

Spring Boot Starter 文档总结

一、核心概念

Spring Boot Starter 是一组便捷的依赖描述符，提供"一站式"解决方案，聚合特定功能所需的所有依赖，无需手动复制粘贴大量依赖坐标。

二、官方 Starter 分类

分类	核心启动器示例
核心启动器	<code>spring-boot-starter</code> 、 <code>spring-boot-starter-test</code>
Web 开发	<code>spring-boot-starter-webmvc</code> 、 <code>spring-boot-starter-webflux</code>
数据访问	<code>spring-boot-starter-data-jpa</code> 、 <code>spring-boot-starter-data-redis</code>
安全认证	<code>spring-boot-starter-security</code> 、 <code>spring-boot-starter-security-oauth2-client</code>
消息驱动	<code>spring-boot-starter-amqp</code> 、 <code>spring-boot-starter-kafka</code>
生产就绪	<code>spring-boot-starter-actuator</code> 、 <code>spring-boot-starter-micrometer-metrics</code>
模板引擎	<code>spring-boot-starter-thymeleaf</code> 、 <code>spring-boot-starter-freemarker</code>
嵌入式容器	<code>spring-boot-starter-tomcat</code> 、 <code>spring-boot-starter-jetty</code>
测试启动器	<code>spring-boot-starter-test</code> (每个模块都有对应 <code>-test</code> 变体)

三、常用启动器及功能

启动器	核心功能
<code>spring-boot-starter</code>	自动配置、日志、YAML 解析 (基础依赖)
<code>spring-boot-starter-webmvc</code>	Spring MVC + Tomcat 构建 RESTful Web 应用
<code>spring-boot-starter-data-jpa</code>	JPA + Hibernate 数据库访问
<code>spring-boot-starter-security</code>	Spring Security 认证授权
<code>spring-boot-starter-test</code>	JUnit、Mockito、Hamcrest 测试框架
<code>spring-boot-starter-actuator</code>	生产监控、健康检查、指标收集
<code>spring-boot-starter-cache</code>	Spring 缓存抽象支持

四、官方 Starter 命名规范

类型	命名格式	示例
基础启动器	<code>spring-boot-starter</code>	<code>spring-boot-starter</code>
功能启动器	<code>spring-boot-starter-{模块}</code>	<code>spring-boot-starter-webmvc</code>
测试启动器	<code>spring-boot-starter-{模块}-test</code>	<code>spring-boot-starter-data-jpa-test</code>
响应式变体	<code>spring-boot-starter-{模块}-reactive</code>	<code>spring-boot-starter-data-mongodb-reactive</code>
容器替换	<code>spring-boot-starter-{容器名}</code>	<code>spring-boot-starter-jetty</code>

已弃用提醒：

- `spring-boot-starter-web` → `spring-boot-starter-webmvc`
- `spring-boot-starter-oauth2-client` → `spring-boot-starter-security-oauth2-client`

五、创建第三方 Starter

5.1 命名规范（社区公认）

模块类型	命名格式	示例
Starter 主模块	<code>{your-name}-spring-boot-starter</code>	<code>mycompany-spring-boot-starter</code>
自动配置模块	<code>{your-name}-spring-boot-autoconfigure</code>	<code>mycompany-spring-boot-autoconfigure</code>

 不要使用 `spring-boot-starter-{xxx}` 格式，该前缀保留给官方启动器。

5.2 核心组件

组件	作用
<code>MyServiceProperties</code>	配置属性类，使用 <code>@ConfigurationProperties</code>
<code>MyService</code>	服务类，提供业务功能
<code>MyServiceAutoConfiguration</code>	自动配置类，使用条件注解控制装配
<code>AutoConfiguration.imports</code>	元数据文件，位于 <code>META-INF/spring/</code> 目录

5.3 常用条件注解

注解	作用
@ConditionalOnClass	指定类存在于 classpath 时生效
@ConditionalOnMissingBean	容器中不存在指定 Bean 时生效
@ConditionalOnProperty	配置属性满足条件时生效
@ConditionalOnWebApplication	Web 应用时生效

5.4 用户使用方式

Maven 依赖：

xml

```
<dependency>
  <groupId>com.mycompany</groupId>
  <artifactId>mycompany-spring-boot-starter</artifactId>
  <version>1.0.0</version>
</dependency>
```

配置示例 (application.yml)：

yaml

```
mycompany:
  host: myserver.example.com
  port: 9090
  enabled: true
```