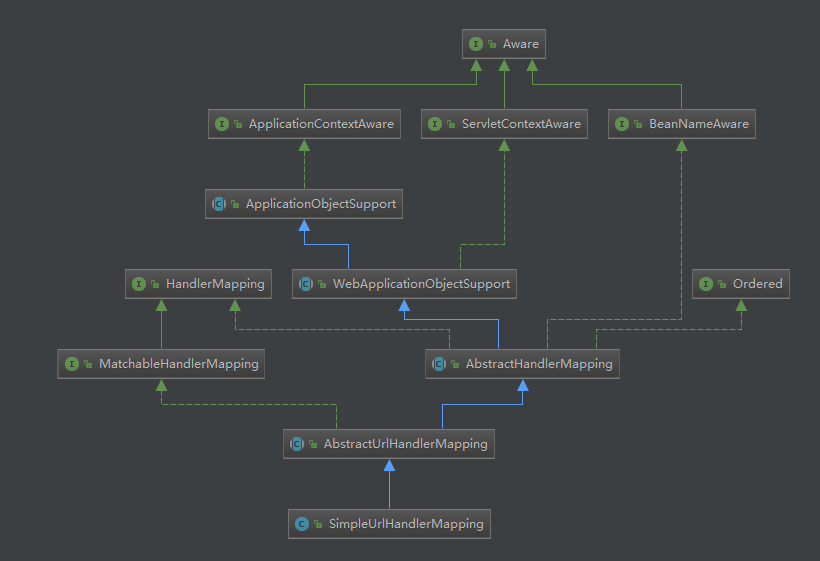
**Flow：**

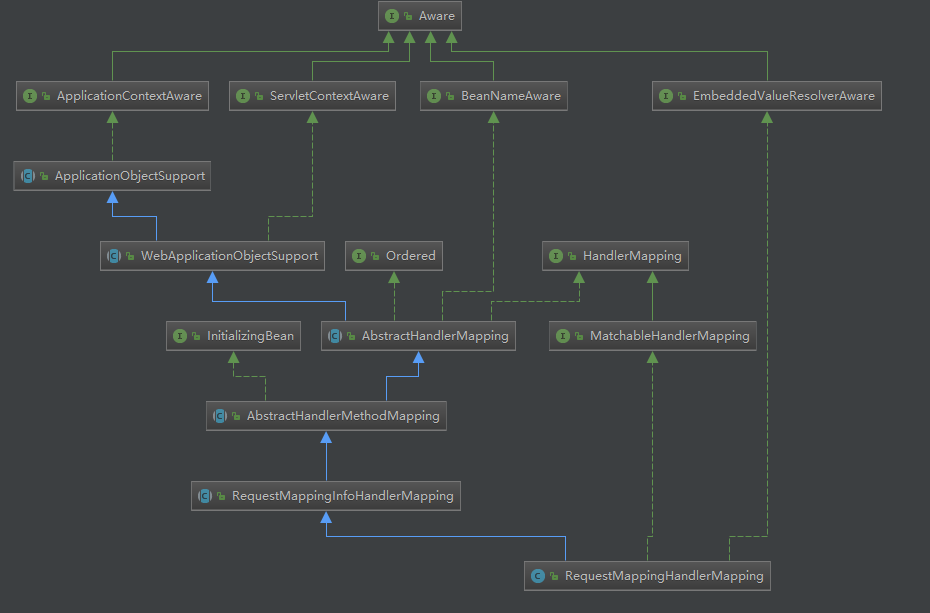
1. org.springframework.web.servlet.DispatcherServlet#doDispatch 请求进入
2. mv = ha.handle(processedRequest, response, mappedHandler.getHandler()); 反射调用controller方法拿到返回值，封装成对应类型，如果是某些非view类型，就直接调用HttpMessageConverter处理
3. processDispatchResult(processedRequest, response, mappedHandler, mv, dispatchException); 有view在这里render

**org.springframework.web.servlet.DispatcherServlet#getHandler 遍历所有Handler，如果不满足条件，则返回null，最外层，如果是null，则继续遍历，直到handler有值。**

SimpleUrlHandlerMapping：

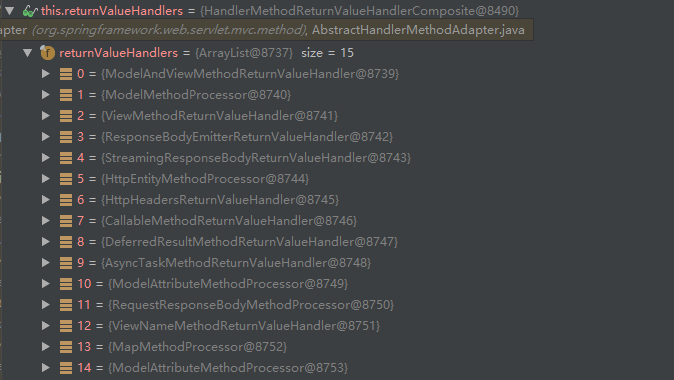


RequestMappingHandlerMapping：



HandlerMethodReturnValueHandler

ServletInvocableHandlerMethod.returnValueHandlers：



**HandlerMethodReturnValueHandlerComposite.selectHandler选择返回值处理器**

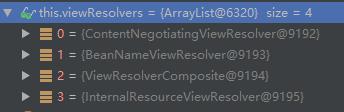
**org.springframework.web.method.support.HandlerMethodReturnValueHandler#handleReturnValue 每个返回值处理器处理返回值，封装成view，viewandmodel，还是通过HttpMessageConverter输出等。**

通过supportsReturnType进行范围类型的判断，从而决定使用哪个HandlerMethodReturnValueHandler

RequestResponseBodyMethodProcessor中supportsReturnType对ResponseBody.class进行判断，类或者方法上是否有@ResponseBody注解

ViewNameMethodReturnValueHandler中supportsReturnType对返回类型进行判断，如果是String，则是viewname

DispatcherServlet.viewResolvers：



UrlBasedViewResolver.buildView view.setUrl(getPrefix() + viewName + getSuffix());

UrlBasedViewResolver.viewClass 决定了使用什么类去构建View

Jsp是InternalResourceView

lookupPath=/abc

AbstractUrlHandlerMapping.handlerMap =>

key = "/\*\*/favicon.ico"

value = {ResourceHttpRequestHandler@8248} "ResourceHttpRequestHandler [class path resource [META-INF/resources/], class path resource [resources/], class path resource [static/], class path resource [public/], ServletContext resource [/], class path resource []]"

AbstractHandlerMethodMapping.MappingRegistry.urlLookup /abc => /abc get请求类型

AbstractHandlerMethodMapping.MappingRegistry.mappingLookup /abc => 类.方法

ServletInvocableHandlerMethod.returnValueHandlers

HandlerMethodReturnValueHandlerComposite.selectHandler 定义返回值处理

SimpleUrlHandlerMapping extends AbstractUrlHandlerMapping => getHandlerInternal

RequestMappingHandlerMapping extends AbstractHandlerMethodMapping => getHandlerInternal

AbstractHandlerMapping.getHandler

DispatcherServlet.doDispatch

mv = ha.handle(processedRequest, response, mappedHandler.getHandler());

InternalResourceView

DispatcherServlet.viewResolvers

UrlBasedViewResolver.loadView

UrlBasedViewResolver.buildView view.setUrl(getPrefix() + viewName + getSuffix());

UrlBasedViewResolver.viewClass

核心：

1.

// Actually invoke the handler.

mv = ha.handle(processedRequest, response, mappedHandler.getHandler());

2.

processDispatchResult(processedRequest, response, mappedHandler, mv, dispatchException);