# SpringMvc框架

## 常用的HttpMessageConverter

|  |
| --- |
| ByteArrayHttpMessageConverter converts byte arrays. StringHttpMessageConverter converts strings. ResourceHttpMessageConverter converts to/from org.springframework.core.io.Resource for all media types. SourceHttpMessageConverter converts to/from a javax.xml.transform.Source. FormHttpMessageConverter converts form data to/from a MultiValueMap<String, String>. Jaxb2RootElementHttpMessageConverter converts Java objects to/from XML — added if JAXB2 is present on the classpath. MappingJacksonHttpMessageConverter converts to/from JSON — added if Jackson is present on the classpath. AtomFeedHttpMessageConverter converts Atom feeds — added if Rome is present on the classpath. RssChannelHttpMessageConverter converts RSS feeds — added if Rome is present on the classpath.  ByteArrayHttpMessageConverter: 负责读取二进制格式的数据和写出二进制格式的数据；  StringHttpMessageConverter：负责读取字符串格式的数据和写出二进制格式的数据；  ResourceHttpMessageConverter：负责读取资源文件和写出资源文件数据；  FormHttpMessageConverter：负责读取form提交的数据（能读取的数据格式为 application/x-www-form-urlencoded，不能读取multipart/form-data格式数据）；负责写入application/x-www-from-urlencoded和multipart/form-data格式的数据；  MappingJacksonHttpMessageConverter: 负责读取和写入json格式的数据；  SouceHttpMessageConverter：负责读取和写入 xml 中javax.xml.transform.Source定义的数据；  Jaxb2RootElementHttpMessageConverter：负责读取和写入xml 标签格式的数据；  AtomFeedHttpMessageConverter：负责读取和写入Atom格式的数据；  RssChannelHttpMessageConverter：负责读取和写入RSS格式的数据； |

## [spring mvc 注解配置StringHttpMessageConverter不生效](http://blog.csdn.net/sen19910708/article/details/49933213)

### 方法一

当使用@ResponseBody返回的字符串带有中文时，返回类型为String会被StringHttpMessageConverter处理，当时查看源码发现默认的Charset DEFAULT\_CHARSET使用的是ISO-8859-1，这里除了修改配置外，可以使用覆盖类的方式来修改这个类的源码，即在自己的项目下新建org.springframework.http.converter包，然后新建类StringHttpMessageConverter，修改里面的

**public** **static** **final** Charset DEFAULT\_CHARSET = Charset.forName("utf-8");

这样当项目启动后，工程的类会优先被加载，这样可以达到我们想要的效果。

### 方法二

 另外一种方式就是直接配置DefaultAnnotationHandlerMapping和AnnotationMethodHandlerAdapter这两个类，当我们使用 mvc:annotation-driven 这个标签时，其实也是来配置这两个类的，直接采用下面的配置

|  |
| --- |
| 1. <!-- 手动配置Springmvc注解驱动 --> 2. **<bean** class="org.springframework.web.servlet.mvc.annotation.DefaultAnnotationHandlerMapping"**/>** 4. <!-- 启动Spring MVC的注解功能,设置编码方式，防止乱码 --> 5. **<bean** class="org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter"**>** 6. **<property** name="messageConverters"**>** 7. **<list>** 8. **<bean** class="org.springframework.http.converter.ByteArrayHttpMessageConverter" **/>** 9. **<bean** class = "org.springframework.http.converter.StringHttpMessageConverter"**>** 10. **<property** name="supportedMediaTypes" value="text/plain;charset=UTF-8" **/>** 11. **</bean>** 12. **<bean** class="org.springframework.http.converter.json.MappingJackson2HttpMessageConverter"**>** 13. **<property** name="supportedMediaTypes"**>** 14. **<list>** 15. **<value>**text/html;charset=UTF-8**</value>** 16. **</list>** 17. **</property>** 18. **</bean>** 19. **</list>** 20. **</property>** 21. **</bean>** |