DispatcherServlet

http://jinnianshilongnian.iteye.co...

- 1、Controller:处理器/页面控制器,做的是MVC中的C的事情,但控制逻辑转移到前端控制器了,用于对请求进行处理;
- 2、HandlerMapping:请求到处理器的映射,如果映射成功返回一个HandlerExecutionChain对象(包含一个Handler处理器(页面控制器)对象、多个HandlerInterceptor拦截器)对象;如BeanNameUrlHandlerMapping将URL与Bean名字映射,映射成功的Bean就是此处的处理器;
- 3、HandlerAdapter: HandlerAdapter将会把处理器包装为适配器,从而支持多种类型的处理器,即适配器设计模式的应用,从而很容易支持很多类型的处理器;如SimpleControllerHandlerAdapter将对实现了Controller接口的Bean进行适配,并且掉处理器的handleRequest方法进行功能处理;
- 4、ViewResolver: ViewResolver将把逻辑视图名解析为具体的View,通过这种策略模式,很容易更换其他视图技术;如InternalResourceViewResolver将逻辑视图名映射为jsp视图;
- 5、LocalResover:本地化解析,因为Spring支持国际化,因此LocalResover解析客户端的Locale信息从而方便进行国际化;
- 6、ThemeResovler:主题解析,通过它来实现一个页面多套风格,即常见的类似于软件皮肤效果;
- 7、MultipartResolver:文件上传解析,用于支持文件上传;
- 8、HandlerExceptionResolver:处理器异常解析,可以将异常映射到相应的统一错误界面,从而显示用户友好的界面(而不是给用户看到具体的错误信息);
- 9、RequestToViewNameTranslator:当处理器没有返回逻辑视图名等相关信息时,自动将请求URL映射为逻辑视图名;
- 10、FlashMapManager:用于管理FlashMap的策略接口,FlashMap用于存储一个请求的输出,当进入另一个请求时作为该请求的输入,通常用于重定向场景,后边会细述。

