**全心生活.系统消息发送业务**

1. **消息模板**

消息模板通过后台配置，并缓存在Redis服务器上，消息发送程序通过连接并获取Redis中对应业务的消息模板。

1. **消息内容**

根据业务特征，构造消息发送器时传入关键参数值后进行业务数据整理，通过替换消息模板中的特殊占位符形成最终系统消息内容。

1. **业务及对应的消息发送器方法**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **业务** | **业务说明** | **对应消息发送器** | **参数说明** | | | **发送方法** |
| **参数** | **值类型** | **参数说明** |
| **提交投诉/举报信息成功** | 提交投诉/举报信息成功后，发送消息给投诉（或举报）用户 | AddComplaintMessageSender | complaintID | Int | 举报或投诉信息的数据ID | Send()  返回:bool |
| userID | Long | 举报或投诉人用户ID |
| **提交反馈信息成功** | 提交反馈信息成功后发送消息给反馈用户 | AddFeedbackMessageSender | feedbackID | Int | 反馈信息数据ID | Send()  返回:bool |
| userID | Long | 反馈用户ID |
| **用户被设置为圈子版主** | 用户被设置为圈子版主后发送消息给用户 | SetModeratorByForumMessageSender | fromID | Long | 圈子ID | Send()  返回:bool |
| userID | Long | 被设置为版主的用户ID |
| **用户被取消圈子版主** | 用户被取消版主后发送消息给用户 | CancelModeratorByForumMessageSender | forumID | Long | 圈子ID | Send()  返回:bool |
| userID | Long | 被取消版主的用户ID |
| **社区活动提醒** | 兴趣社区中活动开始前对活动报名者发送提醒消息 | CircleEventRemindMessageSender | topicID | Long | 活动帖子ID | Send()  返回:bool |
| eventBeginTime | DateTime | 活动开始时间 |
| **用户首次登录** | 用户首次登录时发送消息给用户 | FirstLoginMessageSender | areaID | Int | 登录区域ID | Send()  返回:bool |
| userID | Long | 登录的用户ID |
| **商家取消商品订单** | 商家取消商品订单后发送消息给买家用户 | MerchantCancelGoodsOrderMessageSender | orderID | Long | 订单ID | Send()  返回:bool |
| **商家取消服务订单** | 商家取消服务订单后发送消息给买家用户 | MerchantCancelServiceOrderMessageSender | orderID | Long | 订单ID | Send()  返回:bool |
| **商家确定商品订单** | 商家确定商品订单后发送消息给买家用户 | MerchantConfirmGoodsOrderMessageSender | orderID | Long | 订单ID | Send()  返回:bool |
| **商家确定服务订单** | 商家确定服务订单后发送消息给买家用户 | MerchantConfirmServiceOrderMessageSender | orderID | Long | 订单ID | Send()  返回:bool |
| **帖子被取消热门** | 取消热门后发送消息给发帖人 | TopicCancelHotMessageSender | topicID | Long | 帖子ID | Send()  返回:bool |
| reason | String | 取消原因 |
| serverPhone | String | 客服电话 |
| **帖子被取消置顶** | 取消置顶后发送消息给发帖人 | TopicCancelTopMessageSender | topicID | Long | 帖子ID | Send()  返回:bool |
| reason | String | 取消原因 |
| serverPhone | String | 客服电话 |
| **帖子被删除** | 删除后发送消息给发帖人 | TopicDeletedMesseageSender | topicID | Long | 帖子ID | Send()  返回:bool |
| reason | String | 取消原因 |
| serverPhone | String | 客服电话 |
| **帖子设置为热门** | 设置为热门后发送消息给发帖人 | TopicSetHotMessageSender | topicID | Long | 帖子ID | Send()  返回:bool |
| serverPhone | String | 客服电话 |
| **帖子置顶** | 置顶后发送消息给发帖人 | TopicSetTopMessageSender | topicID | Long | 帖子ID | Send()  返回:bool |
| serverPhone | String | 客服电话 |
| **用户等级提升** | 等级提升后发送消息给用户 | UserLevelUpMessageSender | userID | Long | 用户ID | Send()  返回:bool |
| remark | String | 备注说明 |
| **用户实名认证审核失败** | 审核失败后发送消息给用户 | UserRealNameCertificationFailureMessageSender | userID | Long | 用户ID | Send()  返回:bool |
| reason | String | 失败原因 |
| serverPhone | String | 客服电话 |
| **用户实名认证审核成功** | 审核成功后发送消息给用户 | UserRealNameCertificationSuccessMessageSender | userID | Long | 用户ID | Send()  返回:bool |
| serverPhone | String | 客服电话 |
| **商家福利审核失败** | 审核失败后发送消息通知商家 | WelfareAuditFailureMessageSender | welfareID | Logn | 福利ID | Send()  返回:bool |
| serverPhone | String | 客服电话 |
| **商家福利审核成功** | 审核成功后发送消息通知商家 | WelfareAuditSuccessMessageSender | welfareID | Logn | 福利ID | Send()  返回:bool |
| serverPhone | String | 客服电话 |
| **用户福利即将过期提醒** | 失效前发送消息通知用户 | WelfareBeforLoseEfficacyMessageSender | consumerCode | Long | 福利消费码 | Send()  返回:bool |
| **福利开奖前提醒** | 开奖前提醒该福利参与用户 | WelfareLotteryBeforeMessageSender | welfareID | Long | 福利ID | Send()  返回:bool |
| **商家福利开奖后消息通知** | 开奖后向所有参与该福利的用户发送消息 | WelfareLotteryResultMessageSender | welfareID | Logn | 福利ID | Send()  返回:bool |
| serverPhone | String | 客服电话 |
| **商家福利中奖用户通知** | 开奖后向中奖用户发送消息 | WelfareLotteryWinnerMessageSender | welfareID | Long | 福利ID | Send()  返回:bool |
| **用户福利使用后通知** | 福利使用后发送通知给用户 | WelfareUsedMessageSender | consumerCode | Long | 福利消费码 | Send()  返回:bool |

