# mongodb说明文档

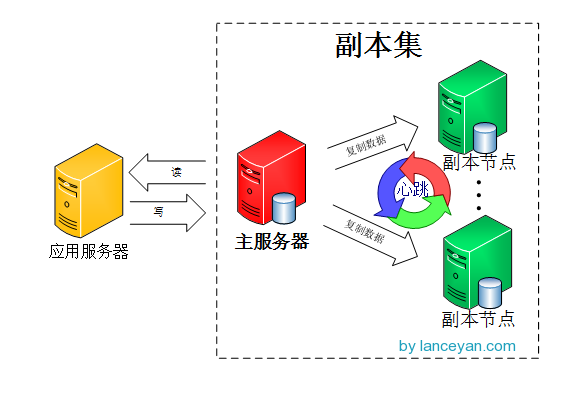
## 作用

对mongodb二次封装，支持对象缓存，并支持自定义集合对象缓存

## 目标

使用mongodb分片分布式集群管理服务层异常监控、平台监控

## 原理



## 使用方式

注入MongoService调用其中的方法实现数据的持久化，如：

|  |
| --- |
| @Autowired  **private** MongoService mongoService;  //构建查询条件  BasicDBObjectQuerySupport query = **new** BasicDBObjectQuerySupport();  query.$like("appName", mongoSystemException.getAppName())  .$like("modelName", mongoSystemException.getModelName())  .$like("className", mongoSystemException.getClassName())  .$like("methodName", mongoSystemException.getMethodName())  .$like("parameters", mongoSystemException.getParameters())  .$like("exceptionTrace", mongoSystemException.getExceptionTrace())  .$gteAndLte("exceptionTime", startTime, endTime);  //构建排序规则  BasicDBObjectOrderBySupport order = **new** BasicDBObjectOrderBySupport();  order.$orderBy("exceptionTime",OrderBy.***ASC***);  //查询结果  List<MongoSystemException> list = mongoService.query(MongoSystemException.**class**, query, order, page, limit); |

## 代码位置

|  |  |  |
| --- | --- | --- |
| com.zgxcw.springboot.framework.mongodb.config | BasicDBObjectOrderBySupport | 排序规则扩展类 |
| BasicDBObjectQuerySupport | BasicDBObject扩展模版类 |
| MongoAspect | AOP |
| MongoBuilder | 数据源参数封装类 |
| MongoConfig | mongodb配置 |
| MongoDataSourceCluster | 数据源 |
| MongoDataSourceHolder | mongodb数据源参数 |
| com.zgxcw.springboot.framework.mongodb.enums | Collection | Mongodb集合类型 |
| OrderBy | 排序枚举 |
| com.zgxcw.springboot.framework.mongodb.model | MongoSystemException | 系统异常实体 |
| com.zgxcw.springboot.framework.mongodb.service | MongoService | Mongodb服务接口 |
| com.zgxcw.springboot.framework.mongodb.service.imp | MongoServiceImpl | mongodb 接口实现 |

## API

MongoService

|  |  |
| --- | --- |
| public <T> String insert(T t); | 插入一条数据, 插入的集合目标为T |
| public <T> boolean insert(List<T> t); | 批量插入数据, 插入的集合目标为T |
| public <T> boolean insert(List<T> t); | 批量插入数据, 插入的集合目标为T |
| public <T> String insert(Collection collection,T t); | 插入一条数据 - 指定集合 |
| public <T> boolean insert(Collection collection,List<T> t); | 批量插入数据 - 指定集合 |
| public <T> boolean update(T t); | 根据\_id更新一条数据, 更新的集合目标为T |
| public <T> boolean update(Collection collection,T t); | 匹配更新一条数据-指定集合  如果在collection内已经存在一个和x对象相同的"\_id"的记录,  mongodb就会把x对象替换collection内已经存在的记录，否则将会插入x对象，  如果x内没有\_id,系统会自动生成一个再插入。相当于上面update语句的upsert=true,multi=false的情况。 |
| public <T> boolean delete(Class<T> clazz,String \_id); | 删除一条数据, 目标集合为T |
| public <T> boolean delete(Collection collection,Class<T> clazz,String \_id); | 删除一条数据-指定集合 |
| public <T> T query(Class<T> clazz,String \_id); | 根据id和集合类型获取一条数据 |
| public <T> List<T> query(Class<T> clazz,DBObject query); | 根据条件获得数据数据列表 |
| public <T> List<T> query(Class<T> clazz,DBObject query,DBObject order,Integer page,Integer limit); | 带分页的自定义查询功能 |
| public <T> Long getCount(Class<T> clazz,DBObject query); | 带分页的自定义查询功能（总数） |
| public <T> T query(Collection collection,Class<T> clazz,String \_id); | 获得一条数据 -指定集合 |
| public <T> List<T> query(Collection collection,Class<T> clazz,DBObject query); | 根据条件获得指定集合的数据列表 |
| public <T> List<T> query(Collection collection,Class<T> clazz,DBObject query,DBObject order,Integer page,Integer limit); | 带分页的自定义查询功能 -指定集合 |
| public <T> Long getCount(Collection collection,DBObject query); | 带分页的自定义查询功能（总数）-指定集合 |

## 运行方式

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