# 

# 程序文档\_扩展（实践10）版本

Git链接：<https://github.com/BrunoQin/Open-Reuse.git>

组员：1352958 金敏 1352911 曹琦 1354366 潘舜达

1352961 秦博 1352839 饶伊文

## 目录

[程序文档\_扩展（实践10）版本 1](#_Toc1090)

[1. 引言 2](#_Toc25140)

[1.1 编写目的 2](#_Toc17598)

[1.2 背景 2](#_Toc11568)

[2. 项目设计说明 2](#_Toc29720)

[2.1 项目描述 2](#_Toc19803)

[2.2 新增扩展功能 2](#_Toc27534)

[2.3 新增数据自卸车队列说明 2](#_Toc14131)

[2.4 新增压缩文件加密和解密](#_Toc13560) [4](#_Toc13560)

[2.5 修改可配置路径文件归档压缩说明 4](#_Toc24376)

[2.6 日志管理 4](#_Toc25789)

2.7 新增/改进接口..........................................................................................................4

[3. 开发成果及说明 5](#_Toc14996)

**1. 引言**

**1.1 编写目的**

本报告旨在说明本项目的第九次扩展功能版本的功能设计及代码实现。预期读者为架构师，开发者，测试人员等软件行业专业人员。

**1.2 背景**

本项目的开发形式为 Team Work，成员包括 1352958 金敏，1352911 曹琦，1354366 潘舜达，1352961 秦博，1352839 饶伊文。

本项目开发环境为Java，IDE为Intellij IDEA，使用了gradle构建工具和Hibernate映射框架。数据库使用MySQL。

版本控制工具使用 Github (https://github.com/BrunoQin/Open-Reuse.git )。测试工具使用 Junit 和 Mockito测试框架。

**2. 项目设计说明**

**2.1 项目描述**

该项目主要完成客户端和服务器的基本通信的设计和实现，包括客户端的登录，注册，收发消息以及服务器的接收和转发消息。

本次版本更新为新增简单的扩展功能，是基于前一次对用户分组之后的消息保存做了一些改动与更新，具体更新内容如下。

**2.2 新增扩展功能及实现说明**

**1. 分组功能**

配置同组成员。

只向同组成员广播消息。

**2. 维护同组成员列表**

Client维护已经登录的同组成员列表。

**2. 维护历史消息记录**

Client登陆后，有序接收所有遗漏的信息。

**2.3 分组功能实现说明**

由于在之前的设计中各部分解耦度高，只是简单对消息类型进行更改并在服务端添加简单的分组文本消息处理就可以实现。客户端添加一个监听后，只要发送分组信息类型的信息就能实现分组处理。具体如下：

