# **十二、集成Springboot**

Axon框架为Spring提供了广泛的支持，但并不要求您使用Spring来使用Axon。所有组件都可以通过编程方式配置，并且不需要在类路径上使用Spring。但是，如果您确实使用Spring，那么使用Spring的注释支持可以使大部分配置变得更简单。Axon在此基础上提供了SpringBoot启动器，因此您也可以从自动配置中获益。

## **12.1 自动配置**

到目前为止，Axon的springboot自动配置是开始配置Axon组件的最简单的选择。通过简单地声明对axonspringbootstarter的依赖性，axon将自动配置基础结构组件（命令总线、事件总线、查询总线），以及运行和存储聚合和saga所需的任何组件。

## **12.2揭开Axon Springboot的神秘面纱**

由于很多事情都是在后台发生的，有时候很难理解注释或者仅仅包含一个依赖项是如何支持这么多特性的。

axon spring boot starter在构造启动器时遵循一般的spring boot约定。它依赖于axon spring boot autoconfigure，它包含axon自动配置的具体实现。当AxonSpringBoot应用程序启动时，它会在类路径中查spring factory。此文件位于axon spring boot自动配置模块的META-INF目录中。

  这个文件映射了AxonSpringBoot，应用程序将尝试应用的不同配置类。因此，根据这个片段，springboot将尝试为AxonServerAutoConfiguration、AxonAutoConfiguration应用所有的配置类。

是否应用这些配置类，将取决于在这些类上定义的条件：

AxonServerAutoConfiguration将Axon服务器配置为命令总线、查询总线和事件存储的实现。它将在AxonAutoConfiguration之前应用，并且仅当org.axonframework.axonserver.connector.axonServer配置类在类路径中可用。可以通过设置已启用axon.axonserver属性在.properties/.yml文件中设置为false。

AxonAutoConfiguration配置命令总线、查询总线、事件存储/事件总线和其他axon组件的“非axon服务器”实现。这些组件只有在不在Spring应用程序上下文中时才会被初始化，例如@ConditionalOnMissingBean(事件总线.class). 由于AxonServerAutoConfiguration将在AxonServer自动配置之后应用，这些Axon组件将已经在Spring应用程序上下文中，因此Axon服务器的命令总线、查询总线和事件存储/事件总线将会成功实现。

Axon Spring Boot自动配置不具侵入性。它将只定义尚未在应用程序上下文中显式定义的Spring组件。这允许您通过在一个@Configuration类中定义自己的bean来完全覆盖自动配置的bean。

具体的Axon（Spring）组件配置将在本指南的以下部分进行详细说明。

除此之外，Axon包还提供了其他很多工具框架的支持：