1. **消息插件注入方式**

注入方法：MessageSenderExtensions.Factory(string connectionString, SqlProviderType sqlType, DataCacheServerOptions options);

参数说明：connectionString，业务操作的数据库连接字符串

sqlType，业务操作的数据库类型,枚举：SqlProviderType

options，数据缓存服务配置，配置类型：DataCacheServerOptions

注入场景代码，如下：

//数据库连接字符串

string sqlConn = "data source=.;initial catalog=dbname;user id=userid;password=pwd;";

//数据库类型

SqlProviderType sqlType= Td.Kylin.EnumLibrary.SqlProviderType.SqlServer;

//Redis连接

string redisConn = "139.129.194.132:6399,abortConnect=false,password=kylinjonfee++";

//消息发送器构建注入

MessageSenderExtensions.Factory(

connectionString: sqlConn, //业务操作的数据库连接字符串

sqlType: sqlType, //业务操作的数据库类型

options: new DataCacheServerOptions //数据缓存服务配置项

{

KeepAlive = true, //是否长连接

CacheItems = null, //需要缓存的数据项，枚举：CacheItems

RedisConnectionString = redisConn, //缓存数据所在的Redis服务器连接

InitIfNull = false, //缓存数据为null时是否初始化

SqlType = sqlType, //缓存数据的源数据库类型

SqlConnection = sqlConn, //缓存数据的源数据库连接字符串

Level2CacheSeconds = 600 //数据二级缓存的时间（单位：秒）

});

1. **举例**

以用户首次登录为例：

假设首次登录用户ID为1624894317931569，当前所在区域为深圳市（440300），则登录成功后发送系统消息代码如下：

int areaID = 440300;

long userID = 1624894317931569;

FirstLoginMessageSender sender = new FirstLoginMessageSender(areaID, userID);

bool success = sender.Send();

if(success){

//系统消息发送成功

……

}

else{

//系统消息发送失败

……

}

1. **其它**