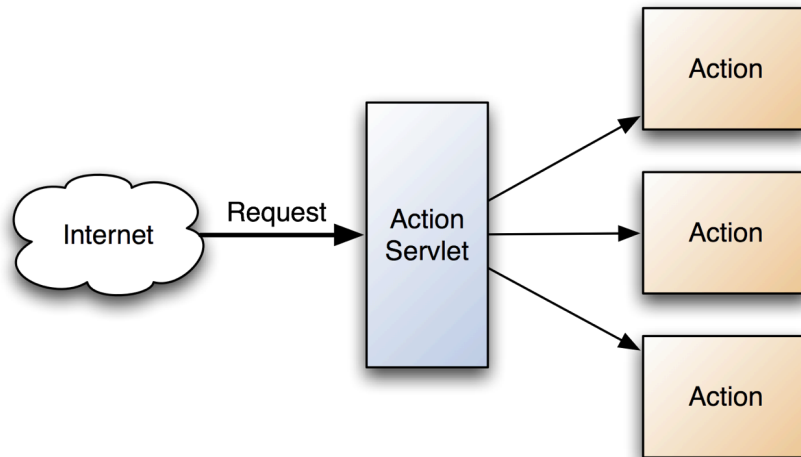
SPRING MVC

THE “OTHER” WEB FRAMEWORK

WHY SPRING MVC?

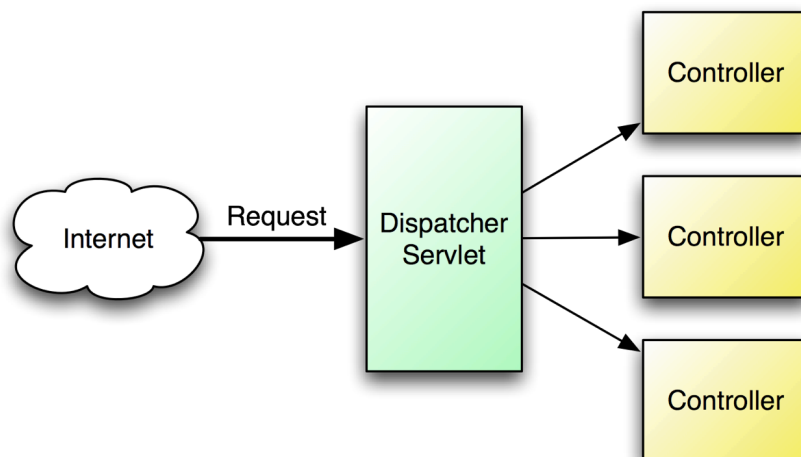
- Spring was first released in June 2003
 - 1.0 final in March 2004
- MVC choices were slim
 - At the time, Struts led the pack
 - WebWork was the next best choice
- Spring could do it better

SPRING MVC VS. STRUTS



E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://www.springloaded.info) SOURCE CODE: [SVN://SVN.GEEKISP.COM/SIA](http://svn.geekisp.com/SIA) [SVN://SVN.GEEKISP.COM/HABUMA](http://svn.geekisp.com/habuma)

SPRING MVC VS. STRUTS



E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://www.springloaded.info) SOURCE CODE: [SVN://SVN.GEEKISP.COM/SIA](http://svn.geekisp.com/SIA) [SVN://SVN.GEEKISP.COM/HABUMA](http://svn.geekisp.com/habuma)

DISPATCHERServlet

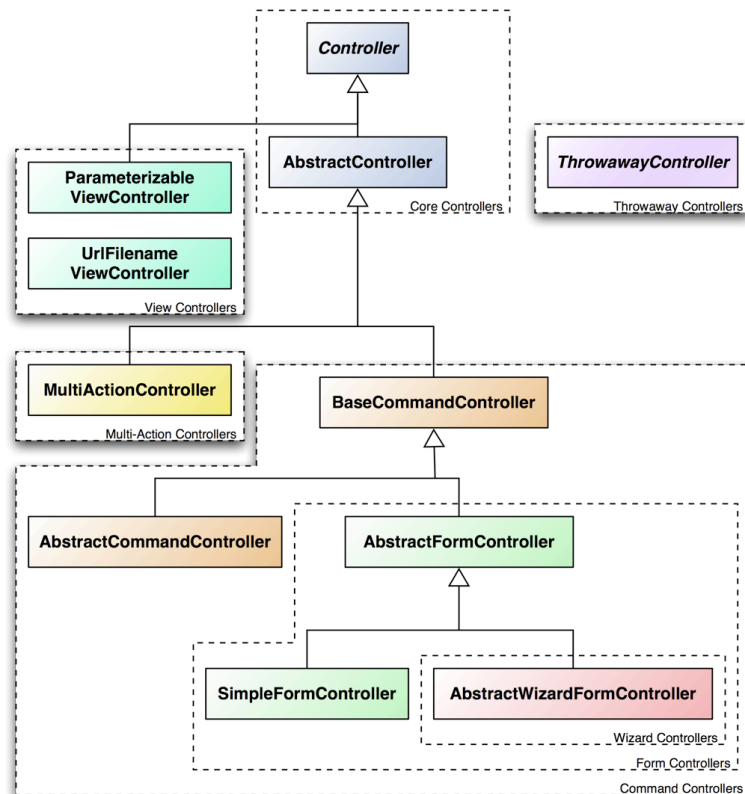
In WEB-INF/web.xml:

```
<servlet>
  <servlet-name>roadrantz</servlet-name>
  <servlet-class>
    org.springframework.web.servlet.DispatcherServlet
  </servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>roadrantz</servlet-name>
  <url-pattern>*.htm</url-pattern>
</servlet-mapping>
```

E-MAIL: CRAIG@HABUMA.COM BLOG: HTTP://WWW.SPRINGLOADED.INFO SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

SPRING'S CONTROLLERS



E-MAIL: CRAIG@HABUMA.COM BLOG: HTTP://WWW.SPRINGLOADED.INFO SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

A VERY BASIC CONTROLLER

```
public class HomeController extends AbstractController {
    protected ModelAndView handleRequestInternal(
        HttpServletRequest request, HttpServletResponse response)
        throws Exception {

        List recentRants = rantService.getRecentRants();
        return new ModelAndView("home", "rants", recentRants);
    }

    private RantService rantService;
    public void setRantService(RantService rantService) {
        this.rantService = rantService;
    }
}
```

