Spring MVC 在eFID系統中的配置

Spring MVC需要在加載的時候對請求進行監控，這個工作是有一個servlet來負責的，所以在web.xml中需要對該servlet進行配置。

<servlet>

<servlet-name>iPLM SpringFramework Servlet</servlet-name>

<servlet-class>

org.springframework.web.servlet.DispatcherServlet

</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/context/spring.plugins.\*.xml</param-value>

</init-param>

</servlet>

<servlet-mapping>

<servlet-name>iPLM SpringFramework Servlet</servlet-name>

<url-pattern>\*.do</url-pattern>

</servlet-mapping>

**DispatcherServlet** 是Spring MVC 中负责请求调度的核心引擎，所有的请求将

由此Servlet 根据配置分发至各个逻辑处理单元。其内部同时也维护了一个

ApplicationContext实例

我们在**<init-param>**节点中配置了名为“**contextConfigLocation**”的

Servlet参数，此参数指定了Spring配置文件的位置“**/WEB-INF/Config.xml**”。

如果忽略此设定，则默认为“**/WEB-INF/<servlet name>-servlet.xml**”，其

中**<servlet name>**以Servlet 名替换（在当前环境下，默认值也就是

“**/WEB-INF/Dispatcher-servlet.xml**）。

我们将所有以.do结尾的请求交给Spring MVC进行处理。当然，也可以设为其他值，

如.action、.action等。

為了保證所有的配置文件都能被載入，在web.xml中配置了上下文載入器，該文件除了講dispatcherservlet載入到配置文件之外，其他的上下文配置文件也被載入。最常用的上下文載入器是一個servlet，名為：contextLoaderListener

<listener>

<listener-class>

org.springframework.web.context.ContextLoaderListener

</listener-class>

</listener>

配置了之後，如果沒有指定，則上下文載入器會在/WEB-INF/applicationContext.xm中找spring的配置文件。但是這樣無法講應用上下文分散到應用系統的個層，所以我們要取代這種默認方式。

通過在servlet上下文中設置contextConfigLocation參數來為上下文載入器配置一個或多個spring配置文件。

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/context/spring.\*.beans.xml</param-value>

</context-param>

這樣就講所有的beans.xml都載入了。

(web在啟動的時候，需要講所有的上下文配置文件都加載進來)

首先在init-param中載入了spring.plugin.jbpm.xml，該配置文件裏定義了一些基本的bean。首先定義了數據源dataSource，然後通過數據源定義hibernate的sessionbean，對數據庫中的表進行操作。然後是對工作流進行配置（此部分還不是很懂）

其次，比較重要的是spring.common.beans.xml，該文件中定義了許多與系統配置有關的文件。首先導入spring.properties資源文件，spring.common.view.xml導入了視圖解析器，spring.common.theme.xml是主題文件，spring.common.db.xml是數據庫配置文件，spring.common.mapping.xml是地址映射文件,spring.common.menu.xml是菜單文件，sprng.common.security.xml是安全機制文件等。。。。。。。。。。

在spring.common.view.xml文件中，定義了四種視圖解析器。

InternalResourceViewResolver 將邏輯視圖名解析為一個用模板分析(如jsp和Velocity模板)渲染的視圖對象，他只是簡單的講請求分發肺jsp來完成實際的渲染工作。由於網頁中用到JSTL標籤，，所以可以選用jstlView來替換InternalResourceView，及設置期viewclass為org.springframework.web.servlet.view.JstlView,這樣可以利用jstl實現國際化。

BeanNameViewResolver是一個講邏輯試圖名與應用上下文中的bean匹配的視圖解析器。

視圖解析器定義好之後，需要定義控制器，控制器是用來處理請求的