# 缓存操作模块（cache）

## 缓存操作模块概述

缓存操作模块为工作圈提供了一个对Redis缓存操作封装的工具。这样做的好处有以下几点

1. 开发者关注更少的代码，不需要关注缓存的操作具体的实现。
2. 对缓存的资源进行统一管理，避免资源泄漏。
3. 方便维护。当缓存升级之后，只需要将这模块的实现升级，然后依赖此模块的服务更新版本即可。

工作圈中的需要对数据存取加速的服务都依赖于这个模块。

## 缓存操作模块类图



图3-1 mongo模块类图

## DbPool类概述

MongoDB连接池的封装。通过zkHelper从注册中心监听某服务的MongoDB集测配置，初始化MongoDB集群的连接池。通过工作圈服务应用的数据分片key来计算数据映射的节点，并从池中取出此节点的连接供应用来使用。

### 类功能定义

|  |  |  |
| --- | --- | --- |
| **序号** | **功能点** | **功能点详细内容** |
| 1 | getInstance | 获取DbPool的实例 |
| 2 | initialPoolByURI | 初始化连接池，设置参数。 读zookeeper |
| 3 | initialPool | 初始化连接池，设置参数。 读zookeeper |
| 4 | getMongoDataBase | 获取DB实例 |
| 5 | getCollection | 获取MongoDB的集合 |
| 6 | releaseRs | 释放资源 |

## TBaseDao类概述

MongoDB某集合数据的操作类，封装了数据的增，删，改，查操作。

### 类功能定义

|  |  |  |
| --- | --- | --- |
| **序号** | **功能点** | **功能点详细内容** |
| 1 | getInstance | 获取一个TBaseDao的实例 |
| 2 | findByParams | 简单的条件查找 |
| 3 | findByParamsDBObject | 复杂的条件查找 |
| 4 | findByParamsDBObject2 | 复杂的条件查找（返回MongoDB原生对象） |
| 5 | findByParamsDBObjectWithOrderLimit | 复杂条件查找,增加limit和sort |
| 6 | findByParamsDBObjectWithOrderLimit | 复杂条件查找,支持limit skip和sort |
| 7 | findOne | 条件查找唯一记录 |
| 8 | findOne2 | 条件查找唯一记录 |
| 9 | findOneById | 根据主键查找 |
| 10 | addorupdate | 添加或者修改记录 |
| 11 | dropCollection | Drop一个集合 |
| 12 | addBatch | 批量插入（这个慎用，因为一次插入是一个库和表，所以不能走分表逻辑） |
| 13 | update | 修改对象 |
| 14 | updateOrAdd | 修改对象，当对象不存在的时候创建 |
| 15 | remove | 删除对象 |
| 16 | getCount | 获取记录数 |
| 17 | createIndex | 创建索引 |
| 18 | db2Bean | MongoDB对象转换成普通java bean对象 |
| 19 | bean2BasicDb | 对象数据转为MongoDB对象 |