# Solar端集成通知模块详细设计文档

## 概要

根据solarMgr告警通知文档，编写solar端集成通知模块，设计通知功能模块化，表明需要实现的接口，开放公用的方法，事件模块设计，包括各个模块（任务、客户端）有哪些事件，触发机制、触发规则如何设计。

## 实现过程

对于整个流程，主要是实现过程大致为：模块监控（任务状态异常/客户端异常等...）-->

模块解析 --> 事件模块 --> 通知模块 --> 消息发送。

其中模块监控主要是：对需要解析得事件进行监控，监控方式结合solar现有流程进行结合，如果现有的逻辑满足不了监控，可对solar进行增加。

模块解析得主要作用是：接受各个模块监控触发时传入的数据，进行判断下发到事件模块。其对事件触发传过来得数据进行统一化。

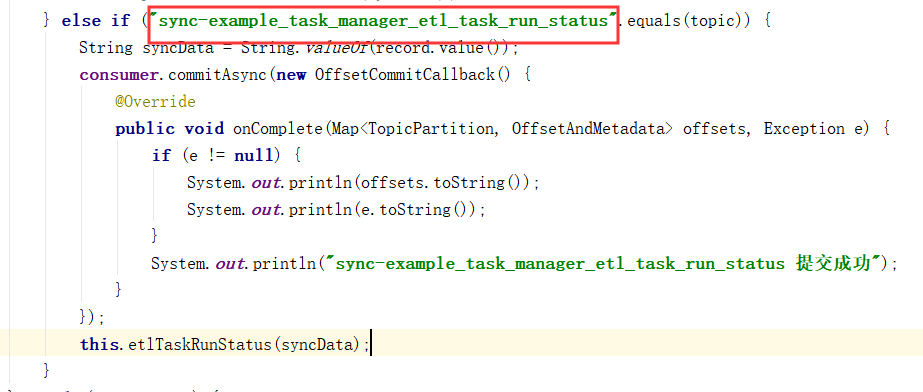
事件模块将接收模块解析过来的数据，对数据进一步精细化解析，不同于模块解析在于，事件模块设计将精细区分各个模块（任务/客户端/队列...）,处理发送内容等，然后将处理好得数据进行下发准备。

通知模块开放sendMsg方法接收参数。通知模块包括通知类型判断、通知实现类获取、通知模版解析、消息发送

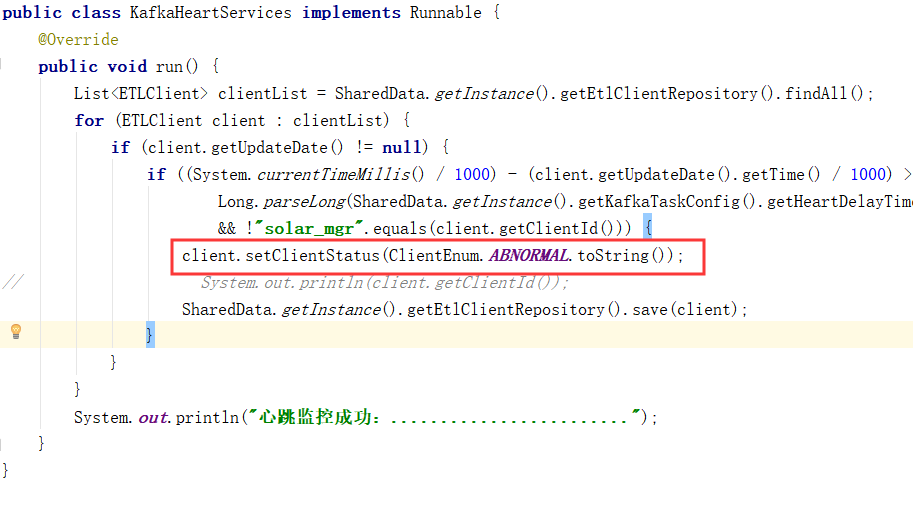
最后会将发送记录保存在数据库中作为历史记录。

### 模块监控

从现有得业务分析，首先实现任务状态监控，客户端状态监控。

1. 任务状态监控可从现有bin-log日志监控，监控任务状态中触发。

当任务状态发生变化时，获取任务状态id，通知来源为task。调用模块解析（ModelAnalysis）。

2、客户端监控可从客户端监控中获取。

当客户端两分钟没有状态修改，监控将被触发，获取客户端id，通知来源（客户端client），调用模块解析（ModelAnalysis）。

### 模块解析（ModelAnalysis）

模块解析获取模块触发来的数据：接收id，通知来源（客户端client/任务task等......）。根据通知来源，判断需要从那张表获取当前监听模块的状态，从而获取事件类型（FAIL/SUCCESS/TIMEOUT...）。如果通知来源时客户端（client），则从etl\_client获取客户端状态；如果是任务（task），则从etl\_task\_run\_status中获取当前任务状态；若是其他来源，后期可扩展类型......