COMMAND CONTROLLER

```
public class RantsForVehicleController extends AbstractCommandController {
    public RantsForVehicleController() {
        setCommandClass(Vehicle.class);
        setCommandName("vehicle");
    }

    protected ModelAndView handle(HttpServletRequest request,
        HttpServletResponse response, Object command,
        BindException errors) throws Exception {

        Vehicle vehicle = (Vehicle) command;

        Map model = errors.getModel();
        model.put("rants", rantService.getRantsForVehicle(vehicle));
        model.put("vehicle", vehicle);

        return new ModelAndView("rantsForVehicle", model);
    }

    private RantService rantService;
    public void setRantService(RantService rantService) {
        this.rantService = rantService;
    }
}
```

FORM CONTROLLER

```

public class AddRantFormController extends SimpleFormController {
    private static final String[] ALL_STATES = { "AL", "AK", ... };

    public AddRantFormController() {
        setCommandClass(Rant.class);
        setCommandName("rant");
    }

    protected Object formBackingObject(HttpServletRequest request)
        throws Exception {
        Rant rantForm = (Rant) super.formBackingObject(request);
        rantForm.setVehicle(new Vehicle());
        return rantForm;
    }

    protected Map referenceData(HttpServletRequest request) throws Exception {
        Map referenceData = new HashMap();
        referenceData.put("states", ALL_STATES);
        return referenceData;
    }

    protected ModelAndView onSubmit(Object command, BindException bindException)
        throws Exception {
        Rant rant = (Rant) command;
        rantService.addRant(rant);
        return new ModelAndView(getSuccessView());
    }

    private RantService rantService;
    public void setRantService(RantService rantService) {
        this.rantService = rantService;
    }
}

```

E-MAIL: CRAIG@HABUMA.COM BLOG: HTTP://WWW.SPRINGLOADED.INFO SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

CONFIGURING CONTROLLERS

In WEB-INF/roadrantz-servlet.xml:

```

<bean id="homePageController"
    class="com.roadrantz.mvc.HomePageController">
    <property name="rantService" ref="rantService" />
</bean>

<bean id="rantsForVehicleController"
    class="com.roadrantz.mvc.RantsForVehicleController">
    <property name="rantService" ref="rantService" />
</bean>

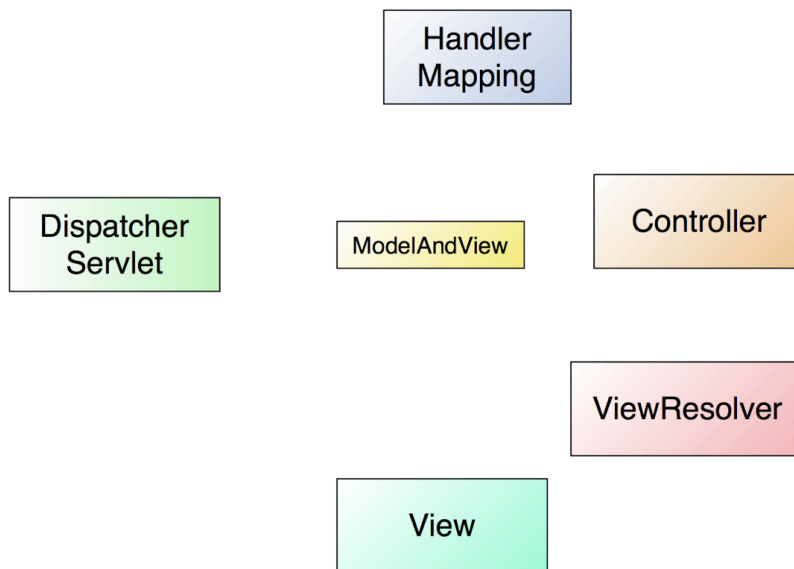
<bean id="addRantController"
    class="com.roadrantz.mvc.AddRantFormController">
    <property name="formView" value="addRant" />
    <property name="successView" value="rantAdded" />
    <property name="rantService" ref="rantService" />
</bean>

<bean id="loginController"
    class="org.springframework.web.servlet.mvc.UrlFilenameViewController" />

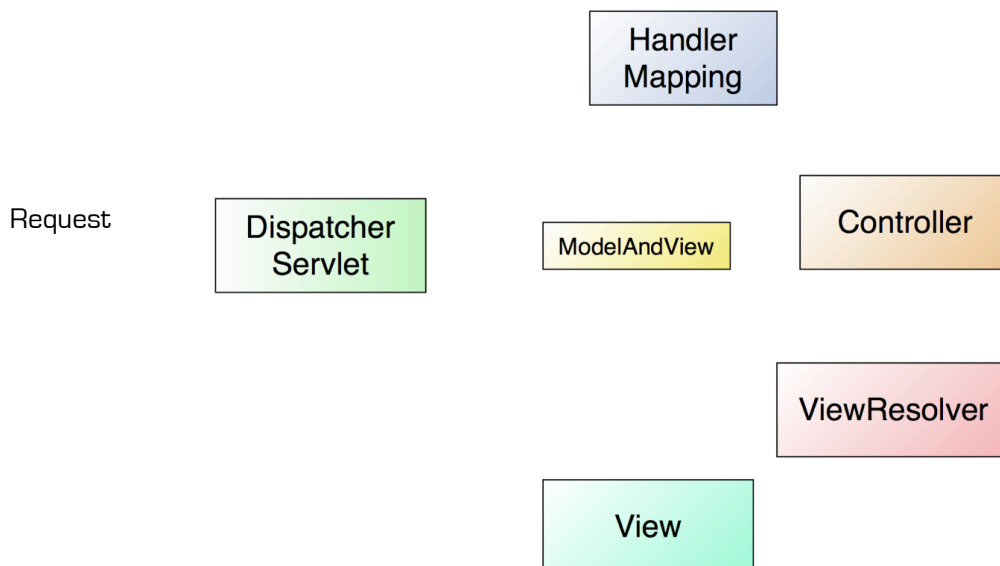
```