在公共文档enum MessageType 信息类型枚举中添加分组的消息类型：

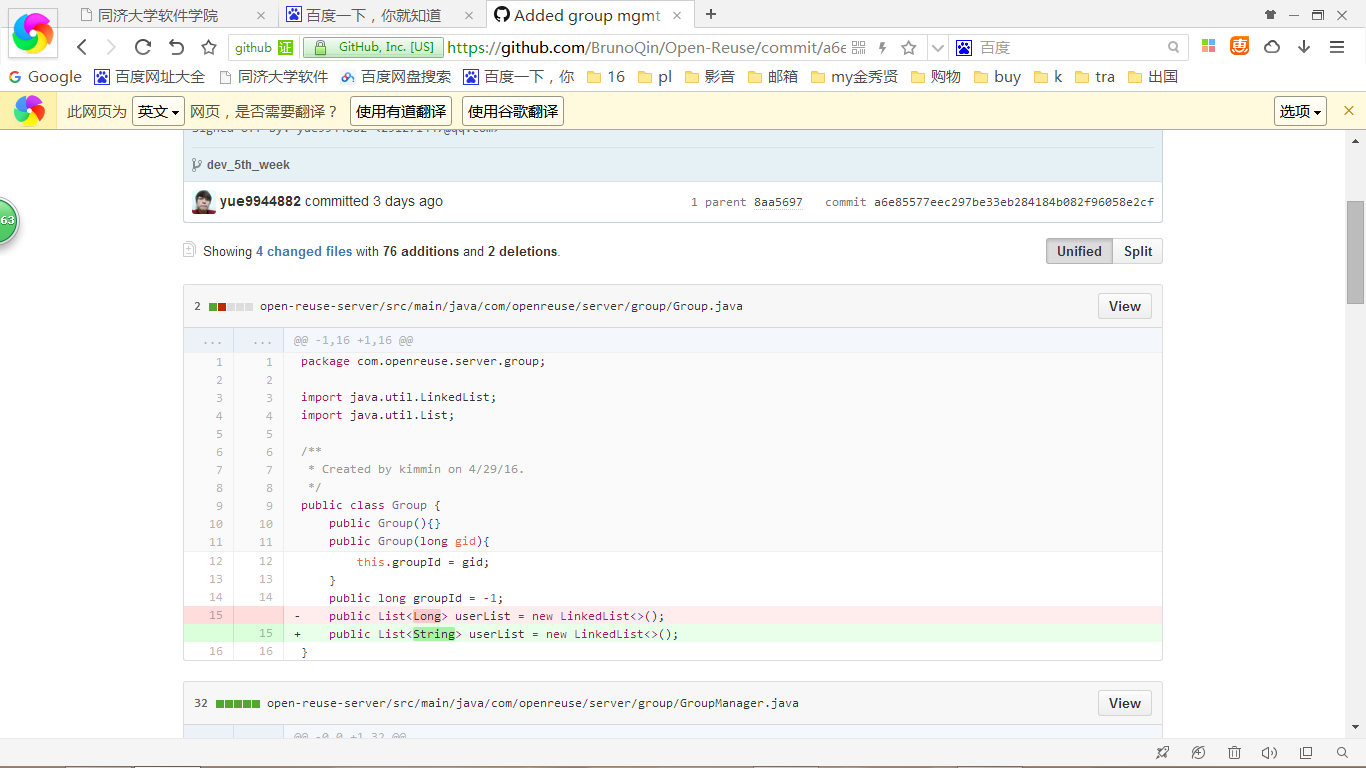
TEXT\_MESSAGE\_GROUP("text\_group"),

REGISTER\_GROUP\_MESSAGE("register\_group"),

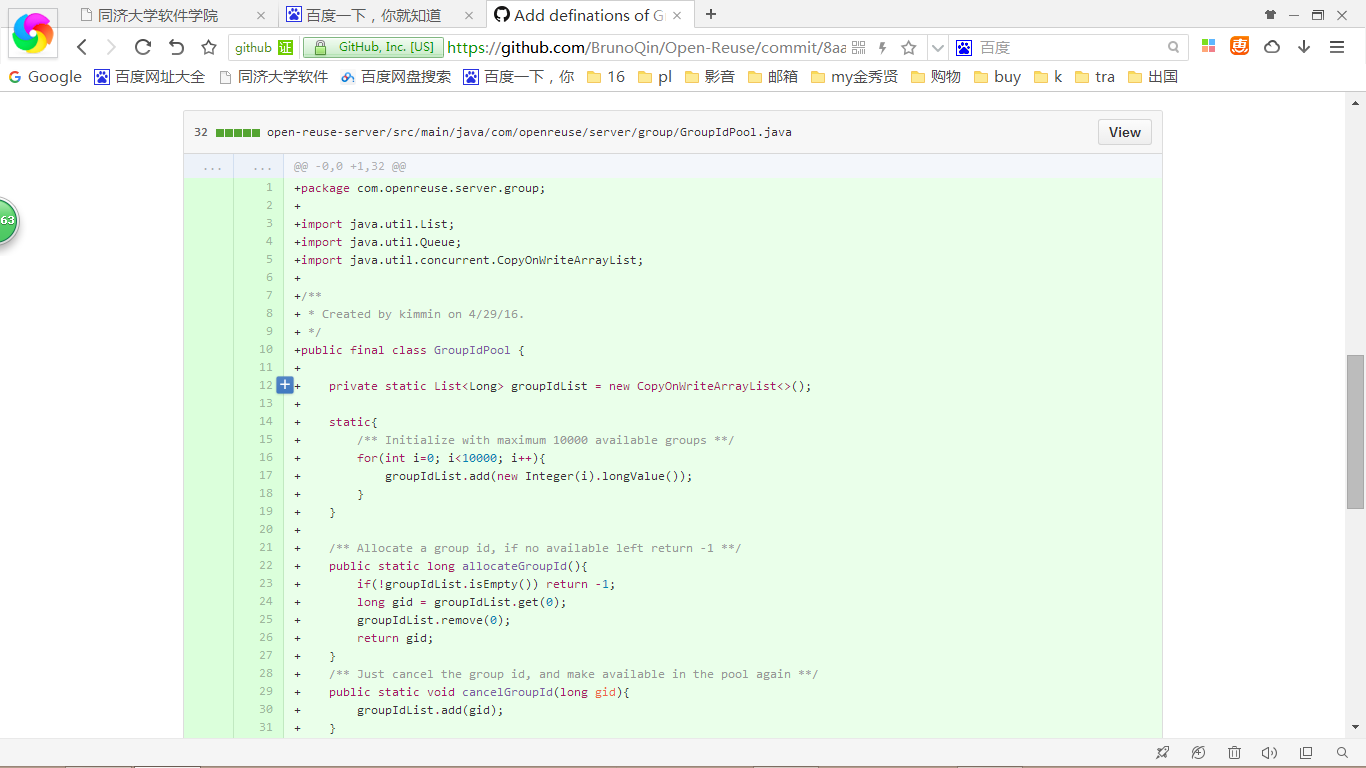
RESP\_REGISTER\_GROUP\_OK("response\_register\_group");

服务器端:

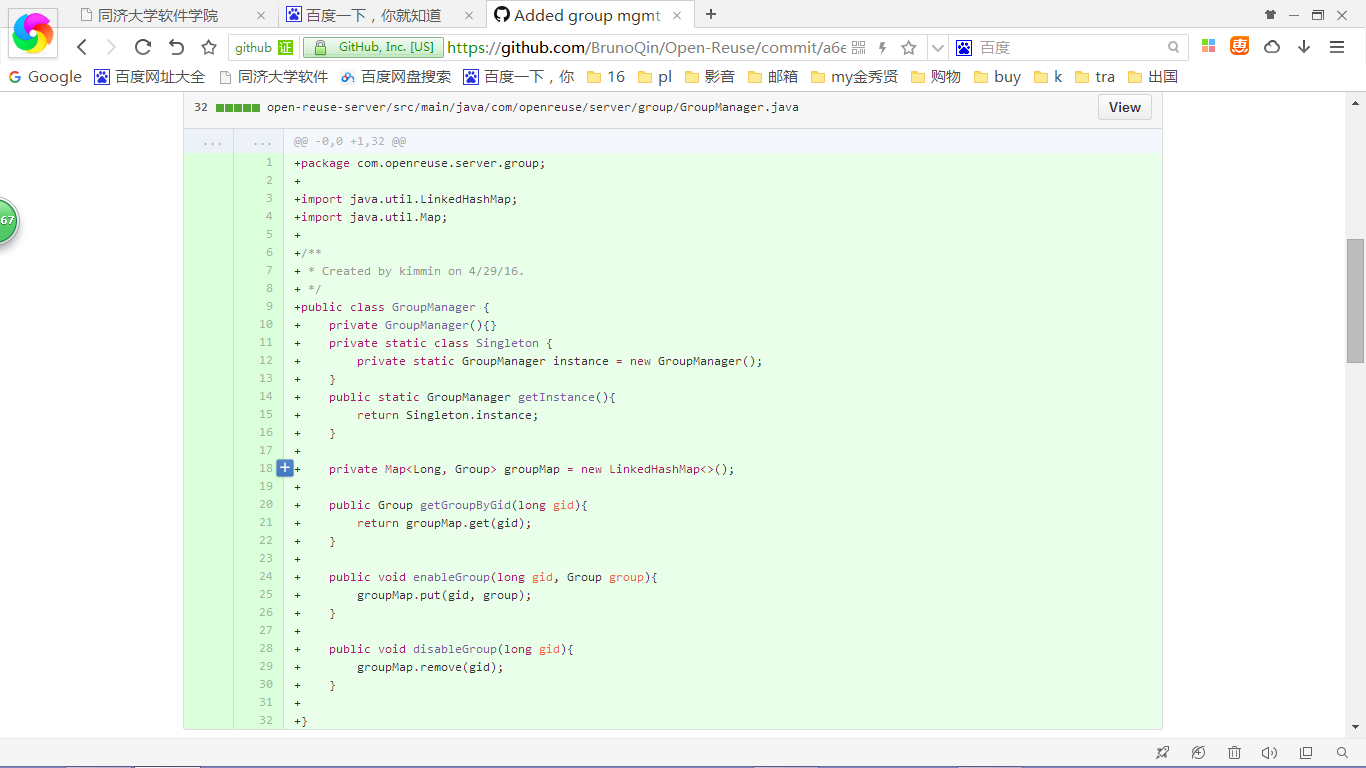
在服务器端添加分组类Group,为每组储存组号和组员信息。



在服务器端添加分配组号的类：GroupIdPool（默认最多10000组）

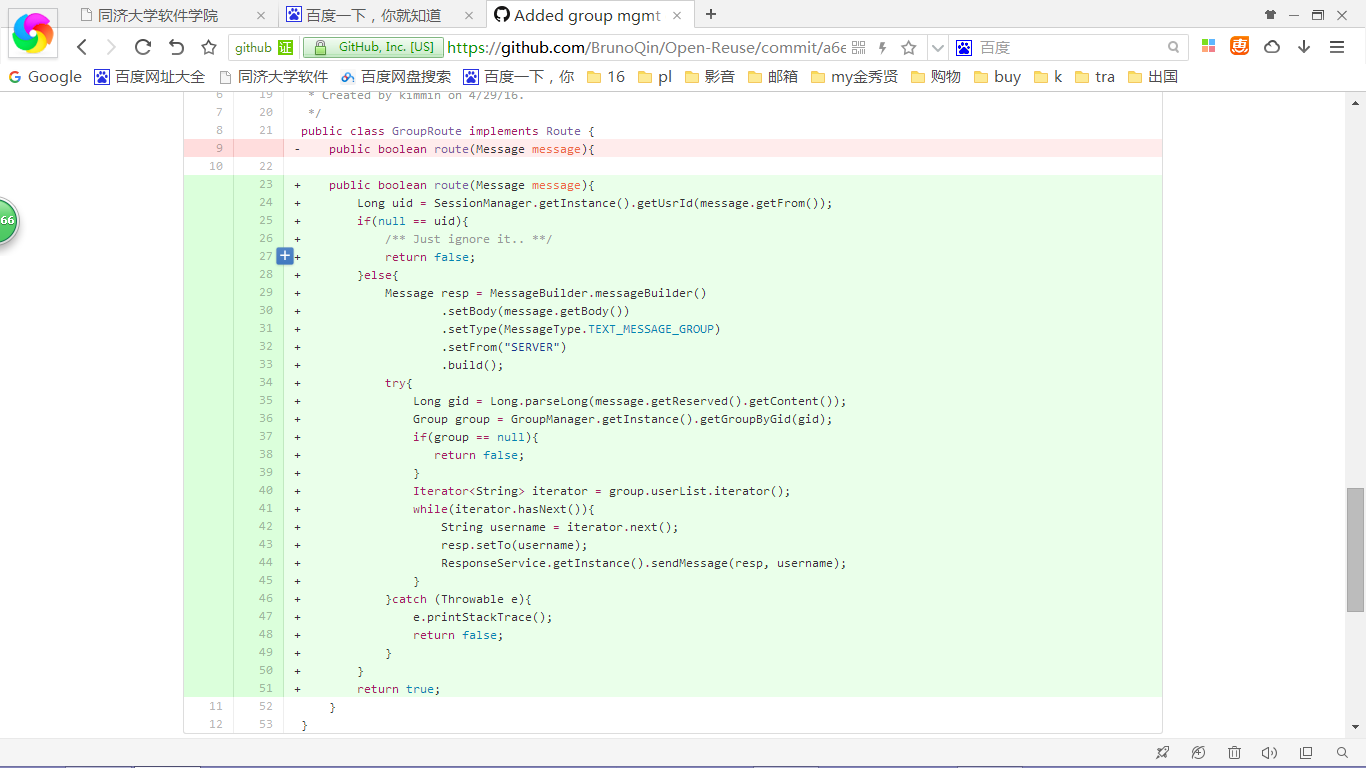


添加分组管理类GroupManager，使用GroupMap记录分组信息，实现分组，分组删除，取得分组id的功能（单例模式）

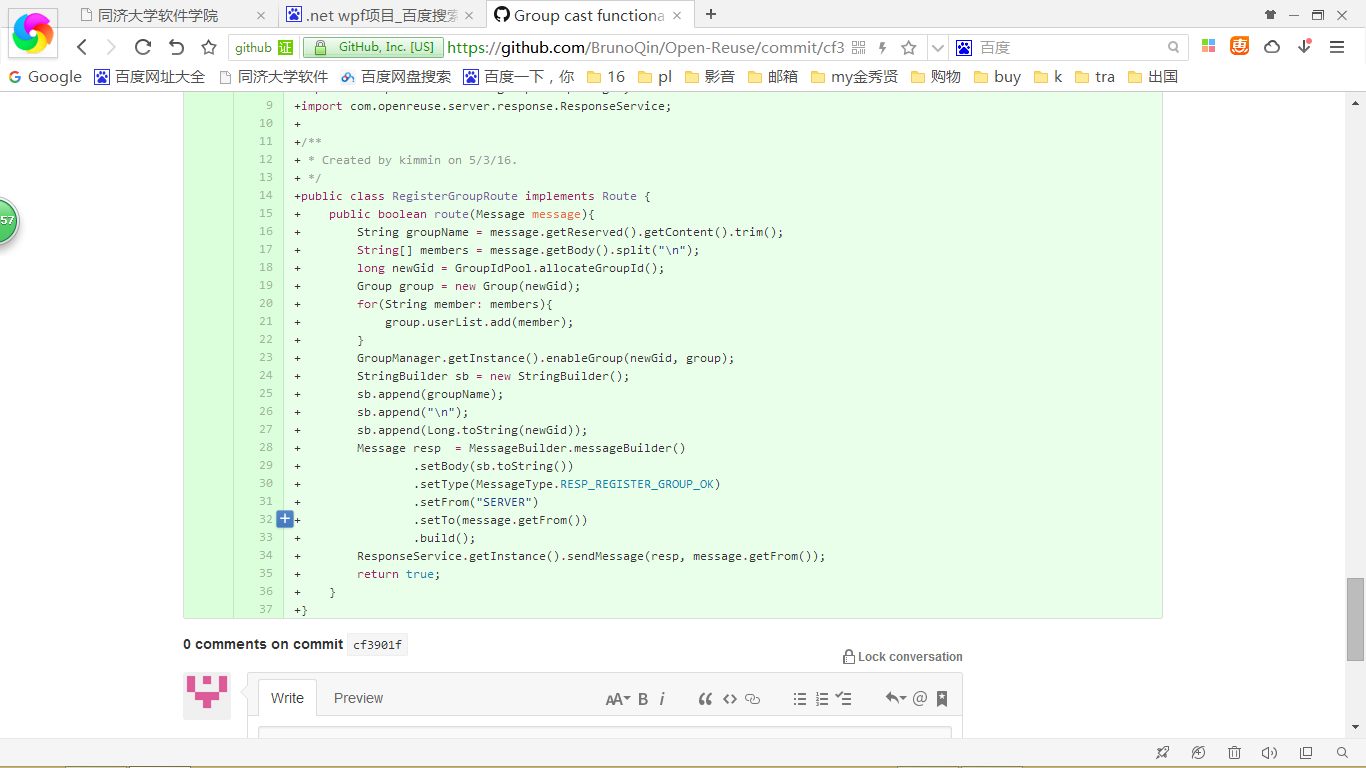


在服务端消息路径中添加分组文本消息处理路径和注册分组处理路径，并添加到RouteMap中

open-reuse-server/src/main/java/com/openreuse/server/request/route/GroupRoute.java



open-reuse-server/src/main/java/com/openreuse/server/request/route/RegisterGroupRoute.java



open-reuse-server/src/main/java/com/openreuse/server/request/dispatcher/RouteDispatcher.java

routeMap.put(MessageType.TEXT\_MESSAGE\_GROUP, new GroupRoute());

routeMap.put(MessageType.REGISTER\_GROUP\_MESSAGE, new RegisterGroupRoute());

