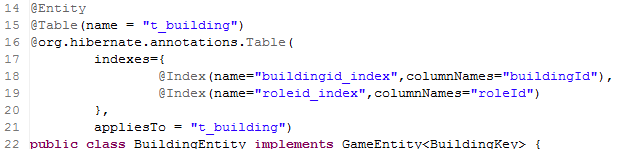
gameserver玩家数据开发流程

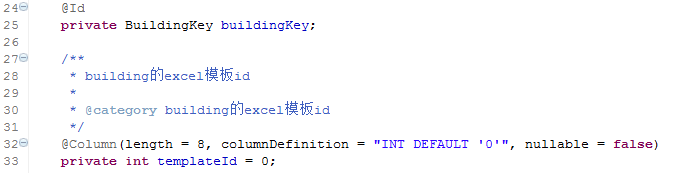
阅读前请先看关于《[数据存储](../1-系统架构/数据存储)》的三个文档

以添加、修改用户建筑表t\_building为例：

1. 在game\_db工程src/com/dl/xcoc/db/entity/下创建文件BuildingEntity.java

部分代码如下





实体实现GameEntity接口

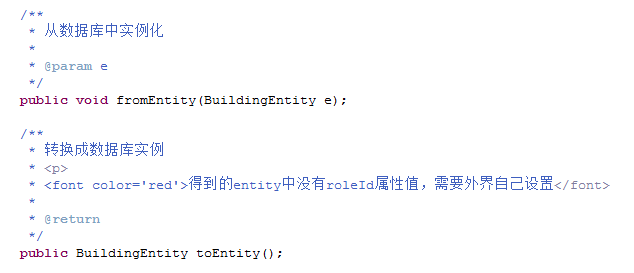
1. 根据配置文件databases.xml里添加一个配置，程序启动时会自动创建表
2. 在game\_db工程src/com/dl/xcoc/db/dao/impl/下编写dao文件BuildingDao.java

部分代码如下

继承AbstractGameDao，AbstractGameDao里包含基本的增删查改等操作，若需要其他个性化操作，则需要进行手工添加。如32行，表示一个按照roleId获得建筑的方法。

1. 在game\_server工程src/com/dl/xcoc/ building下编写Building.java ,继承CacheProperty。

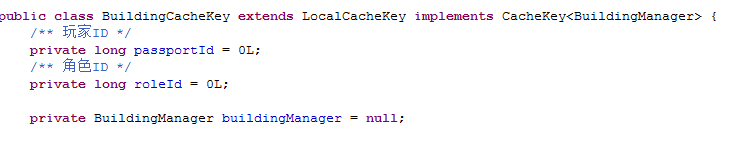


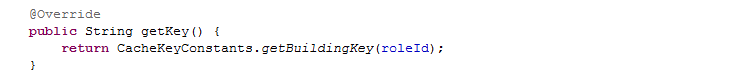


该文件是游戏中业务逻辑使用的Building对象。

编写BuildingManager.java，负责管理玩家的所有Building对象。

1. 在game\_server工程src/com/dl/xcoc/ cache/key下编写BuildingCacheKey.java

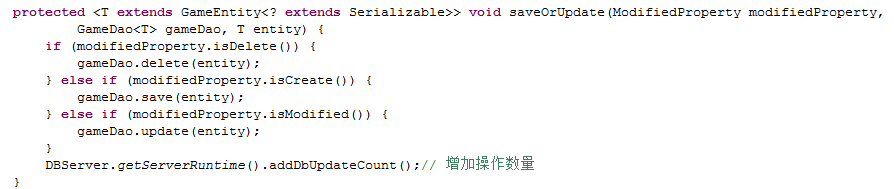




BuildingManager对象发生改变时，使用该类定义的Key，被序列化到Redis缓存中，并自动在Redis缓存中添加相应的变更标示(ModifiedProperty)。

1. 在game\_server工程src/com/dl/xcoc/ building下编写UpdateHumanBuilding.java，实现CacheKey接口。DBS进行延时存库时，根据ModifiedProperty将Redis中缓存的Building数据转换成BuildingEntity入库。

UpdateCacheModified的saveOrUpdate方法如下所示



其中saveOrUpdate包含插入、更新、删除三种操作，具体根据传入ModifiedProperty的操作类型判断。