

**Front Controller - DispatcherServlet** - uses **{servletname}-servlet.xml** in webinf dirctory as config file by default. override this by passing file name in init param - **contextConfigLocation**

## special bean types

**HandlerMapping**: Maps incoming requests to handlers and a list of pre- and post-processors (handler interceptors) based on some criteria the details of which vary by HandlerMapping implementation.

**HandlerAdapter** Helps the DispatcherServlet to invoke a handler mapped to a request

regardless of the handler is actually invoked. For example, invoking an annotated

controller requires resolving various annotations. Thus the main purpose of a

HandlerAdapter is to shield the DispatcherServlet from such details.

**HandlerExceptionResolaver** : Maps exceptions to views also allowing for more complex exception handling code.

**ViewResolver:**  Resolves logical String-based view names to actual View types.

**LocaleResolver**: Resolves the locale a client is using, in order to be able to offer internationalized views

**ThemeResolver**: Resolves themes your web application can use, for example, to offer personalized layouts

**MultipartResolver:** Parses multi-part requests for example to support processing file uploads from HTML forms.

**FlashMapManager** Stores and retrieves the "input" and the "output" FlashMap that can be used to pass attributes from one request to another, usually across a redirect.

## Workflow/Architecture

DispatcherServlet starts processing the request as follows:

1. The WebApplicationContext is searched for and bound in the request as an attribute that the controller and other elements in the process can use. It is bound by default under the key DispatcherServlet.WEB\_APPLICATION\_CONTEXT\_ATTRIBUTE.

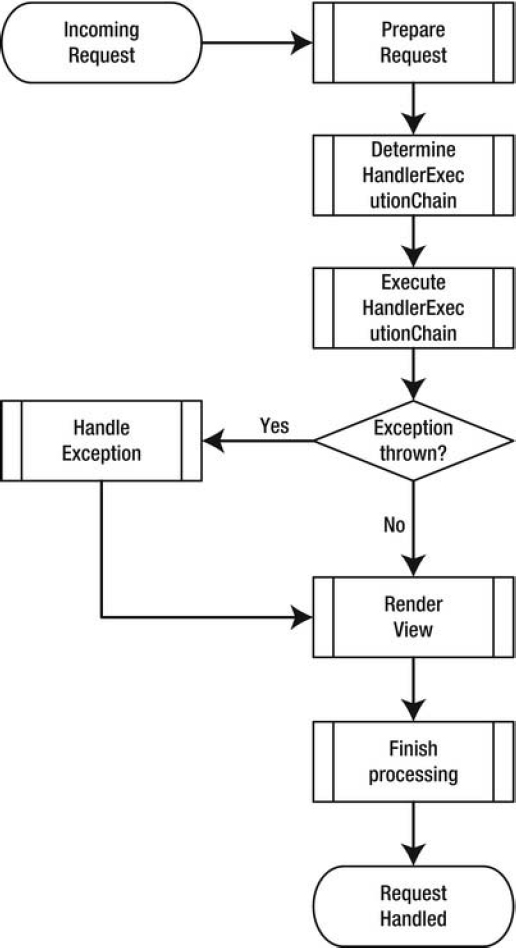
2. The locale resolver is bound to the request to enable elements in the process to resolve the locale to use when processing the request (rendering the view, preparing data, and so on). If you do not need locale resolving, you do not need it.

3. The theme resolver is bound to the request to let elements such as views determine which theme to use. If you do not use themes, you can ignore it.

