1. 引言
2. 介绍
   1. 主题解析器(Theme resolver):

能够解析你的web应用所使用的主题，以提供个性化的布局。

* 1. 使用：

<bean id="themeSource" class="org.springframework.ui.context.support.ResourceBundleThemeSource">

<property name="basenamePrefix" value="themes/theme-" />

</bean>

<bean id="themeResolver" class="org.springframework.web.servlet.theme.CookieThemeResolver">

<property name="defaultThemeName" value="blue"/>

</bean>

<mvc:interceptors>

<bean class="org.springframework.web.servlet.theme.ThemeChangeInterceptor">

<property name="paramName" value="theme" />

</bean>

</mvc:interceptors>

1. 源码解析：
   1. 初始化ThemeResolver,如果未配置ThemeResolver,则使用默认的FixedThemeResolver

DispatcherServlet. onRefresh(…);

|\_ initStrategies(…)

|\_ initThemeResolver(…)

* 1. 执行过程处理：

DispatcherServlet. doService(…)

request.setAttribute(WEB\_APPLICATION\_CONTEXT\_ATTRIBUTE, getWebApplicationContext());

request.setAttribute(LOCALE\_RESOLVER\_ATTRIBUTE, this.localeResolver);

request.setAttribute(THEME\_RESOLVER\_ATTRIBUTE, this.themeResolver);

request.setAttribute(THEME\_SOURCE\_ATTRIBUTE, getThemeSource());