紧接着,容器创建一个ServletContext(application),这个WEB项目所有部分都将共享这个上下文。

容器以<context-param></context-param>的name作为键,value作为值,将其转化为键值对,存入ServletContext。

web.xml加载过程

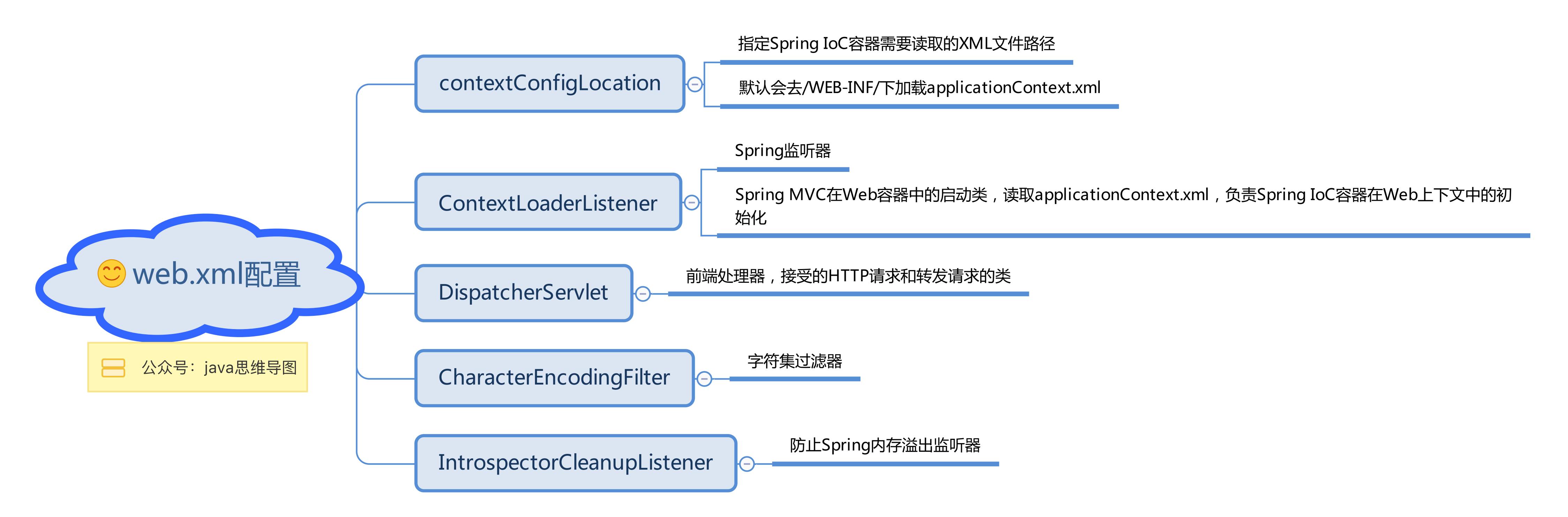
一 公众号: java思维导图

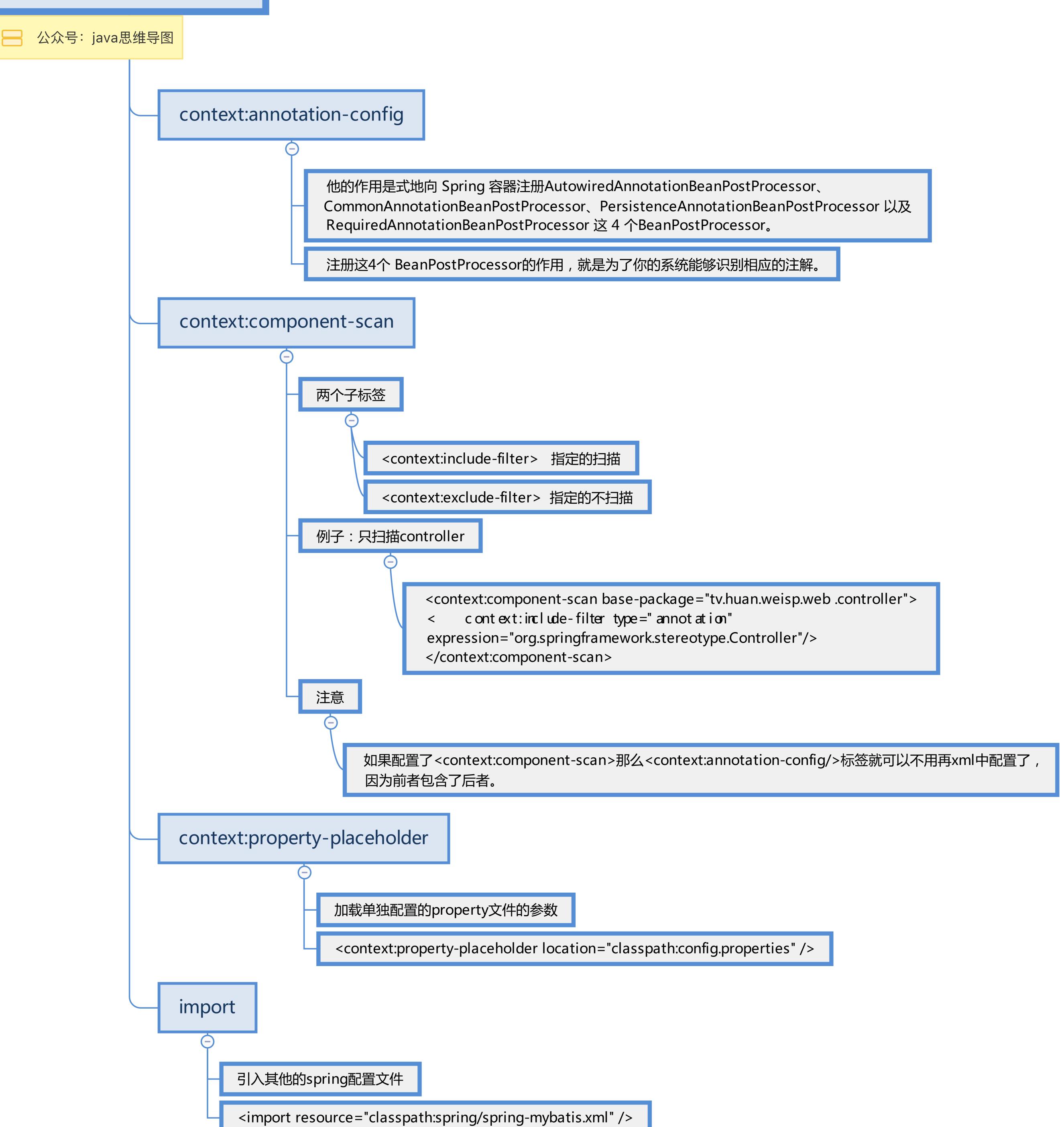
容器创建 < listener > < / listener > 中的类实例,根据配置的class类路径 < listener - class > 来创建监听,在监听中 会有contextInitialized(ServletContextEvent args)初始化方法,启动Web应用时,系统调用Listener的该方 法,在这个方法中获得:

ServletContext application = ServletContextEvent.getServletContext(); context-param的值= application.getInitParameter("context-param的键"); 得到这个context-param的值之后,你就可以做一些操作了。

举例:你可能想在项目启动之前就打开数据库,那么这里就可以在<context-param>中设置数据库的连接方式(驱动、url、user、password),在监听类中初始化数据库的连接。这个监听是自己写的一个类,除了初始化方法,它还有销毁方法,用于关闭应用前释放资源。比如:说数据库连接的关闭,此时,调用contextDestroyed(ServletContextEvent args),关闭Web应用时,系统调用Listener的该方法。

5 接着,容器会读取<filter></filter>,根据指定的类路径来实例化过滤器。





特殊bean

一 公众号: java思维导图

