

Please visit website: <http://cxyroad.com>

```
implements AuthCodeService {
    @Override
    public boolean check() {
        // 自定义认证逻辑
        return true;
    }
}

...
```

在上述代码中，`@Primary`注解确保了`CustomAuthCodeServiceImpl`会覆盖原有的实现。然而，这种方法有时可能不起作用，特别是当Bean是通过类路径扫描自动配置时。

方案四：自定义`@ComponentScan`排除规则

`@ComponentScan`注解可以用来指定Spring在启动时扫描哪些包，并排除某些特定的类。通过自定义排除规则，我们可以阻止Spring创建不需要的Bean。

```
...

@SpringBootApplication
@ComponentScan(basePackages = "com.yourcompany",
               excludeFilters = {@Filter(type =
FilterType.ASSIGNABLE_TYPE, classes = AuthCodeServiceImpl.class)})
public class Application {
    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}

...
```

3. 在`/META-INF/spring.factories`中创建配置文件，指定自定义的`ApplicationContextInitializer`。

```
...

org.springframework.context.ApplicationContextInitializer=\
com.yourcompany.context.CustomTypeExcludeFilterApplicationContextIn
```

initializer

...

通过上述步骤，我们可以有效地排除不需要的Bean，解决同名Bean冲突问题。

总结

--

以上提供了5中方案来解决Spring Boot中的同名Bean冲突，希望对你有帮助。
实际工作中可以根据具体的情况进行选择。

**另外，对GPT4.0有需求的同学，请移步[GPT4.0教程](<http://cxyroad.com/>
”<https://7we.cn/subscribe-chatgpt-plus-in-five-minutes/>)。 **

原文链接: <https://juejin.cn/post/7357142305424523318>