# 读写分离(annotation)说明文档

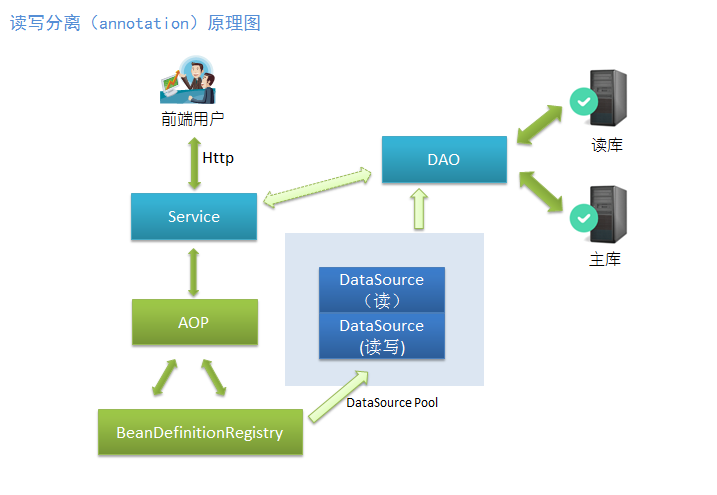
## 作用

实现系统数据持久层对数据库的读写分库操作

## 目标

让主数据库（master）处理事务性增、改、删操作（INSERT、UPDATE、DELETE），而从数据库（slave）处理SELECT查询操作。

## 原理



## 使用方式

添加在方法头部，例如：

1. 当不指定数据源类型时，默认使用主库

|  |
| --- |
| @DataSource  public void insert(){  //do something...  } |

1. 可以指定数据源类型为DataSourceType.WRITE，表示使用主库作为数据源

|  |
| --- |
| @DataSource(DataSourceType.WRITE)  public void insert(){  //do something...  } |

1. 当方法头部不指定数据源是，此方法默认使用主库

|  |
| --- |
| public void insert(){  //do something...  } |

1. 只有当指定DataSourceType的类型为READ时此方法所使用的数据源才切换成读库

|  |
| --- |
| @DataSource(DataSourceType.READ)  public void insert(){  //do something...  } |

## 代码位置

com.zgxcw.springboot.framework.datasource.DataSource（数据源注解入口）

com.zgxcw.springboot.framework.datasource.config.DataSourceAspect (注解实现类)

com.zgxcw.springboot.framework.datasource.config.DataSourceCluster (数据源类型持有者)

com.zgxcw.springboot.framework.datasource.config.DataSourceConfig(数据源参数加载)

com.zgxcw.springboot.framework.datasource.config.DataSourceDynamic(动态数据源bean)

com.zgxcw.springboot.framework.datasource.config.DataSourceDynamicContextHolder(动态数据源持有者)

com.zgxcw.springboot.framework.datasource.config.DataSourceDynamicRegister(动态数据源注册类)

com.zgxcw.springboot.framework.datasource.config. DataSourceHolder(数据源)

## 运行方式

运行入口com.zgxcw.springboot.demo.mybatis. Application.main(String[] args)