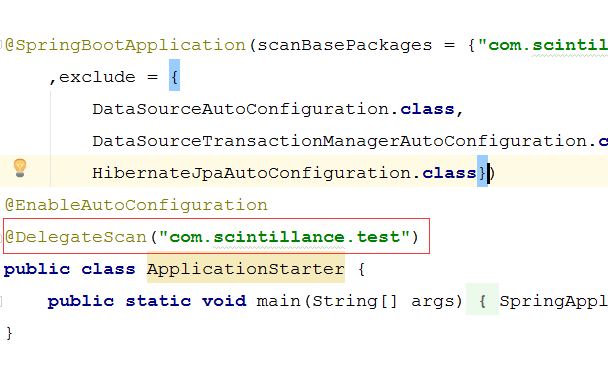
# 接口路由

考虑到同一个接口有多种实现，增加实现路由功能，实现根据指定字段，路由代理到对应实现类

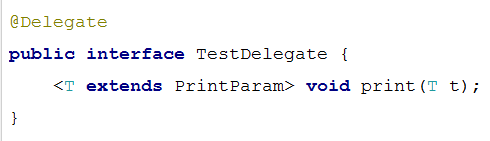
1. 注解说明

|  |  |
| --- | --- |
| DelegateScan | 开启路由功能，并指定扫描路径 |
| Delegate | 用于修饰接口类，自动生成代理 |
| DelegateElement | 修饰接口实现，配置key与实现映射 |
| DelegateReference | 用于注入接口，初始化bean时注入 |
| RouteProperty | 修饰入参中标识不同实现的字段 |

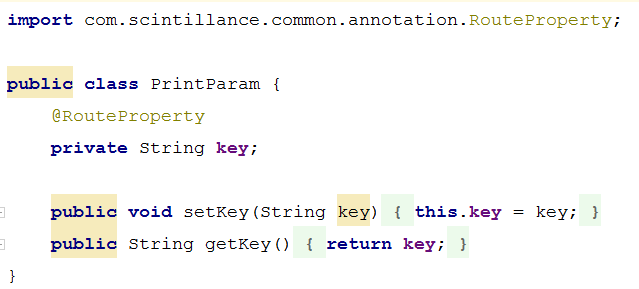
1. 样例
   1. 添加扫描注解



* 1. 定义接口

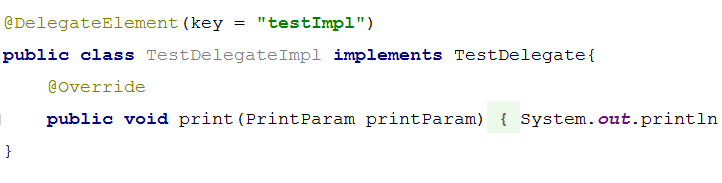


接口这里使用泛型方式，便于实现类书写；定义一个接口公共参数，并指定表示哪个实现的字段

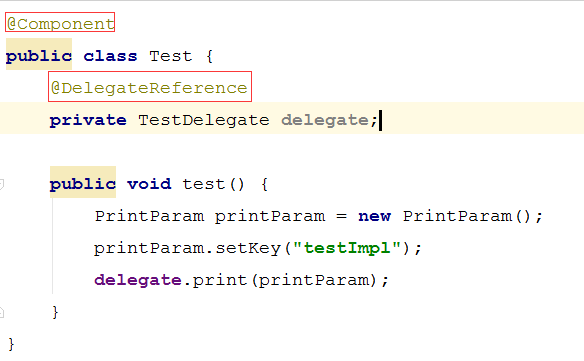


参数中Getter及Setter方法是必要的，可以用lombok注解

* 1. 定义实现

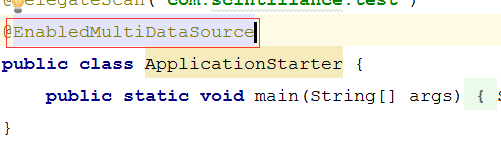


* 1. 注入实现并调用



# 多数据源

1. 开启多数据源注解



1. 配置项

spring.datasource.url=jdbc:mysql://localhost/test

spring.datasource.username=root

spring.datasource.password=123456

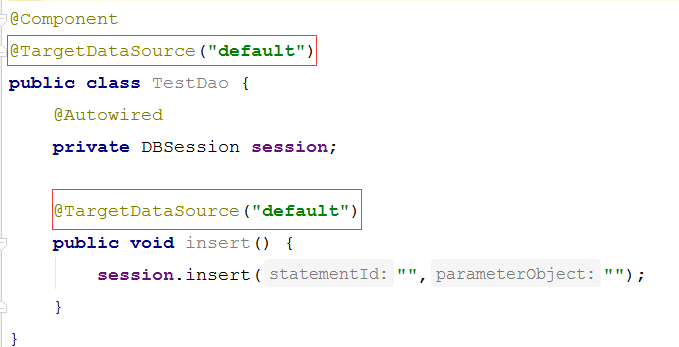
spring.datasource.driver-class-name=com.mysql.jdbc.Driver

spring.datasource.custom.dbname.url=…

前面一段是主数据源，默认数据源，后面动态配置自定义数据源(这一块还没实现…)

1. 指定@TargetDataSource

可以在方法或者类上指定当前方法使用的数据源



二选一，当冲突时，选方法上的，调用数据库基于DBSession来做