E-MAIL: CRAIG@HABUMA.COM BLOG: HTTP://WWW.SPRINGLOADED.INFO SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

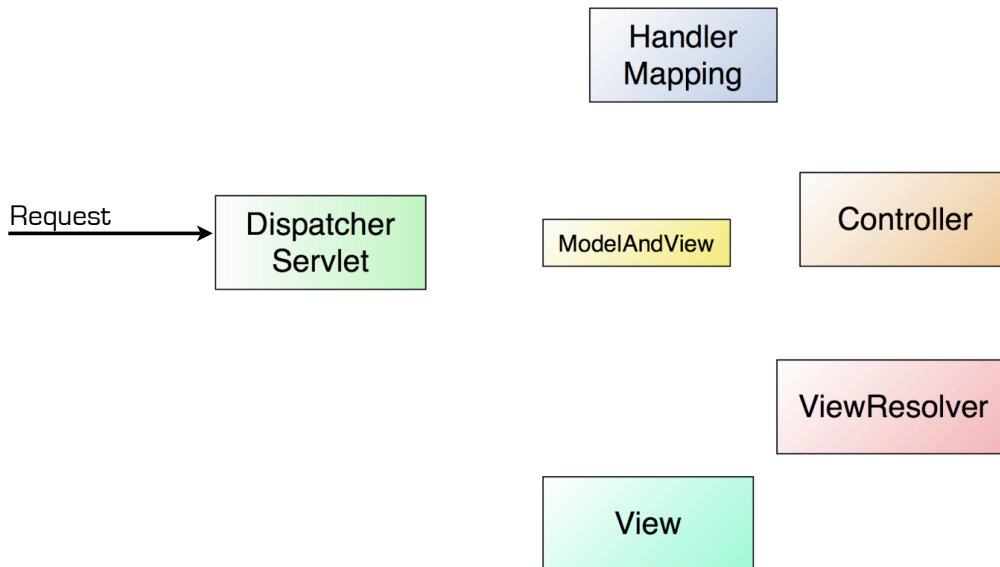
FOLLOW THAT REQUEST



FOLLOW THAT REQUEST

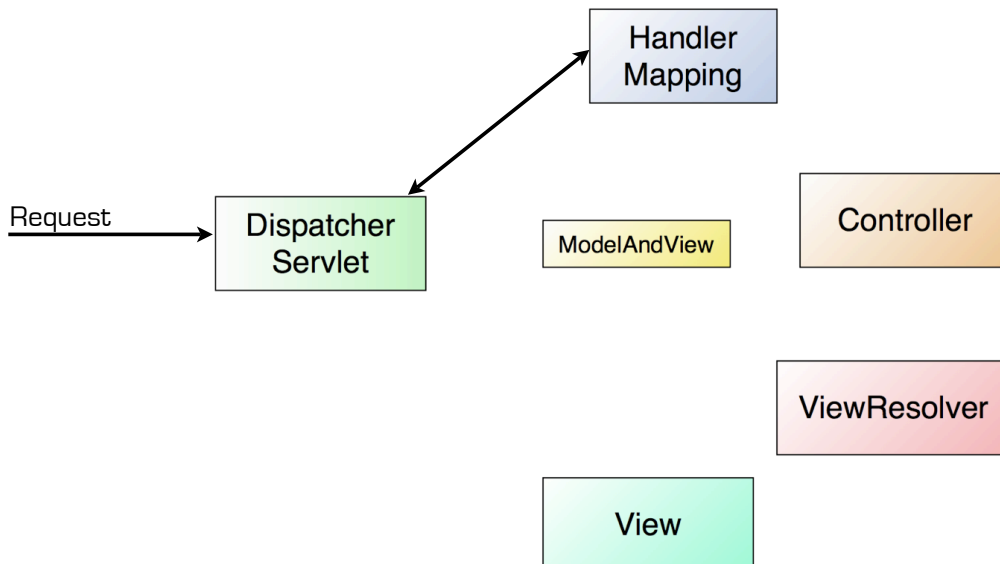


FOLLOW THAT REQUEST



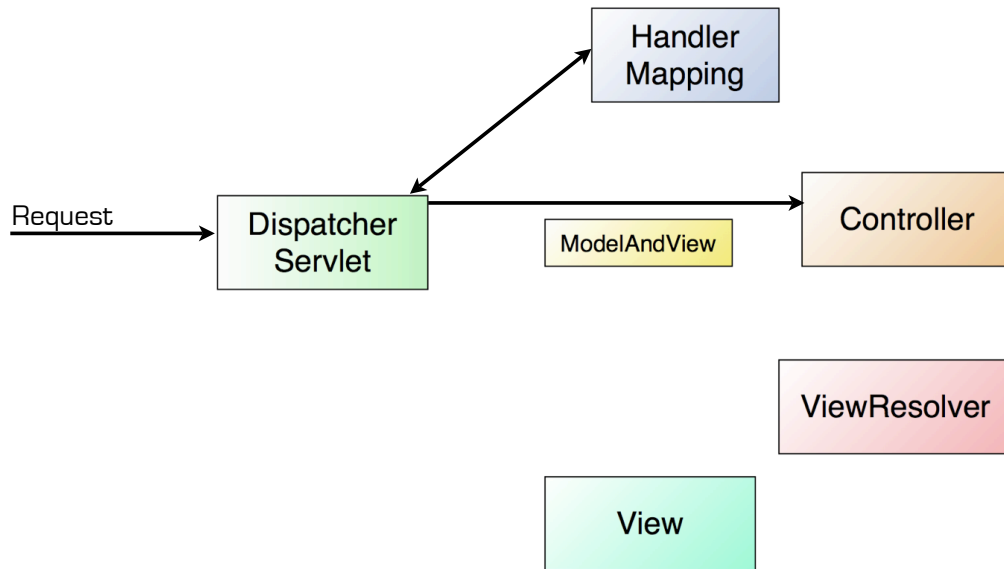
E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://WWW.SPRINGLOADED.INFO) SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

FOLLOW THAT REQUEST



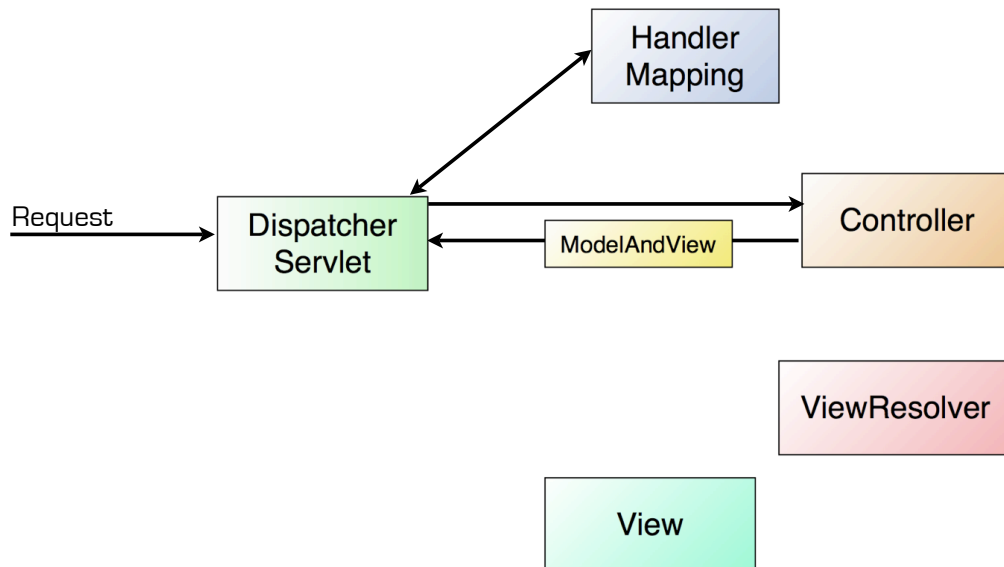
E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://WWW.SPRINGLOADED.INFO) SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