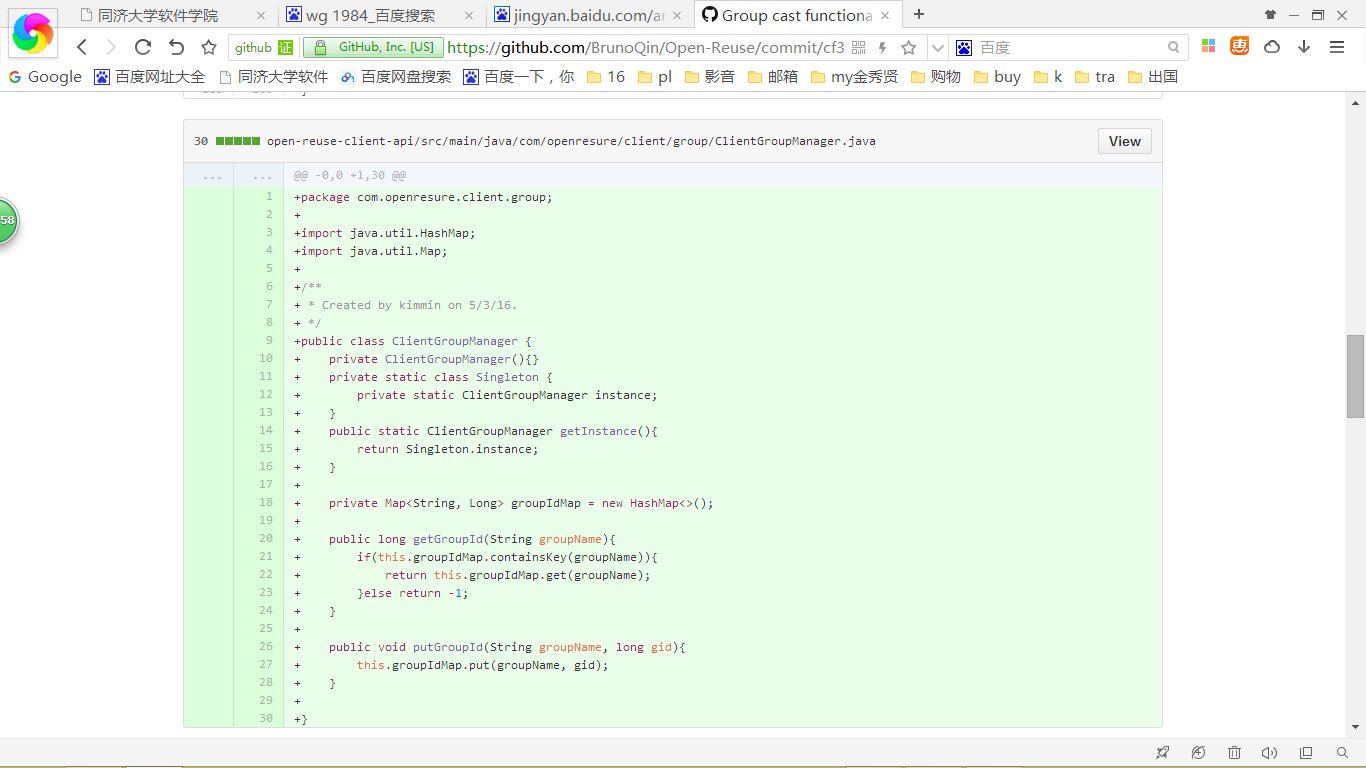
客户端

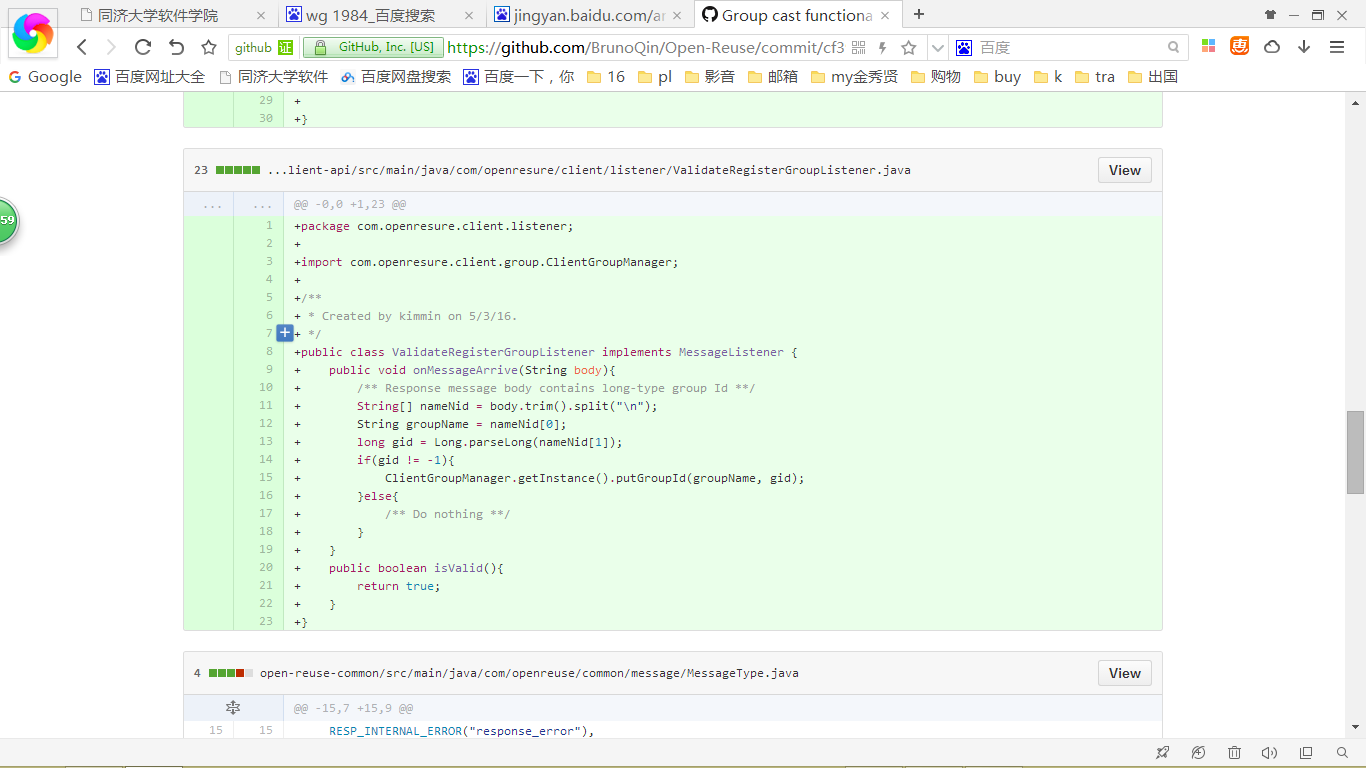
在服务器端代理中添加分组响应函数：注册组和发送组内消息

open-reuse-client-api/src/main/java/com/openresure/client/ClientAgent.java

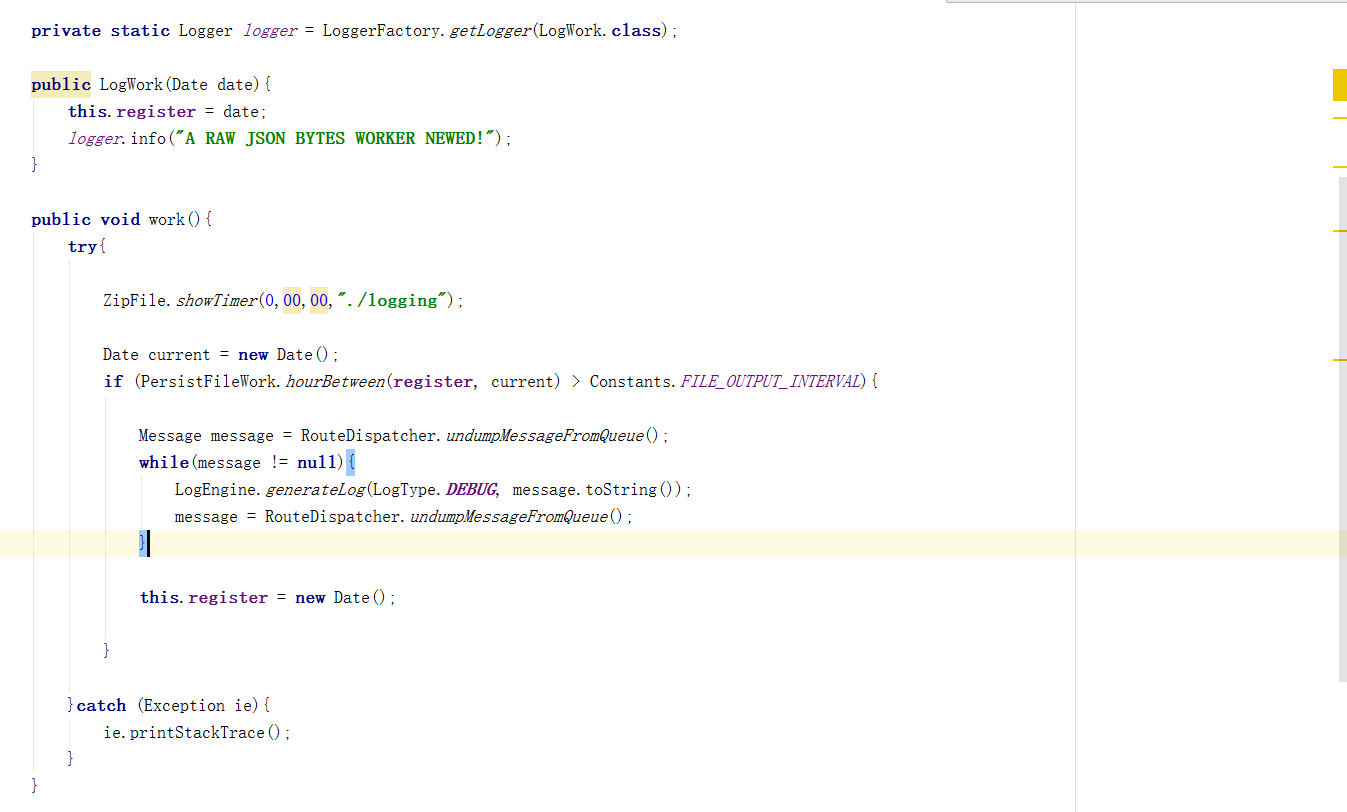


客户端添加分组管理类，用储存本客户端的分组信息，被监听器调用

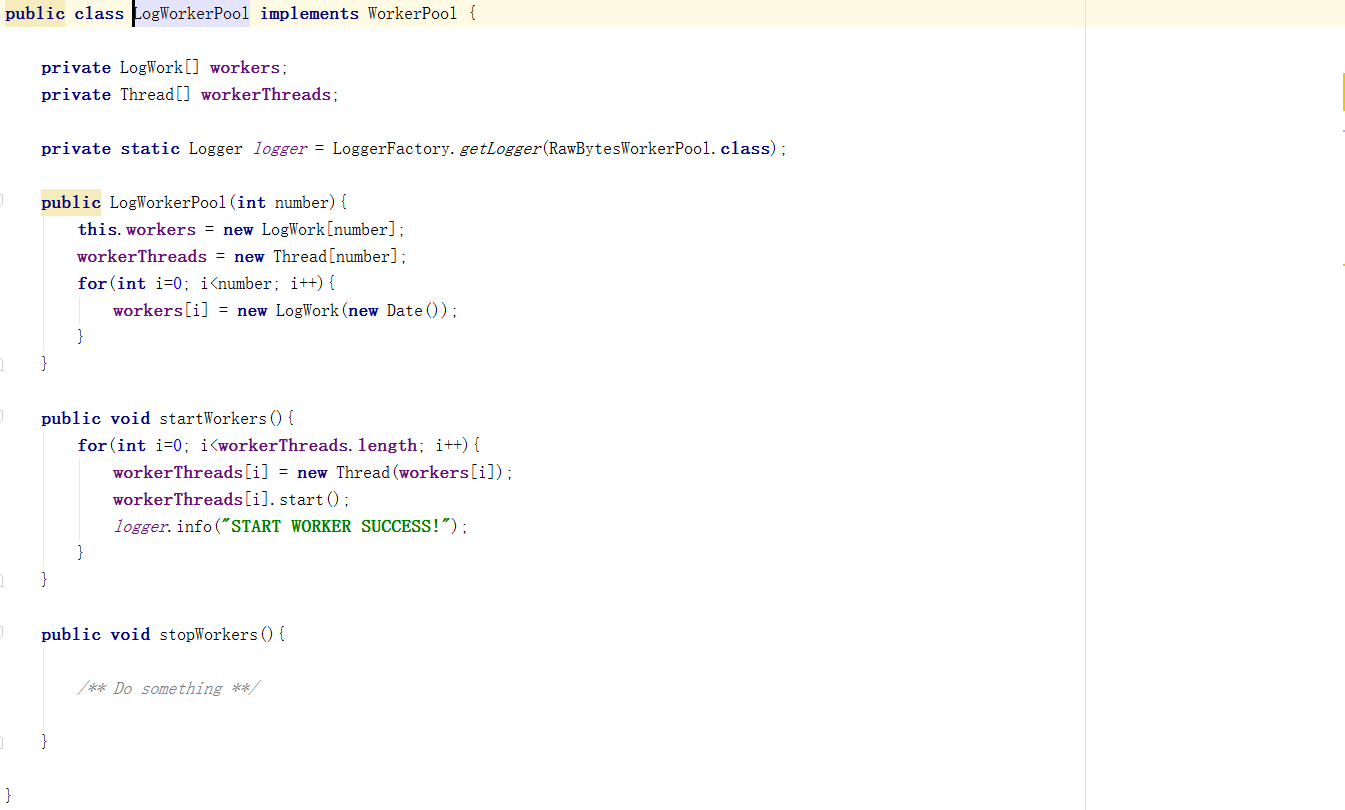




**2.4 日志管理**



LogWork负责一个日志管理的线程。负责对日志进行记录和归档，同时给出相应的操作信息。

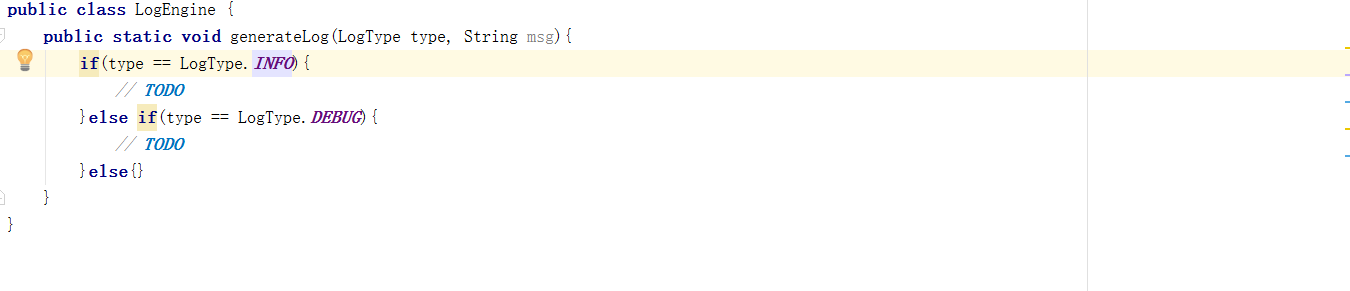


WorkPool是一个管理work线程的线程池。在线程池中可以对线程进行操作，从而使得程序的运行和资源的分配更加合理。

