性能管理重构文档

一、所选构件

https://github.com/SummerWish/TahitiPerformanceMonitor

二、选择理由

1.可使用maven添加项目依赖。

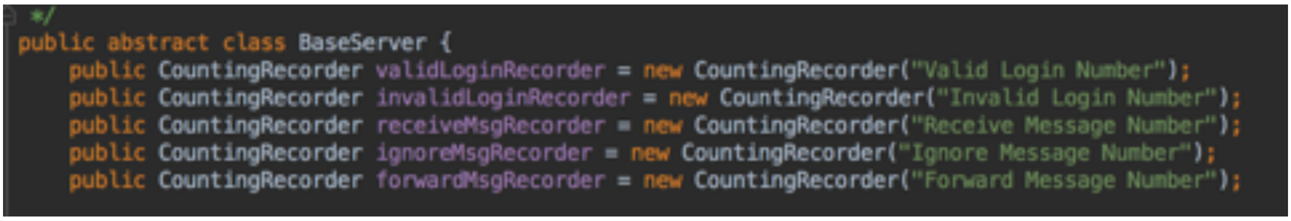
2.接口清晰简洁，易于使用。

3.功能齐全，完全可以满足需求，并具有良好的可扩展性。

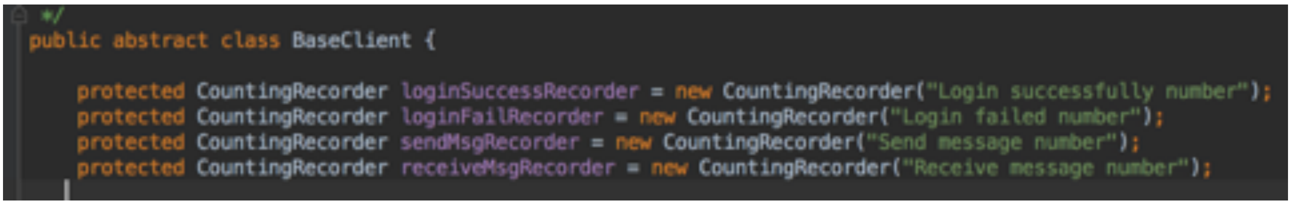
4.可以定义多种文件写入形式，如在写入周期创建新文件写入、在同一个文件中追加写入。

三、重构实现

1.删除原项目中的clientLogger类和serverLogger类，将新添加的构件的一系列监控指标CountingRecorder作为client类和server类的成员变量。



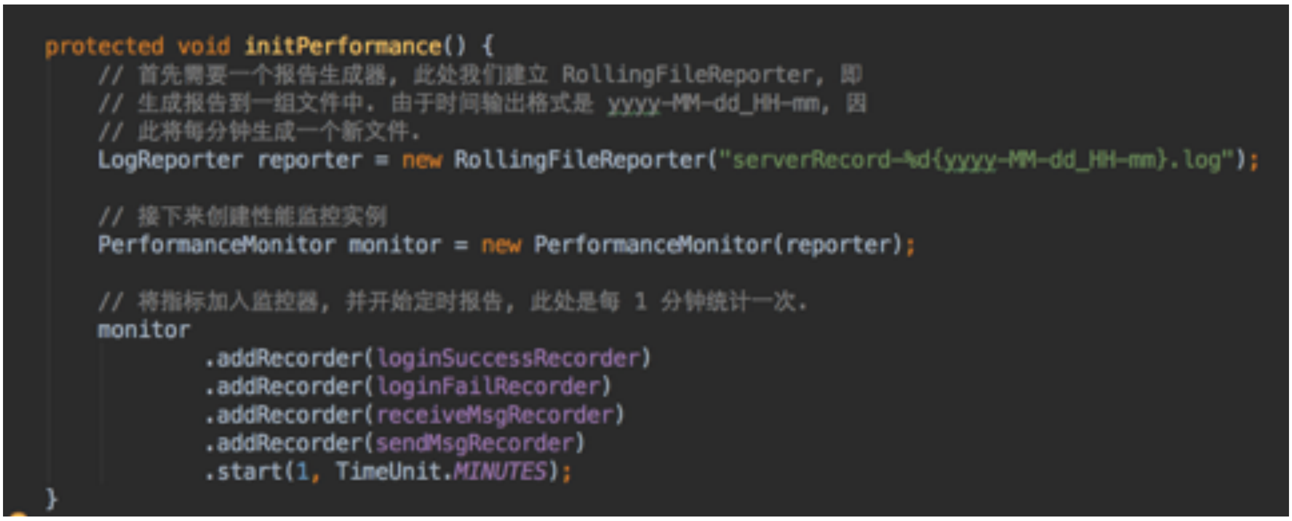
```
 */
public abstract class BaseServer {
    public CountingRecorder validLoginRecorder = new CountingRecorder("Valid Login Number");
    public CountingRecorder invalidLoginRecorder = new CountingRecorder("Invalid Login Number");
    public CountingRecorder receiveMsgRecorder = new CountingRecorder("Receive Message Number");
    public CountingRecorder ignoreMsgRecorder = new CountingRecorder("Ignore Message Number");
    public CountingRecorder forwardMsgRecorder = new CountingRecorder("Forward Message Number");
```



```
 */
public abstract class BaseClient {

    protected CountingRecorder loginSuccessRecorder = new CountingRecorder("Login successfully number");
    protected CountingRecorder loginFailRecorder = new CountingRecorder("Login failed number");
    protected CountingRecorder sendMsgRecorder = new CountingRecorder("Send message number");
    protected CountingRecorder receiveMsgRecorder = new CountingRecorder("Receive message number");
```

