

9.1.2 优雅地替换自动装配

Spring Boot 官方文档的“16.1 Gradually Replacing Auto-configuration”一节中提到：

Auto-configuration is non-invasive. At any point, you can start to define your own configuration to replace specific parts of the auto-configuration. For example, if you add your own `DataSource` bean, the default embedded database support backs away.

以上文字大意是，Spring Boot 自动装配并非是侵占性的，开发人员可在任意一处定义配置类，从而覆盖那些被自动装配的组件。文档又举例说明，自定义的 `DataSource Bean` 能够覆盖默认的嵌入式数据库的数据源。结合目前已有的技术积累，可将其解读为 Spring Boot 优先解析自定义配置类，并且内建的自动装配配置类实际上为默认的条件配置，即一旦应用存在自定义实现，则不再将它们装配。其中的缘由将在“Spring Boot 自动装配原理”一节中深入讨论。

9.1.3 失效自动装配

Spring Boot 官方文档中的“16.2 Disabling Specific Auto-configuration Classes”继续介绍如何失效自动装配：

If you find that specific auto-configuration classes that you do not want are being applied, you can use the `exclude` attribute of `@EnableAutoConfiguration` to disable them, as shown in the following example:

```
import org.springframework.boot.autoconfigure.*;
import org.springframework.boot.autoconfigure.jdbc.*;
import org.springframework.context.annotation.*;

@Configuration
@EnableAutoConfiguration(exclude={DataSourceAutoConfiguration.class})
public class MyConfiguration {
}
```

If the class is not on the classpath, you can use the `excludeName` attribute of the annotation and specify the fully qualified name instead. Finally, you can also control the list of auto-configuration classes to exclude by using the `spring.autoconfigure.exclude` property.