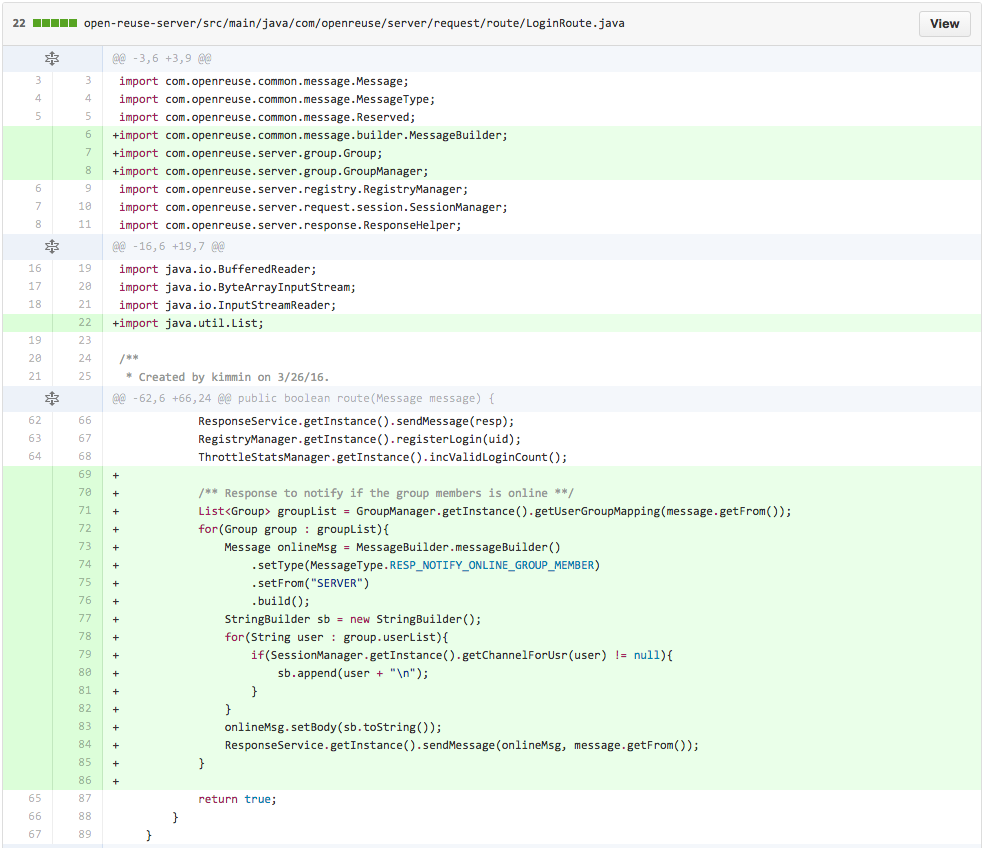


LogService是负责管理线程池的。在这之中可以启动线程池和对线程池进行管理，分配资源。

**2.5日志公共接口**



该类有两个参数，一个是日志类型，一个是日志内容，参数传入后，会对信息进行记录和提示，从而完成该功能。



用户登录以后，系统会根据用户所在的group获取当前group的历史消息，并且将这些消息分发到client中。

**2.6 新增/改进接口**

**com.openreuse.server.group**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| getInstance():ThrottleManager | 新建并获得类实例 |  |  | 对GroupMap操作，内容为Group类 |
| getGroupByGid(gid:long):Group | 通过组号获得组 | gid:long | Group |  |
| enableGroup(gid:long, group:Group):void | 添加组 | gid:long, group:Group |  |  |