FOLLOW THAT REQUEST



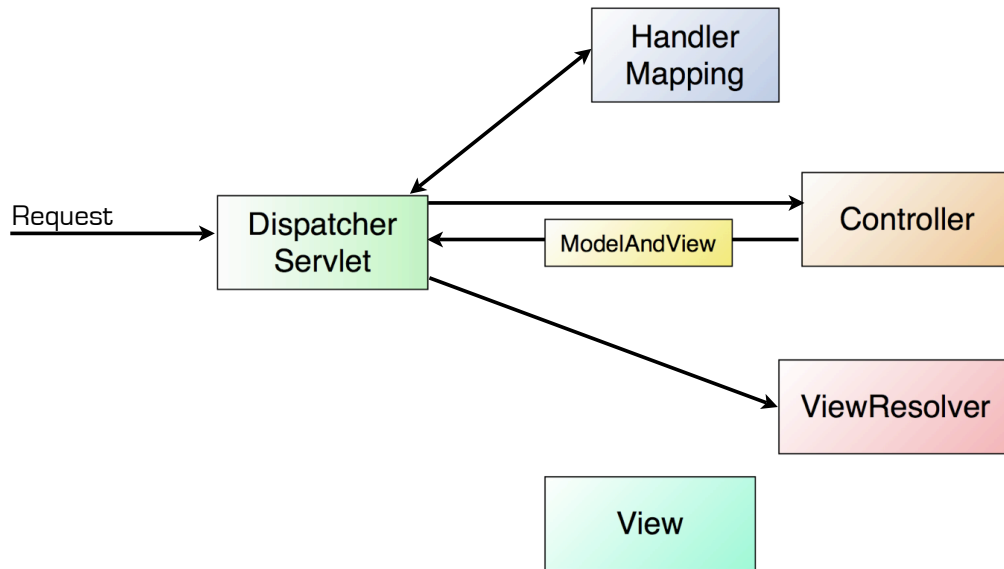
E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://www.springloaded.info) SOURCE CODE: [SVN://SVN.GEEKISP.COM/SIA](http://svn.geekisp.com/SIA) [SVN://SVN.GEEKISP.COM/HABUMA](http://svn.geekisp.com/habuma)

FOLLOW THAT REQUEST

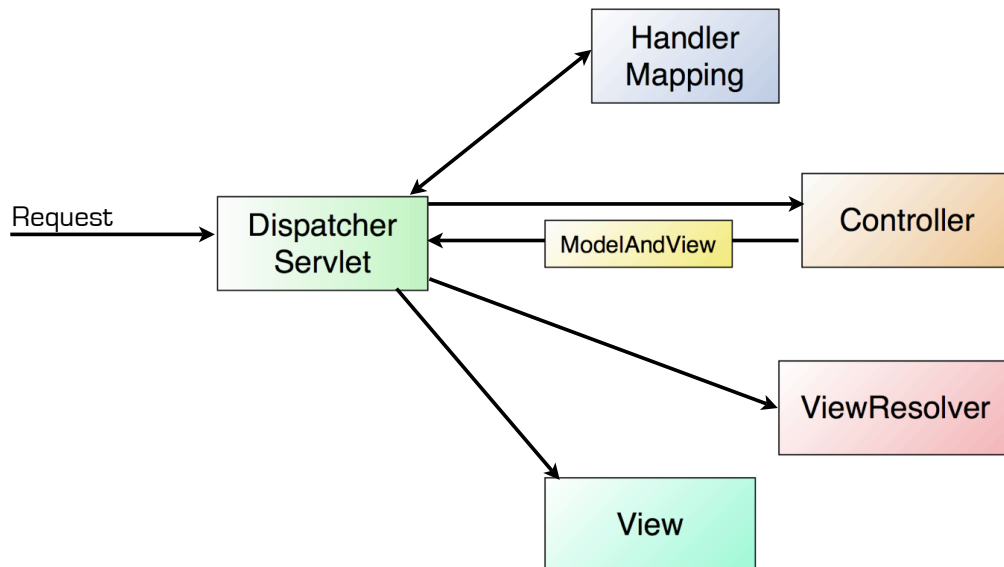


E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://www.springloaded.info) SOURCE CODE: [SVN://SVN.GEEKISP.COM/SIA](http://svn.geekisp.com/SIA) [SVN://SVN.GEEKISP.COM/HABUMA](http://svn.geekisp.com/habuma)

