

消息模板

- 1 维护消息模板
- 2 SDK/API 参数说明
- 3 发送消息SDK
 - 3.1 api sys/
- 4 消息系统架构设计
- 5 发送失败处理
- 6 前端交互格式

本文档描述了在第三代结算系统中，发送消息的方法。发送消息的类型，包括在线消息，邮件；以后会添加短信，或者微信等类型。

维护消息模板

表名: sys_template

说明: 用户表

主键: template_id

唯一索引:

表结构详细说明:

| 字段名称 | 字段类型 | 是否可以为空 | 注释 |
|--------------------|----------------|--------|------------------|
| template_id | VARCHAR2(50) | N | 自行添加，唯一 |
| msg_title | VARCHAR2(100) | N | 例：关于{参数注释}的通知 |
| msg_content | VARCHAR2(1000) | Y | 例：统计数量{参数注释} xxx |
| feature_code | | Y | 选择功能用于获得URL |
| module_code | VARCHAR2(50) | | 子系统 |
| remark | | | 说明 |
| send_type | | | 0, E |
| created_by | VARCHAR2(50) | N | 创建用户 |
| created_date | DATE | N | 创建时间 |
| last_modified_by | VARCHAR2(50) | Y | 修改用户 |
| last_modified_date | DATE | Y | 修改时间 |

msg_title、msg_content 支持大括号 {} 做参数占位符，支持多个。

send_type 字符串类型，是List 的json串。 如： ['0','E']

template_id 自行编写，唯一。

SDK/API 参数说明

* sendType Set<ArafSendType> 发送类型SendType E, 0, S E=邮件 0=在线 S=短信
* msgType 信息类型， WORK_NOTICE, WARNING, SYSTEM_MESSAGE, FOLLOW ArafMessageEnums ArafMessageType

```
* templateId 模板id
* toUsers toUsers Set类型
* ccUsers ccUsers Set类型
* titleParam titleParam List类型 与参数占位符一一 对应
* contentParam contentParam List类型 与参数占位符一一 对应
* urlParam urlParam
* attachment attachment
```

发送消息SDK

```
@Autowired
private MessageService messageService;
```

```
ArafMessage dto = of(ArafMessageType.WORK_NOTICE, templateId, toUser);
dto.setUrlParam("?path=" + filePath); // 在线
dto.setAttachment(filePath);
dto.getTitleParams().add(fileName);
dto.getContentParams().add(fileName);
sendMessage(dto);
```

配置发送邮件。

```
dto.setEmailConfig(new ArafEmailConfig(true, "10.1.3.xx", "opra", "Acxx4", "opra@acca.com.cn"));
sendMessage(dto);
```

或

```
sendMessage(dto, new ArafEmailConfig(true, "10.1.3.xx", "opra", "Acxx4", "opra@acca.com.cn"));
```

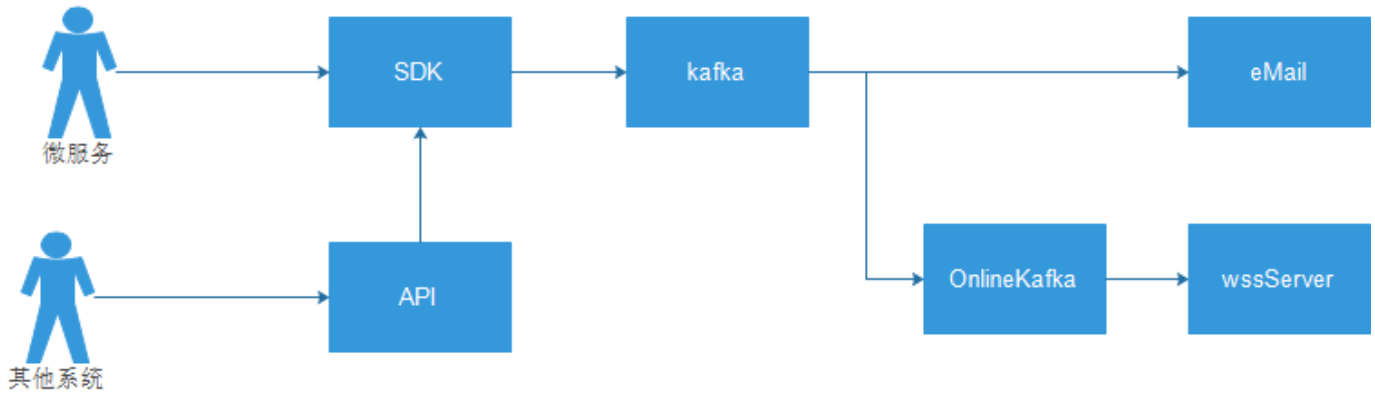
添加maven依赖

```
<dependency>
    <groupId>com.acca.opra</groupId>
    <artifactId>araf-core</artifactId>
</dependency>
```

api sys/

```
@PostMapping("/sendMessage")
ArafMessage sendMessage(@RequestBody ArafMessage dto);
```

消息系统架构设计



发送失败处理

系统会记录发送日志，可以通过界面对发送失败记录进行重发。

制定发件人。调研，需要的信息。

前端交互格式

```
{"messageId": "xx", "body": "xx", "type": "xx"}
```

type: 取值: ANNOUNCEMENT (公告), WORK_NOTICE (工作通知), WARNING (预警), SYSTEM_MESSAGE (系统消息), FOLLOW (关注), LOGOUT (无条件退出), PING (心跳), LOGOUT_ME (判断body与当前的userTags相同退出),

LOGOUT_OTHER (判断body与当前的userTags不相同退出), USER_NULL (userTags无效), TEST (连通测试)

body: 当 type = WORK_NOTICE (工作通知), WARNING (预警), SYSTEM_MESSAGE (系统消息), FOLLOW (关注) 时, body是 SysMessage 对象 json 串

```
public class SysMessage extends AbstractAuditableEntity {  
    private static final long serialVersionUID = 1L;  
  
    @TableId(value = "message_id", type = IdType.ASSIGN_UUID)  
    private String messageId; 消息ID  
  
    @TableField("user_id")  
    private String userId; 用户ID  
  
    @TableField("readed")  
    private YNOption readed; 是否读过  
  
    @TableField("msg_title")  
    private String msgTitle; 消息标题  
  
    @TableField("msg_content")  
    private String msgContent; 消息内容  
  
    @TableField("feature_code")  
    private String featureCode; 功能编码  
  
    @TableField("module_code")  
    private String moduleCode; 模块编码  
  
    @TableField("url")  
    private String url; 砖取 URL  
  
    @TableField("attachment")  
    private String attachment; 附件编码列表  
  
    @TableField("submit_date")  
    private Date submitDate; 发送时间
```

```
@TableField("message_type")
private MessageType messageType; 消息类型
```

```
@TableField("page_id")
private String pageId;    //当前页面标识。 用户异步作业通知，判断页面使用。 pageId 由前端产生 通过 dto 发送。
}
```