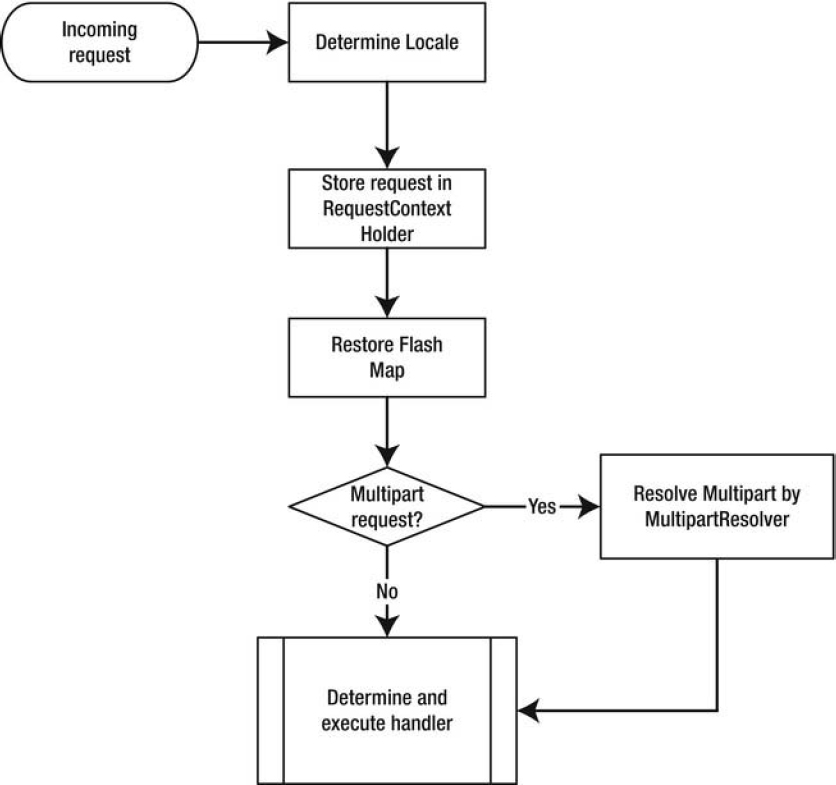
4. If you specify a multipart file resolver, the request is inspected for multiparts; if multiparts are found, the request is wrapped in a MultipartHttpServletRequest for further processing by other elements in the process.

5. An appropriate handler is searched for. If a handler is found, the execution chain associated with the handler (preprocessors, postprocessors, and controllers) is executed in order to prepare a model or rendering.

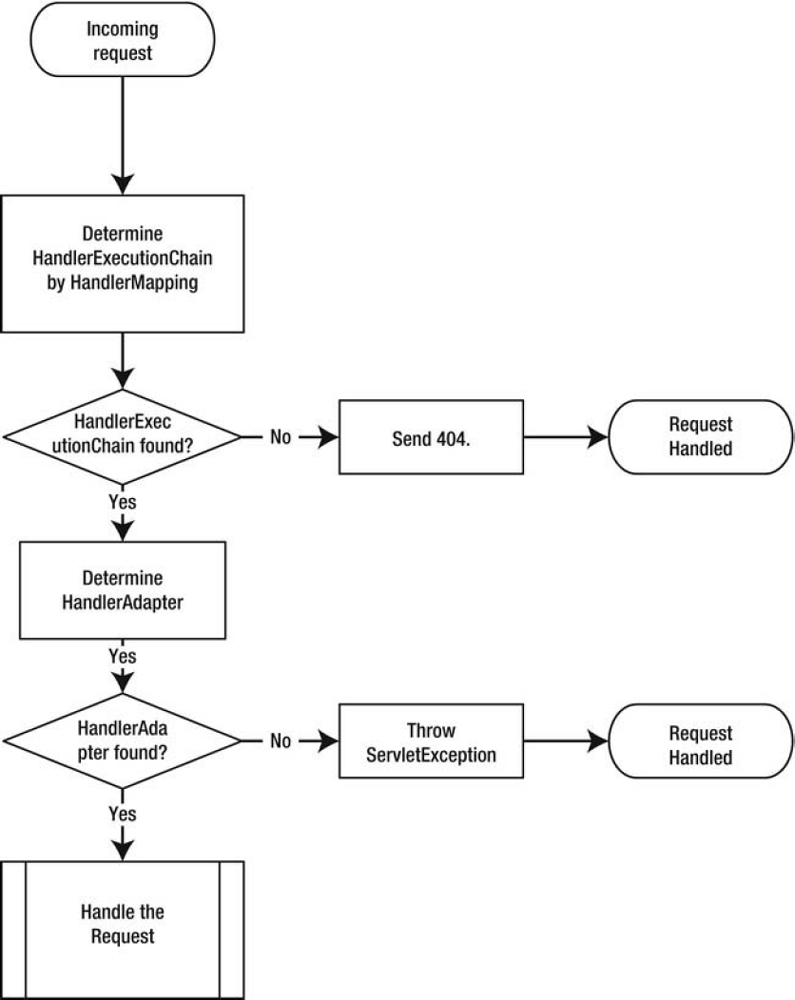
6. If a model is returned, the view is rendered. If no model is returned, (may be due to a preprocessor or postprocessor intercepting the request, perhaps for security reasons), no view is rendered, because the request could already have been fulfilled.



