# Spring Boot核心-Web配置

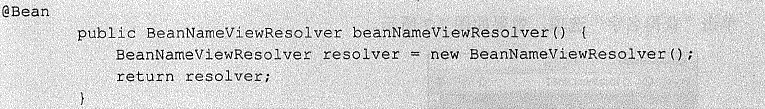
## Spring Boot 提供的自动配置

分析WebMvcAutoConfiguration 及WebMvcProperties 的源码，发现Spring Boot为我们提供了如下自动配置功能。

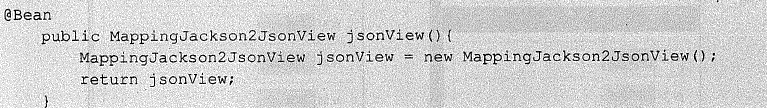
## 自动配置ViewResolver

1. ContentNegotiatingViewResolver：这是一个特殊的ViewResolver, ContentNegotiatingViewResolver 不自己处理View ，而是代理给不同的ViewResolver 来处理，它有最高的优先级。
2. BeanNameViewResolver：根据Bean的名称来解析视图，如控制器（ @Controller ）中的一个方法的返回值的字符串（视图名）为X，BeanNameView Resolver 去查找名为X的View 来渲染视图。、

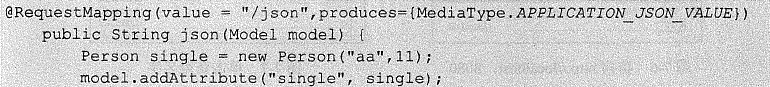
定义BeanNameView Resolver 的Bean:



定义一个View 的Bean ，名称为jsonView



在控制器中，返回值为字符串jsonView ，它会找Bean 的名称为jsonView 的视图来渲染：





1. IntemalResource View Resolver:

这个是一个极为常用的ViewResolver，主要通过设置前缀、后缀，以及控制器中方法来返回视图名的字符串，以得到实际页面， Spring Boot 的源码如下：

|  |
| --- |
| @Bean @ConditionalOnMissingBean **public** InternalResourceViewResolver defaultViewResolver() {  InternalResourceViewResolver resolver = **new** InternalResourceViewResolver();  resolver.setPrefix(**this**.**mvcProperties**.getView().getPrefix());  resolver.setSuffix(**this**.**mvcProperties**.getView().getSuffix());  **return** resolver; } |

## 自动配置的静态资源

在自动自己置类的addResourceHandlers 方法中定义了以下静态资源的自动配置。

( 1) 类路径文件：把类路径下的/static 、/public 、/resources 和/META-INF/resources 文件夹下的静态文件直接映射为/\*\*，可以通过<http://localhost:8080>/\*\*来访问。

(2) webjar：webjar 就是将我们常用的脚本框架封装在jar 包中的jar 包，把webjar 的/META-INF/resources/webjars/下的静态文件映射为/webjar/\*\*，可以通过http://localhost: 8080/webjar/\*\*来访问。

## 自动配置Formatter 和Converter

从WebMvcAutoConfiguration代码中可以看到，只要我们定义了Converter 、GenericConverter 和Formatter 接口的实现类的Bean, 这些 Bean 就会自动注册到Spring 容器中。

|  |
| --- |
| @Override **public void** addFormatters(FormatterRegistry registry) {  **for** (Converter<?, ?> converter : getBeansOfType(Converter.**class**)) {  registry.addConverter(converter);  }  **for** (GenericConverter converter : getBeansOfType(GenericConverter.**class**)) {  registry.addConverter(converter);  }  **for** (Formatter<?> formatter : getBeansOfType(Formatter.**class**)) {  registry.addFormatter(formatter);  } } |