

封装了社区和 Spring 团队的最佳实践，提供合理的默认配置，适合大多数应用场景。

- + 减少了初学者的学习曲线，开发者无需了解底层实现细节即可快速使用。
- * 普通依赖：

+ 配置依赖时需要开发者自行决定各个依赖的配置方式，可能会导致不一致或不合理的配置。

- + 适合需要细粒度控制和定制化配置的场景。

4. **模块化**

* Spring Boot Starter：

+ 提供了功能模块化的方式，通过不同的 Starter 可以快速集成不同的功能模块。

- + 例如，使用 `spring-boot-starter-data-jpa` 可以快速集成 JPA 持久化，使用 `spring-boot-starter-security` 可以快速集成安全功能。

* 普通依赖：

+ 需要手动组合和配置不同的依赖，缺乏模块化的便捷性。

- + 不同的依赖之间可能存在兼容性问题，需要开发者自行解决。

三、为什么需要自定义 Starter?

在 Spring Boot 项目中，自定义 Starter 可以极大地简化和标准化特定功能的集成和配置。以下是一些需要使用自定义 Starter 的场景：

1. 重复使用的通用功能模块

当你有多个项目需要使用相同的功能模块时，可以创建一个自定义 Starter，将这些功能模块封装起来。这样可以减少代码重复，提高代码复用性。

示例：统一的日志处理模块、统一的异常处理模块等。

2. 复杂的第三方库集成

当集成第三方库时，如果需要复杂的配置和初始化，可以通过自定义 Starter 简化集成过程，提供更友好的使用方式。

****示例**：**集成 Kafka、Redis、Elasticsearch 等中间件。

3. 公司内部的标准化开发框架

在企业内部，可以将一些标准化的开发规范和基础设施封装成自定义 Starter，使得团队内的所有项目都能遵循相同的规范和使用相同的基础设施。

****示例**：**公司内部的监控系统、认证授权模块、消息队列封装等。

4. 统一配置和管理

当需要统一管理和配置某些功能时，可以通过自定义 Starter 进行集中管理，提供默认配置，并允许项目根据需要进行覆盖配置。

****示例**：**统一的数据库连接池配置、安全配置等。

5. 封装复杂的初始化逻辑

有些功能的初始化过程非常复杂，如果每个项目都需要重复这部分初始化逻辑，可以通过自定义 Starter 封装这部分逻辑，简化项目的初始化过程。

****示例**：**复杂的缓存管理、事务管理等。

6. 提供公共的 Bean 和配置

当需要在多个项目中提供相同的 Bean 和配置时，可以通过自定义 Starter 提供这些公共的 Bean。Boot 项目中引入自定义 Starter：

```
```
<dependency>
 <groupId>com.example</groupId>
 <artifactId>my-custom-starter</artifactId>
 <version>1.0.0</version>
</dependency>
```
```

```

并在项目中使用 `ExampleService`：

```
```
import com.example.starter.ExampleService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class ExampleController {

  @Autowired
  private ExampleService exampleService;

  @GetMapping("/hello")
  public String hello() {
    return exampleService.sayHello();
  }
}
```
```

```

五、总结

自定义 Spring Boot Starter 是一种简化项目配置和提高代码复用性的重要手段。通过封装复杂的配置逻辑和依赖关系，开发者可以更专注于业务逻辑，实现更高效的开发流程。希望本文的介绍和示例能够帮助你理解和实现自定义 Spring Boot Starter。

下一篇文章分享一个工作中使用自定义starter的案例，欢迎。

参考资料

* [Spring Boot 官方文档](<http://cxyroad.com/>
"https://docs.spring.io/spring-boot/docs/current/reference/html/")
* [Spring Boot Auto Configuration](<http://cxyroad.com/>
"https://docs.spring.io/spring-boot/docs/current/reference/html/boot-features-developing-auto-configuration.html")

通过这些资源，你可以进一步深入了解 Spring Boot Starter 的实现原理和最佳实践。

原文链接: <https://juejin.cn/post/7373867360019267622>