复用文档

项目组： **Team3**

成员： **1352837** 倪雨婷

1352866 罗晓丹

1352873 王刚

1352895 叶坤宇

1352918 刘旭东

一． 可复用构件

1. ActiveMq 消息中间件

对 ActiveMq 进行了封装，并使用 JMX 监视 ActiveMq 以获取相关消息收发的数据信息。使得利用 ActiveMq 进行消息发送便利简单。

2. Configuration 配置管理

配置文件记录了中间件端口号默认 port：61616、时间间隔默认 timeGap：1000ms、客户端每秒登录的最大次数默认 max：5 以及服务端运行期间生成的性能文件的存储路径默认 path：C：\\。可根据需要对配置文件进行修改，项目能对配置文件实时动态加载，且提供查询配置参数的方法。

3. LicenseManager 许可证管理

每收到一次请求，请求数累计加 1，判断消息数是否达到配置的最大值，判断是否可继续提供服务。

4. PerformanceManager 性能管理

接收程序的性能指标：有效登录次数、失败登录次数，且每秒生成性能报告并计算总登录次数，将性能报告按报告时间输出到配置文件路径。

二．使用方法

1. ActiveMq 消息中间件

需引入 Topic.jar

```
long ClientCount=MySubscriber.getConsumerCount();//利用 JMX 获取当前成功登录的客户端总数量；
```

JMX 的配置请参考 Readme。

2. Configuration 配置管理

需引入 JSON 所需的包/jar/json 内，以及 Configuration.jar

```
String ReadFile(String path); //读取配置文件信息，path 为配置文件路径
String getConfigurationFileInfo(); //提供查询接口，返回配置文件信息
String getPath(); //获取文件存储路径
String getMaxRequestTimes(); //获取允许客户端每秒登录的最大次数
String getTimeGap(); //获取服务时间间隔
String getPort(); //获取中间件端口号
```

3. LicenseManager 许可证管理 Capacity

提供 Throughput 和 Capacity 两种检测方式，同样需要引入 LicenseManager.jar

```
//初始化
LicenseManager licenseManager = new LicenseManager();
/** * max: 预设的上限 * init: 计数初始值 */
licenseManager.CapacityInit(int max, int init);
/** * max: 预设的上限 * time: 时间间隔 * init: 计数初始值 */
licenseManager.ThroughputInit(int max,long time, int init);
//使用:// 返回值为 bool，如果是 true 表示未超过上限，如果是 false 表示已超过上限
licenseManager.CapacityCheck(); //消息数上线检查
licenseManager.ThroughputCheck(); //服务时间间隔检查
```

4. PerformanceManager 性能管理

需要引入 PerformanceManager.jar

```
/** * path: 输出文件的路径 * delay: 多长时间输出一次 */
```

```
    PerformanceManager performanceanager = new PerformanceManager(String path,long delay);
```

启动 PM 管理：

```
    performanceManager.start();//现在支持两种性能属性，成功登录次数 performanceManager.successTime 和失败登录次数 performanceManager.failTime;
```