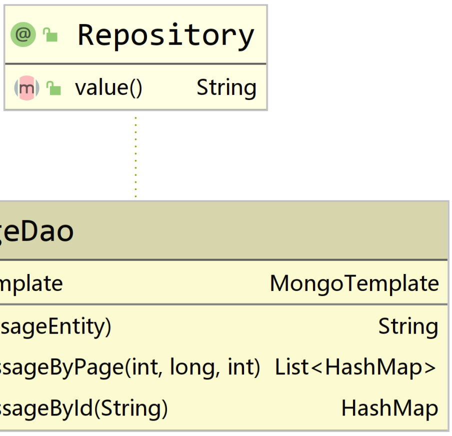
5-4 设计消息模块的持久层

140.143.132.225:8000/project-1/doc-89

一、创建MessageDao类


Repository
value() String
MessageDao
mongoTemplate MongoTemplate
insert(MessageEntity) String
searchMessageByPage(int, long, int) List<HashMap>
searchMessageById(String) HashMap

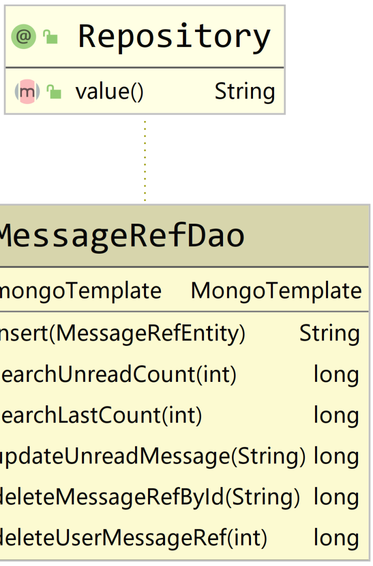
在 `com.example.emos.wx.db.dao` 包中创建 `MessageDao.java` 类

资源众筹换课，加微信 61006008

```java
@Repository
public class MessageDao {
    @Autowired
    private MongoTemplate mongoTemplate;
    public String insert(MessageEntity entity) {
        //把北京时间转换成格林尼治时间
        Date sendTime = entity.getSendTime();
        sendTime = DateUtil.offset(sendTime, DateField.HOUR, 8);
        entity.setSendTime(sendTime);
        entity = mongoTemplate.save(entity);
        return entity.get_id();
    }
    public List<HashMap> searchMessageByPage(int userId, long start, int length) {
        JSONObject json = new JSONObject();
        json.set("$toString", "$_id");
        Aggregation aggregation = Aggregation.newAggregation(
                Aggregation.addFields().addField("id").withValue(json).build(),
                Aggregation.lookup("message_ref", "id", "messageId", "ref"),
                Aggregation.match(Criteria.where("ref.receiverId").is(userId)),
                Aggregation.sort(Sort.by(Sort.Direction.DESC, "sendTime")),
                Aggregation.skip(start),
                Aggregation.limit(length)
        );
        AggregationResults<HashMap> results = mongoTemplate.aggregate(aggregation, "message", HashMap.class);
        List<HashMap> list = results.getMappedResults();
        list.forEach(one -> {
            List<MessageRefEntity> refList = (List<MessageRefEntity>) one.get("ref");
            MessageRefEntity entity = refList.get(0);
            boolean readFlag = entity.getReadFlag();
            String refId = entity.get_id();
            one.remove("ref");
            one.put("readFlag", readFlag);
            one.put("refId", refId);
            one.remove("_id");
            //把格林尼治时间转换成北京时间
            Date sendTime = (Date) one.get("sendTime");
            sendTime = DateUtil.offset(sendTime, DateField.HOUR, -8);
            String today = DateUtil.today();
            //如果是今天的消息，只显示发送时间，不需要显示日期
            if (today.equals(DateUtil.date(sendTime).toDateStr())) {
                one.put("sendTime", DateUtil.format(sendTime, "HH:mm"));
            }
            //如果是以往的消息，只显示日期，不显示发送时间
            else {
                one.put("sendTime", DateUtil.format(sendTime, "yyyy/MM/dd"));
            }
        });
        return list;
    }
    public HashMap searchMessageById(String id) {
        HashMap map = mongoTemplate.findById(id, HashMap.class, "message");
        Date sendTime = (Date) map.get("sendTime");
        //把格林尼治时间转换成北京时间
        sendTime = DateUtil.date(sendTime).offset(DateField.HOUR, -8);
        map.replace("sendTime", DateUtil.format(sendTime, "yyyy-MM-dd HH:mm"));
        return map;
    }
}
```

二、创建MessageRefDao类

在 `com.example.emos.wx.db.dao` 包中创建 `MessageRefDao.java` 类


Repository
value() String
MessageRefDao
mongoTemplate MongoTemplate
insert(MessageRefEntity) String
searchUnreadCount(int) long
searchLastCount(int) long
updateUnreadMessage(String) long
deleteMessageRefById(String) long
deleteUserMessageRef(int) long

资源众筹换课，加微信 61006008

```
@Repository
public class MessageRefDao {
    @Autowired
    private MongoTemplate mongoTemplate;
    public String insert(MessageRefEntity entity) {
        entity = mongoTemplate.save(entity);
        return entity.get_id();
    }
    /**
     * 查询未读消息数量
     *
     * @param userId
     * @return
     */
    public long searchUnreadCount(int userId) {
        Query query = new Query();
query.addCriteria(Criteria.where("readFlag").is(false).and("receiverId").is(userId));
        long count = mongoTemplate.count(query, MessageRefEntity.class);
        return count;
    }
    /**
     * 查询新接收消息数量
     *
     * @param userId
     * @return
     */
    public long searchLastCount(int userId) {
        Query query = new Query();
query.addCriteria(Criteria.where("lastFlag").is(true).and("receiverId").is(userId));
        Update update = new Update();
        update.set("lastFlag", false);
        UpdateResult result = mongoTemplate.updateMulti(query, update, "message_ref");
        long rows = result.getModifiedCount();
        return rows;
    }
    /**
     * 把未读消息变更为已读消息
     *
     * @param id
     * @return
     */
    public long updateUnreadMessage(String id) {
        Query query = new Query();
        query.addCriteria(Criteria.where("_id").is(id));
        Update update = new Update();
        update.set("readFlag", true);
        UpdateResult result = mongoTemplate.updateFirst(query, update, "message_ref");
        long rows = result.getModifiedCount();
        return rows;
    }
    /**
     * 根据ID删除ref消息
     *
```

课，加微信 610060008

```
     * @param id
     * @return
     */
    public long deleteMessageRefById(String id) {
        Query query = new Query();
        query.addCriteria(Criteria.where("_id").is(id));
        DeleteResult result = mongoTemplate.remove(query, "message_ref");
        long rows = result.getDeletedCount();
        return rows;
    }
    /**
     * 删除某个用户全部消息
     *
     * @param userId
     * @return
     */
    public long deleteUserMessageRef(int userId) {
        Query query = new Query();
        query.addCriteria(Criteria.where("receiverId").is(userId));
        DeleteResult result = mongoTemplate.remove(query, "message_ref");
        long rows = result.getDeletedCount();
        return rows;
    }
}
```

资源众筹换课，加微信 610060008