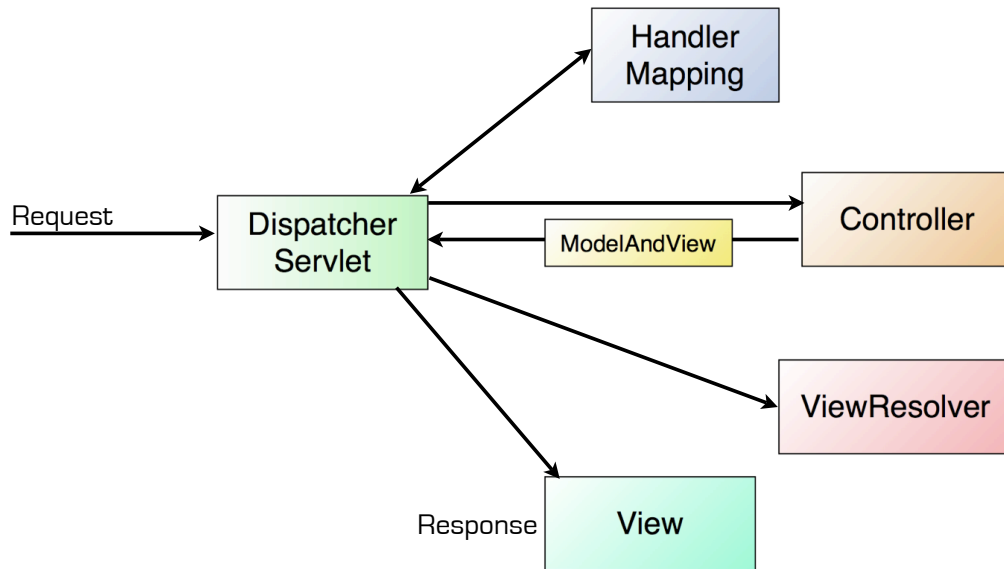
FOLLOW THAT REQUEST



FOLLOW THAT REQUEST

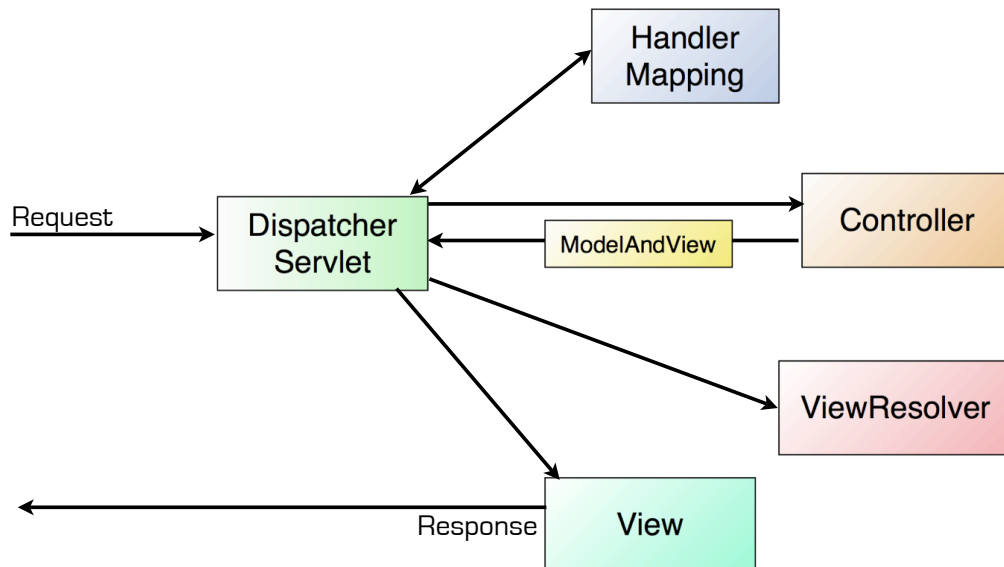


FOLLOW THAT REQUEST



E-MAIL: CRAIG@HABUMA.COM BLOG: HTTP://WWW.SPRINGLOADED.INFO SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

FOLLOW THAT REQUEST



E-MAIL: CRAIG@HABUMA.COM BLOG: HTTP://WWW.SPRINGLOADED.INFO SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

HANDLER MAPPINGS

- Map requested URL patterns to controllers
- Comes in 5 flavors:
 - BeanNameUrlHandlerMapping
 - CommonsPathMapHandlerMapping
 - **ControllerClassNameHandlerMapping**
 - **DefaultAnnotationHandlerMapping**
 - **SimpleUrlHandlerMapping**

E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://WWW.SPRINGLOADED.INFO) SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

SIMPLEURLHANDLERMAPPING

In WEB-INF/roadrantz-servlet.xml:

```
<bean id="urlMapping"
      class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
  <property name="mappings">
    <value>
      /login.htm=loginController
      /home.htm=homePageController
      /rantsForVehicle.htm=rantsForVehicleController
      /addRant.htm=addRantController
    </value>
  </property>
</bean>
```

E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://WWW.SPRINGLOADED.INFO) SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

CONTROLLERCLASSNAMEHANDLERMAPPING

In WEB-INF/roadrantz-servlet.xml:

```
<bean id="urlMapping"  
      class="org.springframework.web.servlet.mvc.support.  
            ControllerClassNameHandlerMapping" />
```

VIEW RESOLVERS

- Map logical view names in ModelAndView to actual view implementation (ex. a JSP file)
- Come in 9 flavors:
 - BeanNameViewResolver
 - FreeMarkerViewResolver
 - **InternalResourceViewResolver**
 - JasperReportsViewResolver
 - ResourceBundleViewResolver
 - VelocityLayoutViewResolver
 - VelocityViewResolver
 - XmlViewResolver
 - XsltViewResolver

INTERNALRESOURCEVIEWRESOLVER

In WEB-INF/roadrantz-servlet.xml:

```
<bean class="org.springframework.web.servlet.view.  
        InternalResourceViewResolver">  
    <property name="prefix" value="/WEB-INF/jsp/" />  
    <property name="suffix" value=".jsp" />  
</bean>
```

E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://WWW.SPRINGLOADED.INFO) SOURCE CODE: [SVN://SVN.GEEKISP.COM/SIA](http://SVN.GEEKISP.COM/SIA) [SVN://SVN.GEEKISP.COM/HABUMA](http://SVN.GEEKISP.COM/HABUMA)

SPRING JSP TAG LIBRARIES

- Two tag libraries:
 - General purpose
 - Form-bindin

E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://WWW.SPRINGLOADED.INFO) SOURCE CODE: [SVN://SVN.GEEKISP.COM/SIA](http://SVN.GEEKISP.COM/SIA) [SVN://SVN.GEEKISP.COM/HABUMA](http://SVN.GEEKISP.COM/HABUMA)

JSP TAG LIBRARIES

In WEB-INF/roadrantz-servlet.xml:

```
<%@ page contentType="text/html" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
<%@ taglib prefix="rr" tagdir="/WEB-INF/tags" %>

<html>
<body>
<h2>Enter a rant...</h2>
<form:form method="POST" action="addRant.htm" commandName="rant">
  <b><spring:message code="field.state" /></b>
  <rr:stateSelection path="vehicle.state" />
  <form:errors path="vehicle.state" cssClass="error"/><br>
  <b><spring:message code="field.plateNumber" /></b>
  <form:input path="vehicle.plateNumber" />
  <form:errors path="vehicle.plateNumber" cssClass="error"/><br>
  <b><spring:message code="field.rantText" /></b>
  <form:errors path="rantText" cssClass="error"/><br>
  <form:textarea path="rantText" rows="5" cols="50" />
  <input type="submit"/>
</form:form>
</body>
</html>
```

E-MAIL: CRAIG@HABUMA.COM BLOG: HTTP://WWW.SPRINGLOADED.INFO SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

SPRING MVC SUMMARY

- DispatcherServlet is front controller
- Handler mappings maps URLs to controllers
- Controllers process requests
 - Returning results and logical view name in ModelAndView
- View resolvers map logical view names to actual views
- Two tag libraries
 - Form-binding and general-purpose

