**Configurazioni WAC**

1. Xml
2. Classpath Scanning
3. @Configuration class

2) component scanning. Specifica classpath da cui partire (sub packages) per caricare i componenti in base alle loro annotations di stereotipo.

3) @Configuration class  
Classe refererenziata da

**COMPONENTI DEFAULT:**

The configuration of a Spring web application integrates quite a few infrastructure beans.

The DispatcherServlet looks for implementations of type:

* HandlerMapping,
* HandlerAdapter,
* ViewResolver,
* HandlerExceptionResolver.

Out-of-the-box implementations for the previously mentioned interfaces are provided by Spring.

**IMPL DEFAULT**  
The default configuration can be found in the

spring-webmvc.jar in   
package org.springframework.web.servlet

DispatcherServlet.properties,

**Ruolo**

**<<Handler>> .handle**Handler della request (Componente.metodo)

**HandlerMapping**In base a dati in http request seleziona (tra quelli disponibili nel WAC) un HandlerAdapter.  
(chain di Interceptors, HandlerAdapter)

**HandlerAdapter:**Astrazione del “gestore richiesta”. (Componente.metodo)  
Di fatto un Wrapper intorno ai Controller.  
  
**ViewResolver**Componente che mappa la String di routing del MAV in un oggetto View.

**View**Oggetto che mappa Model del MAV in una response http.

**HandlerExceptionResolver:**Mappa oggetto Exception con un Handler (Componente

Minore Importanza:

**OVERRIDE DEFAULTS:**

The infrastructure beans mentioned earlier can be configured manually, but this is rarely done and is recommended to be avoided, as the explicit configuration cancels the default configuration for that beantype.   
  
In Spring 4.0, <mvc:annotation-driven/> and the equivalent @EnableWebMvc do just that—override the default configuration to provide the new features, so you don’t have to struggle with the configuration yourself.

1. Classe default di creazione WAC WebMvcConfigurationSupport
2. Annotare una classe di @Configuration con @EnableMvc o inserire in uno dei file xml di configurazione <mvc:annotation-driven/> permette di importare (tramite estensione) tutti I bean definiti in questa classe (1)
3. Attraverso dei metodi e’ possibile sovrascrivere uno dei component che si intende modificare senza il rischio di disabilitare/misconfigurare gli altri.

Override default con @EnableWebMvc

Mutlipart

1) SimpleControllerHandlerAdapter (default)

sceglie

in base a criteri BeanNameUrlHandlerMapping

una classe che implementa Controller interface

tra quelle registrate nel WAC

2) RequestMappingHandlerAdapter (SMVC 3.2+)

Tra tutte le classi caricate dal WAC cerca il metodo giusto cercando tra

- Classi annotate con @Controller

- metodi/classi annotati con @RequestMapping

This adapter class was introduced in Spring 3.1, deprecating the AnnotationMethodHandlerAdaptor handler adapter in Spring 3.2.

It’s used with RequestMappingHandlerMapping class, which executes methods annotated with @RequestMapping.

3) AnnotationMethodHandlerAdapter (versione pre SMVC 3.1)

Tra tutte le classi caricate dal WAC cerca il metodo giusto cercando tra

- Classi annotate con @Controller

- metodi/classi annotati con @RequestMapping

4) SimpleServletHandlerAdapter

Delega la gestione della richiesta ad una Servlet

Override default con @EnableWebMvc

The infrastructure beans mentioned earlier can be configured manually, but this is rarely done and is recommended to be avoided, as the explicit configuration cancels the default configuration for that beantype.   
  
In Spring 4.0, <mvc:annotation-driven/> and the equivalent @EnableWebMvc do just that—override the default configuration to provide the new features, so you don’t have to struggle with the configuration yourself.

Chaining configurazioni con priorita

A Spring web application can use more than one infrastructure bean of a specific type. In this case, the

beans can be chained and have an associated priority value specified using the order property. For example,

you can have multiple HandlerMapping implementations:

@EnableWebMvc E request mapping

When <mvc:annotation-driven/> or @EnableWebMVC is used in the application configuration,

the RequestMappingHandlerMapping implementation is registered internally with Spring MVC. This

class was added in Spring 3.1; it allows RequestMappings for the same URL to be in different controller

classes. It is meant to replace the DefaultAnnotationHandlerMapping implementation. It was introduced

to make the annotation controller support class more customizable and open for extension. When

using the RequestMappingHandlerMapping, the actual handler is an instance of HandlerMethod, which

identifies the specific controller method that will be invoked. Starting with Spring version 4.0, the

DefaultAnnotationHandlerMapping was marked as deprecated.

**Annotations:**

Annotation @EnableWebMvc   
  
Vedere su javadocs:

* 1. Da sola estende la classe WebMvcConfigurationSupport con gli oggetti di default del contesto web
  2. Insieme a extends WebMvcConfigurerAdapter permette alcune customizzazioni dei component di default tramite override degli apposite metodi.
  3. Per avere il Massimo controllo sugli oggetti di default del wac estendere direttamente WebMvcConfigurationSupport (e non annotare la classe)

.

**CODE REVIEW:**

1. DispatcherServlet
2. Ruolo ContextLoaderListener

org.springframework.web.context.ContextLoaderListener

.contextInitialized

. initWebApplicationContext