2.在client类初始化时初始化报告器LogReporter和监控器PerformanceMonitor。

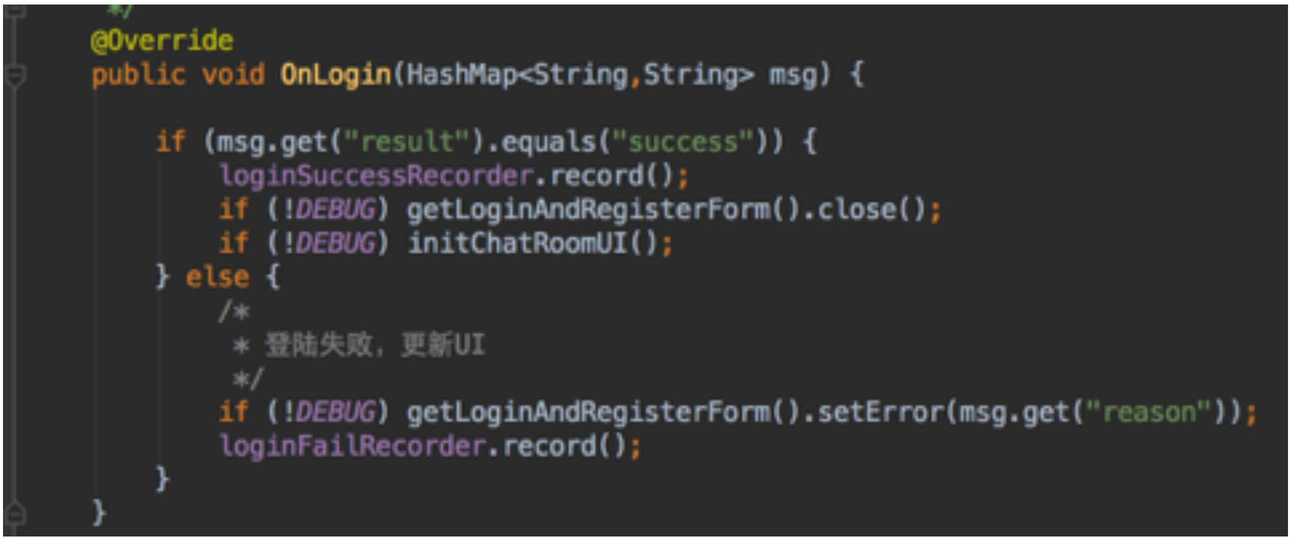


```
protected void initPerformance() {
    // 首先需要一个报告生成器，此处我们建立 RollingFileReporter，即
    // 生成报告到一组文件中。由于时间输出格式是 yyyy-MM-dd_HH-mm，因
    // 此将每分钟生成一个新文件。
    LogReporter reporter = new RollingFileReporter("serverRecord-%d{yyyy-MM-dd_HH-mm}.log");

    // 接下来创建性能监控实例
    PerformanceMonitor monitor = new PerformanceMonitor(reporter);

    // 将指标加入监控器，并开始定时报告，此处是每 1 分钟统计一次。
    monitor
            .addRecorder(loginSuccessRecorder)
            .addRecorder(loginFailRecorder)
            .addRecorder(receiveMsgRecorder)
            .addRecorder(sendMsgRecorder)
            .start(1, TimeUnit.MINUTES);
}
```

3.在相应的函数中调用相应的CountingRecorder.record()记录相应的指标。



```
     */
    @Override
    public void OnLogin(HashMap<String,String> msg) {

        if (msg.get("result").equals("success")) {
            loginSuccessRecorder.record();
            if (!DEBUG) getLoginAndRegisterForm().close();
            if (!DEBUG) initChatRoomUI();
        } else {
            /*
             * 登陆失败，更新UI
             */
            if (!DEBUG) getLoginAndRegisterForm().setError(msg.get("reason"));
            loginFailRecorder.record();
        }
    }
```

四、改进意见

1. 希望可以加入自定义输出格式的功能。