8.7.1　组件排序策略

HandlerMapping、HandlerAdapter、HandlerExceptionResolver 和 ViewResolver 这 4 个组件是通过列表形式提供的，在这几个列表组件的使用时，是通过列表顺序遍历全部组件，且最终可处理目标的组件只能是组件列表中的一个组件，所以这个列表中组件的顺序尤为重要。而在初始化过程中，从应用上下文中获取组件列表后经过了一个排序，那么这个排序策略是如何实现的，即如何确定 Bean 在列表中的顺序？排序策略如下。

（1）如果一个 Bean 实现了 org.springframework.core.Ordered 接口，则直接通过接口的 getOrder 方法获取顺序数字，数字越小排序越靠前。

（2）如果在第（1）步获取不到顺序时，则通过对应组件类型上的 org.springframework.core.annotation.Order 注解中的 value 值获取 order。

（3）如果在第（2）步中仍无法获取顺序时，可以通过 javax.annotation.Priority 注解中的 value 获取 order。

如果需要把自定义的组件注册到 DispatcherServlet 的组件列表中时，可以通过这种方式控制组件遍历的顺序，即组件的优先级。

8.7.2　资源包

在 ThemeSource 与 MessageSource 两种源中，都提供了 Locale 区域支持资源包，该资源包是通过 Java 标准中用于实现国际化的标准机制而实现的，即基于 Resource Bundle 资源包的不同形式提供不同 Locale 对应的资源文件名。只需在资源文件名的名称后面添加不同的 Locale 后缀即可，标准形式为：资源名_Locale.properties。

ResourceBundle 类中提供了 getBundle 的静态方法，可以传入资源名与 Locale 参数，获取对应的 ResourceBundle 实例。ResourceBundle 实例类似于 Map 的数据结构，可以通过其中的方法根据 key 获取 value。

通过类型 PropertyResourceBundle 来支持.properties 格式的资源文件。除此之外，默认的 ResourceBundle 还支持 class 类型的资源类。可以通过“类名_Locale.class”形式提供属性名与属性值。要求这种类必须继承于 ResourceBundle，获取这种类型的资源时，返回的是这种类型的实例，通过返回的 ResourceBundle 实例中提供的方法，根据 key 获取 value。

如 JDK 中的 CalendarData，就是通过类型提供资源类的实例，还包括 CalendarDataenUS 类，即其他不同 Locale 对应的类，所有的类都继承于 ListResourceBundle，通过 getContents 方法获取二维数组，二维数组中每个元素都是 key 对应 value 的形式。

同时对于 Locale 区域这种类型，除了上面提到了支持语言和地区标识外，还支持更多信息，详细定义格式为：language-script-country-variant-extension-privateuse，各部分定义如下。

- language：这部分就是 ISO639 规定的代码，代表不同国家的语言，例如中文是 zh。