E-MAIL: CRAIG@HABUMA.COM BLOG: HTTP://WWW.SPRINGLOADED.INFO SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

AND THEN THERE'S 2.5

- Annotation-driven
- Much less XML

E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://WWW.SPRINGLOADED.INFO) SOURCE CODE: [SVN://SVN.GEEKISP.COM/SIA](http://SVN.GEEKISP.COM/SIA) [SVN://SVN.GEEKISP.COM/HABUMA](http://SVN.GEEKISP.COM/HABUMA)

SPRING MVC 2.5 XML

In WEB-INF/roadrantz-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:context="http://www.springframework.org/schema/context"
       xmlns:p="http://www.springframework.org/schema/p"
       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="com.roadrantz.mvc" />

    <bean class="org.springframework.web.servlet.mvc.annotation.
                                     DefaultAnnotationHandlerMapping" />

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

E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://WWW.SPRINGLOADED.INFO) SOURCE CODE: [SVN://SVN.GEEKISP.COM/SIA](http://SVN.GEEKISP.COM/SIA) [SVN://SVN.GEEKISP.COM/HABUMA](http://SVN.GEEKISP.COM/HABUMA)

<CONTEXT:COMPONENT-SCAN>

- Scans all beans in a package and sub-packages
- Automatically registers beans from classes that are annotated with certain stereotype annotations
 - @Aspect
 - @Component
 - @Controller
 - @Repository
 - @Service
- Automatically autowires any properties, method parameters, or constructor args that are annotated with @Autowired

E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://WWW.SPRINGLOADED.INFO) SOURCE CODE: [SVN://SVN.GEEKISP.COM/SIA](http://SVN.GEEKISP.COM/SIA) [SVN://SVN.GEEKISP.COM/HABUMA](http://SVN.GEEKISP.COM/HABUMA)

DEFAULTANNOTATIONHANDLERMAPPING

- Maps request URLs to controllers based on @RequestMapping annotation
- @RequestMapping can be placed at class-level or at method-level

E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://WWW.SPRINGLOADED.INFO) SOURCE CODE: [SVN://SVN.GEEKISP.COM/SIA](http://SVN.GEEKISP.COM/SIA) [SVN://SVN.GEEKISP.COM/HABUMA](http://SVN.GEEKISP.COM/HABUMA)

REVISITING THE CONTROLLER

```
@Controller
@RequestMapping("/home.htm")
public class HomePageController {
    @RequestMapping(method = RequestMethod.GET)
    public String showHomePage(ModelMap model) {
        model.addAttribute(rantService.getRecentRants());

        return "home";
    }

    @Autowired
    RantService rantService;
}
```

A FORM CONTROLLER ?

```
@Controller
@RequestMapping("/addRant.htm")
public class AddRantFormController {
    @RequestMapping(method = RequestMethod.GET)
    public String setupForm(ModelMap model) {
        return "addRant";
    }

    @ModelAttribute("rant")
    public Rant setupRant() {
        Rant rant = new Rant();
        rant.setVehicle(new Vehicle());
        return rant;
    }

    @ModelAttribute("states")
    public String[] getAllStates() { return WebConstants.ALL_STATES; }

    @RequestMapping(method = RequestMethod.POST)
    protected String addRant(@ModelAttribute("rant")
        Rant rant) {
        rantService.addRant(rant);
        return "rantAdded";
    }

    @Autowired
    RantService rantService;
}
```

SPRING MVC 2.5 SUMMARY

- `<context:component-scan>` lets you skip writing XML to configure controllers (and other stuff)
- `DefaultAnnotationHandlerMapping` lets you move the URL mappings into the controllers themselves
- No more controller hierarchy...controllers are now just annotated POJOs
 - `@Controller`, `@RequestMapping`, `@ModelAttribute`

SPRING SECURITY

SAVE THE FAIRIES

ACEGI SECURITY FOR SPRING

- Created by Ben Alex in 2003
 - 1.0 in March 2004
- Applies security rules using Servlet Filters and Spring AOP
- Extremely powerful and flexible

WHAT ACEGI OFFERS

- Declarative Security
 - Keeps security details out of your code
- Authentication and Authorization
 - Against virtually any user store
- Support for anonymous sessions, concurrent sessions, remember-me, channel-enforcement, and much more
- Spring-based, but can be used for non-Spring web frameworks