将获取的事件类型（FAIL/SUCCESS/TIMEOUT...）,结合id、通知来源，调用事件模块设计（EventService）的getModelEvent方法。

### 事件模块（EventService）

用于接收模块解析消息，事件模块设计如下：

1. 接收模块解析消息，传入参数主要事件id（masterId）,事件类型（FAIL/SUCCESS/TIMEOUT...），通知来源（task/client...）。
2. 根据事件类型获取通知规则表（inform\_rule\_table），获取通知类型，判断通知类型，如果不通知返回，如果是首次通知，获取连续通知，或者通知，进行下一步。
3. 在getModelEvent方法中，根据通知来源，将任务分发到不同的模块，如果是任务状态，调用任务发送；如果是客户端，调用客户端。传入参数主要事件id（masterId）、事件类型（FAIL/SUCCESS/TIMEOUT...）、通知规则表（inform\_rule\_table）。各个模块在事件模块中。

#### 任务模块（ taskModel(id,type) ）

1. 任务模块接收参数任务id，事件类型（FAIL/SUCCESS/TIMEOUT...）、通知规则表（inform\_rule\_table）。
2. 获取通知规则表中得任务模板（ContentTemp）。
3. 创建Map，在Map中添加：masterId-->主要id（masterId）,eventType-->事件类型，createDate-->当前时间。
4. 根据任务模板，Map参数调用通知解析模板方法。获取待发送消息体（message）。
5. 调用通知模块的发送方法，传入通知规则表（inform\_rule\_table）、通知消息（message）。

#### 客户端模块（ clientModel(id,type) ）

1. 客户端模块接收参数客户端id，事件类型（FAIL/SUCCESS/TIMEOUT...）、通知规则表（inform\_rule\_table）。
2. 获取通知规则表中得任务模板（ContentTemp）。
3. 创建Map，在Map中添加：masterId-->主要id（masterId）,eventType-->事件类型，createDate-->当前时间。
4. 根据任务模板，Map参数调用通知解析模板方法。获取待发送消息体（message）。
5. 调用通知模块的发送方法，传入通知规则表（inform\_rule\_table）、通知消息（message）。

### 通知模块（InformModel）

发送方法（sendMsg）:

1. 接收事件模块里面的解析，获取通知规则表（inform\_rule\_table）、通知消息（message）。
2. 根据通知规则表，获取通知类型（邮箱/短信...）。
3. 如果是邮件通知，调用邮件整合方法（setEmailMsg）。传入通知规则表（inform\_rule\_table）、通知消息（message）；如果是其他通知方法，后续补充。。。

### 邮件发送实现类（SendEmailService）

#### 邮件整合方法（setEmailMsg）

1. 获取传入通知规则表（inform\_rule\_table）、通知消息（message）。
2. 根据通知规则表获取发送邮件的系统用户。
3. 获取接受通知得用户集合，获取用户集合得邮箱，形成集合（List<Emails>）。
4. 调用邮件发送方法（sendEmail）,传入发送通知用户的账户，密码，发送消息，接收用户集合。
5. 将下发得邮件保存在数据库（inform\_message\_log）中。

#### 邮件发送方法（sendEmail）

用于邮件发送实现，设计支持发送同时发送多人。

1. 设计获取发送邮件的配置文件方法（getProperties）。
2. 发送邮件方法（sendEmail），传入发送邮件的系统用户的账号account，密码password，接收者集合（List<userAccount>）,邮件内容。编写邮件下发流程，待需要发送邮件时调用改方法。

### 通知模板解析（ContentTempAnalysis）

解析方法放在现有得Utils中。

1. 传入通知模板（contentTemp），待替换字段（map）。
2. 判断通知模板是否包含占位符，如果有则进行后续步骤，如果没有则直接返回通知模板，作为通知信息。
3. 解析任务模板，替换其中的占位符（暂定 ’${}$’ 格式），将占位符形成集合。
4. 遍历占位符，判断占位符类型，从map中获取数据。
5. 将解析完的通知模板返回作为通知信息。

### 事件类型

根据模块进行划分：数据库中，各个模块可能出现的任务状态。

任务模块：等待（WATITING），创建（CEREATE），初始化（INIT），执行中（EXEC），成功（SUCCESS），失败（FAIL）。

客户端：空闲（FREE），忙碌（BUSY），异常（ABNORMAL）。

Solar：存活（ALIVE）。

任务队列：成功（SUCCESS），失败（FAILED），执行中（RUNING），等待（WAITING）。

### 触发规则

1. 将所有的事件类型录入在通知规则表中。
2. 将需要通知的事件类型设置为通知。
3. 配置通知模板。