| disableGroup(gid:long):void | 移除组 | gid:long |  |  |
| startWorkers() | 启动线程 |  |  |  |
| LogEngine.generateLog | 记录日志 | LogType type, String msg |  |  |

**com.openreuse.server.group**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| allocateGroupId():long | 分配组号 |  | Long组号 | 默认最多10000组 |
| cancelGroupId(gid:long):void | 释放组号使可以再使用 | gid:long |  |  |
| writeFilePerLogin():Message | 将Message写到创建好的文件中 | message:Message | Bool:true or false | 先创建FileWriter，再调用writeFilePerLogin() |
| work() | 将文件按时间统计分类 |  |  |  |
| run() | 启动work() |  |  |  |

**com.openreuse.server.request.route**

|  |  |
| --- | --- |
| false | route(message:Message):boolean |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| route(message:Message):boolean | 分组消息处理路径 | message:Message | boolean | 添加组员信息到组别 |

**3. 开发成果及说明**

经过一周的设计和开发，新增扩展都得到了实现，并在使用复用构件情况下表现良好。

由于整个项目组织结构清晰，解耦程度大，扩展时带来很多方便。

本次新增扩展功能分工主要为：

分组消息及通知添加： 开发：金敏 文档：秦博

测试： 开发：曹琦 文档：饶伊文