# 开发文档

# 消息生成

服务器的操作：（按顺序执行）

1. 添加 game\_tools/config/msg/model/\*\_message.xml
2. 修改（如果有新增的\*\_message.xml，则需要修改文件，没有就直接跳过第二步）

game\_tools/config/msg/message\_generator.js

game\_tools/config/msg/messageTypeConfig.xls

1. 运行 game\_tools/src/com/imop/lj/tools/msg/MessageGenerator.java
2. 修改与消息对应的handler方法

客户端的操作：

1. 添加

将（服务器）game\_tools/target/cs\_target目录下除db和handler文件夹外的所有文件拷贝到

（客户端）trunk\u3u2\Assets\Scripts\app\net名字一致的文件夹中，MessageReciver.cs和MessageType.cs放在net根目录中。

1. 修改

如果（服务器）handler文件夹中有自己新生成的\*CGHandler.cs和\*GCHandler.cs文件，就直接复制到（客户端）handler文件夹中；如果是以前有的，就将自己在（服务器）\*CGHandler.cs和\*GCHandler.cs文件中新加的东西添加到对应的（客户端）handler文件中。

# 增加数据表（仿写酒馆的数据存取）

1. 添加

game\_db/src/com/imop/lj/db/model/PubTaskEntity.java（新建数据库实体类）

game\_server/src/com/imop/lj/gameserver/pubtask/PubTask.java（新建数据库对象类）

game\_server/src/com/imop/lj/gameserver/pubtask/persistance/PubTaskDbManager.java

game\_server/src/com/imop/lj/gameserver/pubtask/persistance/PubTaskUpdater.java

game\_db/src/com/imop/lj/db/dao/PubTaskDao.java（新建数据库操作类）

1. 修改

game\_server/src/com/imop/lj/gameserver/player/persistance/PlayerDataUpdater.java

game\_server/src/com/imop/lj/gameserver/common/db/GameDaoService.java（添加DAO管理类初始化）

game\_server/config/game\_server\_hibernate\_query.xml（添加数据库查询条件）

game\_server\_hibernate.cfg.xml（添加实体类映射）

**增加属性管理（仿写装备的特殊属性）**

1. 添加

game\_server/src/com/imop/lj/gameserver/item/feature/AbstractEquipFeature.java（新建属性 的抽象定义）

game\_server/src/com/imop/lj/gameserver/item/feature/EquipFeature.java（继承抽象定义）

game\_server/src/com/imop/lj/gameserver/item/template/EquipItemTemplate.java（通过模版初始化空的feature）

1. 测试新添加feature用到的类

game\_server/src/com/imop/lj/gameserver/command/impl/GiveItemCmd.java（添加gm获取物品命令）

# log生成

1. 添加 game\_tools/config/log/model/\*.txt
2. 修改 game\_tools/config/log/model\_msg\_gen.config
3. 运行 game\_tools/src/com.imop.lj.tools.log.LogMsgGenerator
4. 修复game\_server 的错误

# 增加定时任务

## player相关定时任务

## scene相关定时任务

# 模板生成VO文件

服务器的操作：

1. 添加 resources/scripts/\*.xls

game\_tools/config/excel/model/\*Template.txt

1. 修改 game\_tools/config/excel/model\_template\_gen.config
2. 运行 game\_tools/src/com/imop/lj/tools/excel/ExcelTemplateGenerator.java
3. 添加 找到新生成的\*TemplateVO文件，在同一个包下添加\*Template.java文件
4. 关联 game\_server/config/templates.xml

客户端的操作：

1. 将（服务器）game\_tools/target/cs\_target/db目录下的所有文件拷贝到

（客户端）trunk\u3u2\Assets\Scripts\app\db\AutoCfg中，在trunk\u3u2\Assets\Scripts\app\db\cfg

文件夹中新添加与AutoCfg中对应的\*.cs文件（仿写该文件夹中的前两个）

# 添加多语言

1. 添加 core/src/com/imop/lj/common/constants/LangConstants.java（添加语言相关的常量定义）
2. 运行 game\_tools/src/com/imop/lj/tools/i18n/SysLangGenerator.java

# RobotMain测试

1. 添加 robot/src/com/imop/lj/robot/strategy/impl/\*TestStrategy.java（新建机器人执行策略）、
2. 修改 robot/src/com/imop/lj/robot/startup/RobotClientMsgRecognizer.java（添加测试监听）
3. 运行 robot/src/com/imop/lj/robot/RobotMain.java（robot中添加一个机器人执行策略并运行）