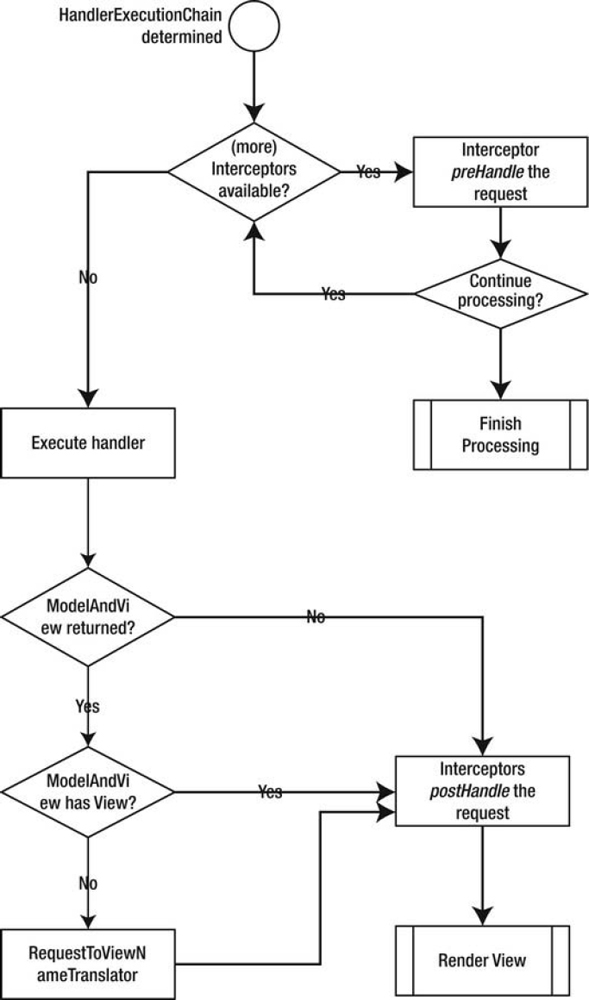
**Prepare a Request**



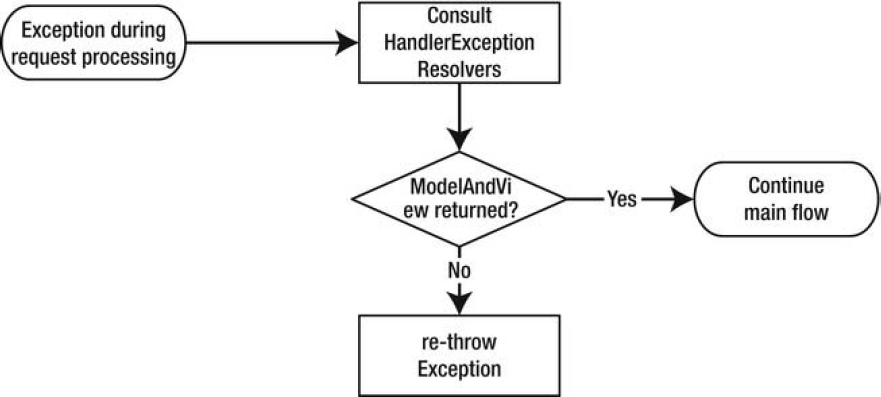
**Determine the HandlerExecutionChain**



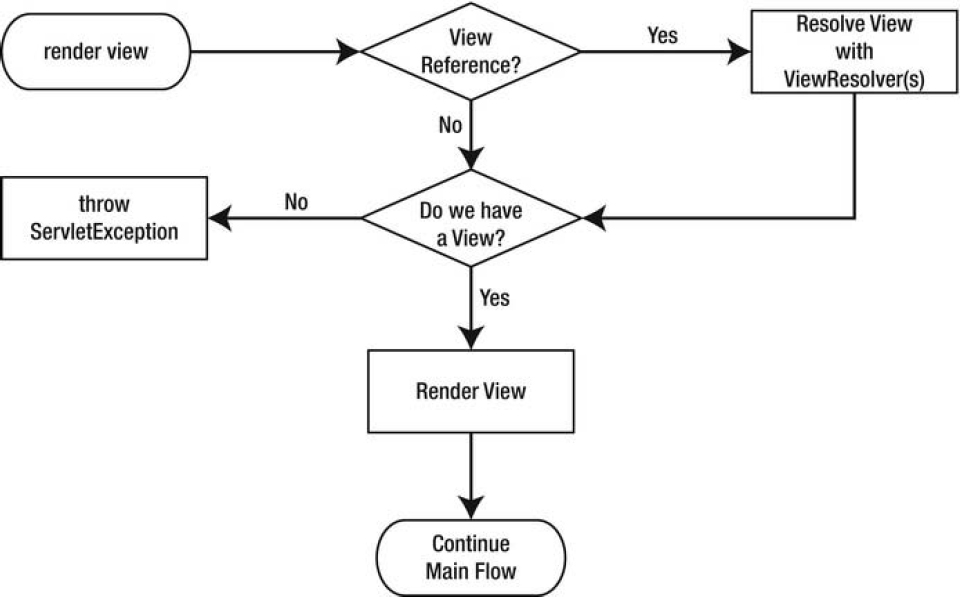
**Execute the HandlerExecutionChain**



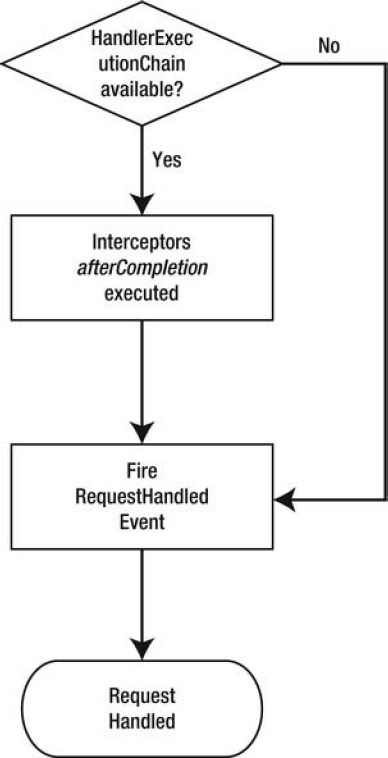
**Handle Exceptions**



**Render a View**



**Finish the Processing**



|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Dispatcher Servlet Properties**