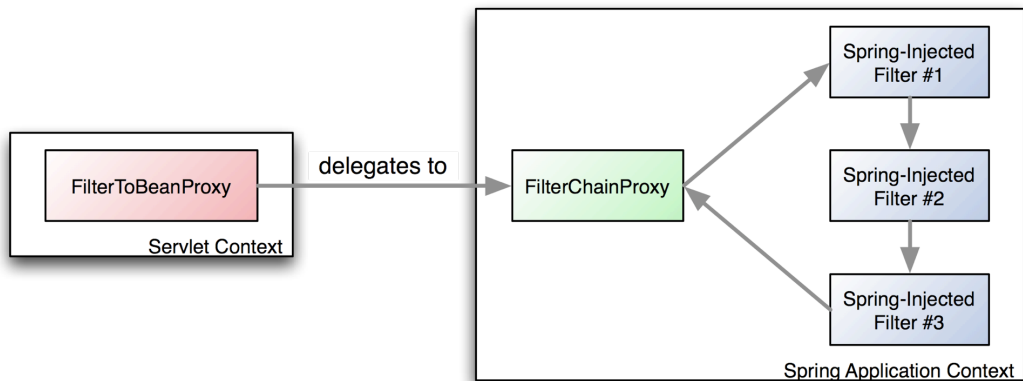
ACEGI'S FILTER PROXY

In WEB-INF/web.xml:

```
<filter>
  <filter-name>Acegi Filter Chain Proxy</filter-name>
  <filter-class>org.acegisecurity.util.FilterToBeanProxy</filter-class>
  <init-param>
    <param-name>targetClass</param-name>
    <param-value>org.acegisecurity.util.FilterChainProxy</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>Acegi Filter Chain Proxy</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

E-MAIL: CRAIG@HABUMA.COM BLOG: HTTP://WWW.SPRINGLOADED.INFO SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

FILTER PROXY



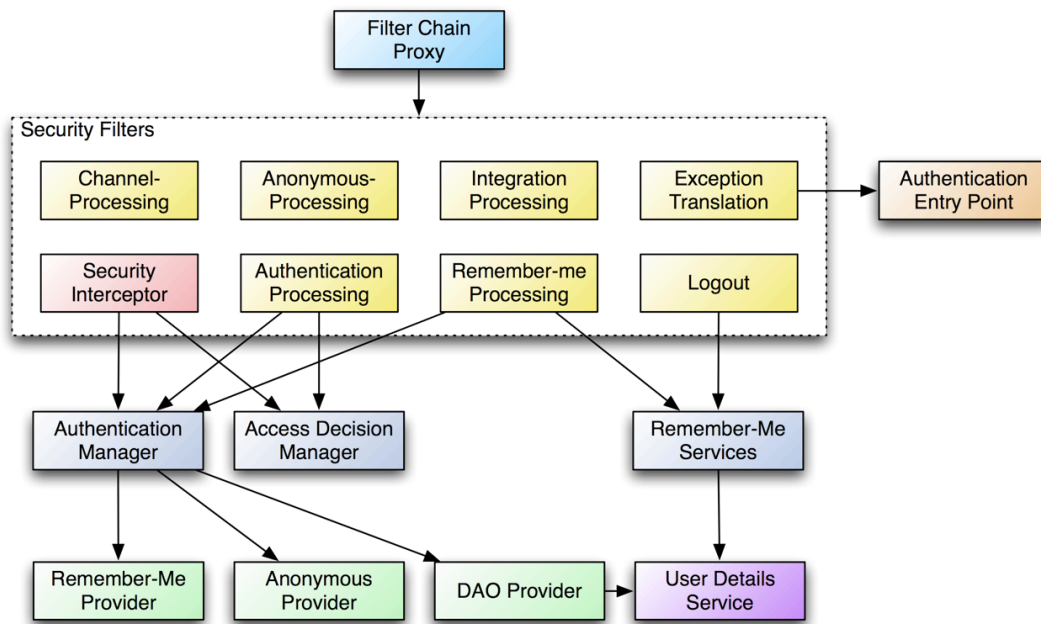
E-MAIL: CRAIG@HABUMA.COM BLOG: HTTP://WWW.SPRINGLOADED.INFO SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

EXAMPLE ACEGI CONFIG

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2003/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.0.xsd">
    <bean id="filterChainProxy"
          class="org.acegisecurity.util.FilterChainProxy">
        <property name="filterInvocationDefinitionSource">
            <value>
                CONVERT_URL_TO_LOWERCASE_BEFORE_COMPARISON
                PATTERN_TYPE_APACHE_ANT
                /**=channelProcessingFilter,httpsessionIntegrationFilter,
                logoutFilter,authenticationProcessingFilter,rememberMeProcessingFilter,
                anonymousProcessingFilter,exceptionTranslationFilter,filterSecurityInterceptor
            </value>
        </property>
    </bean>
    <bean id="authenticationProcessingFilter"
          class="org.acegisecurity.ui.webapp.AuthenticationProcessingFilter">
        <property name="authenticationManager" ref="authenticationManager"/>
        <property name="authenticationFailureUrl" value="/login.htm?login_error=1"/>
        <property name="defaultRememberUrl" value="/" />
        <property name="filterProcessesUrl" value="/j_acegi_security_check"/>
        <property name="rememberMeServices" ref="rememberMeServices"/>
    </bean>
    <bean id="authenticationManager"
          class="org.acegisecurity.providers.ProviderManager">
        <property name="providers">
            <list>
                <ref bean="daoAuthenticationProvider"/>
                <ref bean="anonymousAuthenticationProvider"/>
                <ref bean="rememberMeAuthenticationProvider"/>
            </list>
        </property>
    </bean>
    <bean id="daoAuthenticationProvider"
          class="org.acegisecurity.providers.dao.DaoAuthenticationProvider">
        <property name="userDetailsService"
            ref="userDetailsService"/>
    </bean>
    <bean id="userDetailsService"
          class="org.acegisecurity.userdetails.jdbc.JdbcDaoImpl">
        <property name="dataSource" ref="dataSource"/>
        <property name="usersByUsernameQuery"
            value="SELECT email as username, password, 'true' FROM Motorist WHERE email=?"/>
        <property name="authoritiesByUsernameQuery"
            value="SELECT email as username, privilege FROM Motorist_Privileges mp, Motorist m WHERE
            mp.motorist_id = m.id AND m.email=?"/>
    </bean>
    <bean id="authenticationEntryPoint"
          class="org.acegisecurity.ui.webapp.AuthenticationProcessingFilterEntryPoint">
        <property name="loginFailureUrl" value="/login.htm"/>
        <property name="forceHttps" value="true"/>
    </bean>
    <bean id="accessDecisionManager"
          class="org.acegisecurity.vote.UnanimousBased">
        <property name="unanimousBased" value="false"/>
        <property name="decisionVoters">
            <list>
                <bean class="org.acegisecurity.vote.RoleVoter"/>
            </list>
        </property>
    </bean>
    <bean id="exceptionTranslationFilter"
          class="org.acegisecurity.ui.ExceptionTranslationFilter">
        <property name="authenticationEntryPoint"
            ref="authenticationEntryPoint"/>
        <property name="accessDeniedHandler">
            <bean class="org.acegisecurity.ui.AccessDeniedHandlerImpl">
                <property name="errorPage" value="/error.htm"/>
            </bean>
        </property>
    </bean>
    <bean id="httpsessionIntegrationFilter"
          class="org.acegisecurity.context.HttpSessionContextIntegrationFilter">
        <property name="forceHttpSessionCreation" value="true"/>
    </bean>
    <bean id="filterSecurityInterceptor"
          class="org.acegisecurity.intercept.web.FilterSecurityInterceptor">
        <property name="authenticationManager" ref="authenticationManager"/>
        <property name="accessDecisionManager" ref="accessDecisionManager"/>
        <property name="objectDefinitionSource">
            <value>
                CONVERT_URL_TO_LOWERCASE_BEFORE_COMPARISON
                PATTERN_TYPE_APACHE_ANT
                /doget*.htm=ROLE_BOOGER
            </value>
        </property>
    </bean>
    <bean id="anonymousProcessingFilter"
          class="org.acegisecurity.providers.anonymous.AnonymousProcessingFilter">
        <property name="key" value="fooBar"/>
        <property name="userAttribute" value="anonymousUser,ROLE_ANONYMOUS"/>
    </bean>
    <bean id="anonymousAuthenticationProvider"
          class="org.acegisecurity.providers.anonymous.AnonymousAuthenticationProvider">
        <property name="key" value="fooBar"/>
    </bean>
    <bean id="rememberMeProcessingFilter"
          class="org.acegisecurity.ui.rememberme.RememberMeProcessingFilter">
        <property name="rememberMeServices" ref="rememberMeServices"/>
        <property name="authenticationManager" ref="authenticationManager"/>
    </bean>
    <bean id="rememberMeServices"
          class="org.acegisecurity.ui.rememberme.TokenBasedRememberMeServices">
        <property name="userDetailsService" ref="userDetailsService"/>
        <property name="key" value="roadAntz"/>
    </bean>
    <bean id="rememberMeAuthenticationProvider"
          class="org.acegisecurity.providers.rememberme.RememberMeAuthenticationProvider">
        <property name="key" value="roadAntz"/>
    </bean>
    <bean id="logoutFilter"
          class="org.acegisecurity.ui.logout.LogoutFilter">
        <constructor-arg value="/name.htm"/>
        <list>
            <ref bean="rememberMeServices"/>
        </list>
        <bean class="org.acegisecurity.ui.logout.SecurityContextLogoutHandler"/>
    </bean>
    <bean id="channelProcessingFilter"
          class="org.acegisecurity.securechannel.ChannelProcessingFilter">
        <property name="channelDecisionManager" ref="channelDecisionManager"/>
        <property name="filterInvocationDefinitionSource">
            <value>
                CONVERT_URL_TO_LOWERCASE_BEFORE_COMPARISON
                PATTERN_TYPE_APACHE_ANT
                /login.htm=REQUIRES_SECURE_CHANNEL
                /j_acegi_security_check=REQUIRES_SECURE_CHANNEL
                /**=REQUIRES_INSECURE_CHANNEL
            </value>
        </property>
    </bean>
</beans>
```