| **Property** | **Default** | **Description** |
| --- | --- | --- |
| cleanupAfterInclude | true | Indicates whether to clean up the request attributes after an include request. In general, the default suffices, and this property should only be set to false in special cases. |
| contextAttribute | null | Stores the application context for this servlet. This is useful if the application context is created by some means other than the servlet itself. |
| contextClass | XmlWebApplicationContext | Configures the type of org.springframework.web. context.WebApplicationContext to be constructed by the servlet (it needs a default constructor). Will be configured using the given contextConfig Location. This isn't needed if you pass in an application context by using the constructor. |
| contextConfigLocation | [servlet-name]-servlet.xml | Indicates the location of the configuration files for the specified application context class. |
| contextId | null | Provides the Id of the application context. For example, this is used when the context is logged or sent toSystem.out. |
| contextInitializers contextInitializerClasses | null | Use the optional org.springframework.context. ApplicationContextInitializer classes to do some initialization logic for the application context, such as activating a certain profile. |
| detectAllHandlerAdapters | true | Detects all org.springframework.web.servlet.HandlerAdapters that can be detected from the application context. When set to false, a single one is detected by using the name, handlerAdapter. |
| detectAllHandlerException Resolvers | true | Detects all org.springframework.web. servlet.HandlerExceptionResolvers from the application context. When set to false, a single one is detected by using the name, handlerExceptionResolver. |
| detectAllHandlerMappings | true | Detects all org.springframework.web. servlet.HandlerMappings from the application context. When set to false, a single one is detected by using the name, handlerMapping. |
| detectAllViewResolvers | true | Detects all org.springframework.web. servlet.ViewResolvers from the application context. When set to false, a single one is detected by using the name, viewResolver. |
| dispatchOptionsRequest | false | Indicates whether to handle HTTP OPTIONS requests. The default is false; when set to true, you can also handle HTTP OPTIONS requests. |
| dispatchTraceRequest | false | Indicates whether to handle HTTP TRACE requests. The default is false; when set to true, you can also handle HTTP TRACE requests. |
| environment | org.springframework.  web.context.support.  StandardServlet  Environment | Configures the org.springframework .core.env.Environment to use for this servlet. The environment specifies which profile is active and can hold properties specific for this environment. |
| namespace | [servletname]-servlet | Use this namespace to configure the application context |
| publishContext | true | Indicates whether the servlet's application context is being published to thejavax.servlet.ServletContext. For production, it is recommended that you set this to false. |
| publishEvents | true | Indicates whether to fire after request processing an org.springframework.web.context. support.ServletRequestHandledEvent. You can use an org.springframework.context. ApplicationListener to receive these events. |
| threadContextInheritable | false | Indicates whether to expose the LocaleContext and RequestAttributes to child threads created from the request handling thread. |

**The DispatcherServlet's Default Components**

| **Component** | **Default implementation(s)** |
| --- | --- |
| MultipartResolver | No default, explicit configuration is required. |
| LocaleResolver | org.springframework.web.servlet.i18n.  AcceptHeaderLocaleResolver |
| ThemeResolver | org.springframework.web.servlet.theme.  FixedThemeResolver |
| HandlerMapping | org.springframework.web.servlet.handler.  BeanNameUrlHandlerMapping |
|  | org.springframework.web.servlet.mvc.annotation.  DefaultAnnotationHandlerMapping |
| HandlerAdapter | org.springframework.web.servlet.mvc.  HttpRequestHandlerAdapter |
|  | org.springframework.web.servlet.mvc.  SimpleControllerHandlerAdapter |
|  | org.springframework.web.servlet.mvc.annotation.  AnnotationMethodHandlerAdapter |
| HandlerExceptionResolver | org.springframework.web.servlet.mvc.annotation.  AnnotationMethodHandlerExceptionResolver |
|  | org.springframework.web.servlet.mvc.annotation.  ResponseStatusExceptionResolver |
|  | org.springframework.web.servlet.mvc.support.  DefaultHandlerExceptionResolver |
| RequestToViewNameTranslator | org.springframework.web.servlet.view.  DefaultRequestToViewNameTranslator |
| ViewResolver | org.springframework.web.servlet.view.  InternalResourceViewResolver |
| FlashMapManager | org.springframework.web.servlet.support.  SessionFlashMapManager |

General defaults

RequestMappingHandlerMapping, SimpleUrlHandlerMapping, BeanNameUrlHandlerMapping, viewControllerHandlerMapping

HttpRequestHandlerAdapter, SimpleControllerHandlerAdapter, RequestMappingHandlerAdapter,

ParameterizableViewController, UrlFilenameViewController, InternalResourceViewResolver