E-MAIL: CRAIG@HABUMA.COM BLOG: HTTP://WWW.SPRINGLOADED.INFO SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

WHAT WAS IN THAT XML?



E-MAIL: CRAIG@HABUMA.COM BLOG: HTTP://WWW.SPRINGLOADED.INFO SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

ACEGI'S MOVING PARTS

- **Security Interceptor** - Determines what resources are to be secured—and how they are to be secured
- **Filters** - Intercept requests and apply various aspects of security
 - Authentication, Remember-me, Concurrent sessions, Anonymous, Logout, Channel-processing, Integration, etc
- **Authentication Manager** - Determines who the user is and what authorities they are granted
 - **Authentication Providers** - Used by the authentication manager to retrieve user information. Includes DAO-based, anonymous, and remember-me providers
- **Authorization Manager** - Allows/Disallows access to a resource, given a user's granted authorities and a resource's required authorization
 - Usually defers to one or more access decision voters to help decide

E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://WWW.SPRINGLOADED.INFO) SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

SPRING SECURITY 2.0

- All the same goodness of Acegi...and more
- Exploits custom Spring configuration namespaces (ala Spring 2.0)
- Acegi's complex configuration details now hidden behind simpler XML elements
- Offers auto-configuration

E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://WWW.SPRINGLOADED.INFO) SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

SPRING SECURITY 2.0 FILTER

In WEB-INF/web.xml:

```
<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>
```

E-MAIL: CRAIG@HABUMA.COM BLOG: HTTP://WWW.SPRINGLOADED.INFO SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

SIMPLER CONFIGURATION

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns="http://www.springframework.org/schema/security"
  xmlns:beans="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-2.5.xsd
    http://www.springframework.org/schema/security
    http://www.springframework.org/schema/security/spring-security-2.0.xsd">

  <http auto-config="true">
    <intercept-url pattern="/addRant.htm" access="ROLE_MOTORIST" />
    <intercept-url pattern="/home.htm" requires-channel="http" />
    <intercept-url pattern="/login.htm" requires-channel="https" />
    <form-login login-page="/login.htm" />
  </http>

  <authentication-provider user-service-ref="userService" />

  <jdbc-user-service id="userService" data-source-ref="dataSource" />
</beans:beans>
```

E-MAIL: CRAIG@HABUMA.COM BLOG: HTTP://WWW.SPRINGLOADED.INFO SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

<HTTP>:THE MAGIC ELEMENT

- The central configuration element for web security
- <intercept-url> declares pages to be secured (and how)
- <form-login> refers to a login page
- Other sub-elements provide support for HTTP Basic authentication, Logout, Remember-Me, and Anonymous sessions...or...
- “auto-config” does all of that for you!
 - In fact, it can also automatically create a login form for you

E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://WWW.SPRINGLOADED.INFO) SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

<AUTHENTICATION-PROVIDER>

- Declares an authentication provider
- Refers to a user details service

E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://WWW.SPRINGLOADED.INFO) SOURCE CODE: SVN://SVN.GEEKISP.COM/SIA SVN://SVN.GEEKISP.COM/HABUMA

<JDBC-USER-SERVICE>

- Declares a JDBC-based user service
- Retrieves user information and authorization details from a database table.
 - Assumes "SELECT username,password,enabled FROM users WHERE username=?" for user details.
 - Assumes "SELECT username,authority FROM authorities WHERE username=?" for authorization info.
 - As of 2.0-m4, can't override SQL. See and (please) vote for SEC-703.

E-MAIL: CRAIG@HABUMA.COM BLOG: [HTTP://WWW.SPRINGLOADED.INFO](http://WWW.SPRINGLOADED.INFO) SOURCE CODE: [SVN://SVN.GEEKISP.COM/SIA](http://SVN.GEEKISP.COM/SIA) [SVN://SVN.GEEKISP.COM/HABUMA](http://SVN.GEEKISP.COM/HABUMA)

SUMMARY

IS IT OVER YET?

SPRING MVC & SECURITY

- Spring continues to evolve and improve
- This especially applies to Spring MVC and Spring Security
 - Spring MVC 2.5 has embraced annotations for configuration, reducing the amount of XML required
 - Spring Security 2.0 has dramatically decreased the amount of XML required by employing security-specific configuration elements

EXAMPLE CODE & SLIDES

- RoadRantz example (from SiA) updated to use the latest Spring goodness:
 - [svn://svn.geekisp.com/SiA/trunk/RoadRantz](http://svn.geekisp.com/SiA/trunk/RoadRantz)
- Updated to use Spring MVC 2.5, Spring Security 2.0 nightly build (with SEC-703 patch), and Spring WebFlow 2.0-m4
- Work in progress—No guarantees
- These slides: <http://www.habuma.com/spring/SpringMVCSecurity.pdf>

Q & A

TIME TO PLAY STUMP THE SPEAKER