**Project Basic Information:**

**CRL-Business Requirement Description: (**Customer**)**

1. System Features

|  |  |  |
| --- | --- | --- |
| Project Name: | PRJ13106\_ESD Alert Service | \* |
| Business Unit: | |  | | --- | |  | | \* |
| CRL Type: | PRJ13106 | \* |
| Requirement Description: | ESD Alert Service |  |

2． ESD Alert Service Portal

1. Home Page

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | Alert Service Portal | **Customer Requirement#** |  |
| **Use Case ID** | *PRJ13061-RUC-001* | | |
| **Description** | cid:image005.png@01D1B769.ED1A54F0 | | |
| **Output** |  | | |
| **Functions List** |  | | |
| 1 | * 目前Menu包含   Trade In //一级  ESD Digital //一级  DropShip Download //二级  Order Process //二级  Sales Channel //一级  JET //二级  Google //二级   * 点击一级菜单要检索出所有一级菜单所有的Template显示在Gird中   即System=一级菜单   * 点击二级菜单要检索出所有一级和二级菜单对应的Template显示在Gird中 即System=一级菜单，Category=二级菜单   菜单数据源：  根据当前<http://10.16.75.24:3000/datastore/v1/esd/as_template>  集合中所有的System去重后为一级菜单  集合中所有的System对应的所有category去重后为二级菜单 | | |
| 2 | Add 按钮点击弹出Template的对话框.  当有选中菜单项的时候，弹出的Temlpate UI 要默认选中的System和Category | | |
| 3 | 查询结果：  <http://10.16.75.24:3000/datastore/v1/esd/as_template>   |  |  | | --- | --- | | System | as\_template. System | | Category | as\_template. Category | | Key | as\_template. Key | | Description | as\_template. Description | | Subject | as\_template. Subject | | From | as\_template. From | | To | as\_template. To | | CC | as\_template. CC | | InUser | as\_template. InUser | | InDate | as\_template. InDate | | ~~LastEditUser~~ | ~~as\_template. LastEditUser~~ | | ~~LastEditDate~~ | ~~as\_template. LastEditDate~~ | | Action | Edit:  弹出Template UI并加载当前行的Template数据  Delete:  弹出删除前提示对话框，确认后删除本行Template数据    删除完给出提示消息： | | | |
| 4 |  | | |
| 5 |  | | |

1. Add/Edit Template page

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | Alert Service Portal | **Customer Requirement#** |  |
| **Use Case ID** | *PRJ13106-RUC-001* | | |
| **Description** | cid:image006.png@01D1B769.ED1A54F0 | | |
| **Output** |  | | |
| **Functions List** |  | | |
| 1 | Add 模式时候：  默认状态：  UI 所有输入默认值为空，key 有提示文字[System-Category-Description]；  对话框的Title为：Add Alert Service Template  2个按钮为：Cancel和Create  点击Create按钮：  1.UI输入验证     |  |  | | --- | --- | | System | 必输；最大长度100 | | Category | ~~必输；~~最大长度100 | | Key | 必输；最大长度300 | | Description | 必输；最大长度255 | | Subject | 必输；最大长度255 | | From | 必输；长度最大100；Email Address | | To | 必输；长度最大100；Email Address；多Email | | CC | 长度最大255；Email Address；多Email | | ContentTemplate | 必输 |   2.点击Save 按钮后将数据保存至  <http://10.16.75.24:3000/datastore/v1/esd/as_template>  保存前需要做Key重复性检查，如果存在同样的Key不保存，给出提示消息.   1. ContentTemplate 保存为HTML格式  |  |  | | --- | --- | | as\_template. System | System | | as\_template. Category | Category | | as\_template. Key | Key | | as\_template. Description | Description | | as\_template. Subject | Subject | | as\_template. From | From | | as\_template. To | To | | as\_template. CC | CC | | as\_template. Template | ContentTemplate | | as\_template. InUser | ESD | | as\_template. InDate | 当前日期 | | as\_template. LastEditUser | Null | | as\_template. LastEditDate | Null | |  |  |  1. 保存成功后给出提示消息 | | |
| 2 | Edit模式时候：   1. 需要根据Template ID查询出Template并显示 2. Key可以修改, 修改的Key 不能和已经存在的key重复(除过自身) 3. 2个按钮为：Cancel和Update，在Update的时候做UI输入检查 4. 更新Template数据到   <http://10.16.75.24:3000/datastore/v1/esd/as_template>   1. ContentTemplate 保存为HTML格式 2. 对话框的Title为Edit：+key  |  |  | | --- | --- | | as\_template. System | System | | as\_template. Category | Category | | as\_template. Key | Key | | as\_template. Description | Description | | as\_template. Subject | Subject | | as\_template. From | From | | as\_template. To | To | | as\_template. CC | CC | | as\_template. Template | ContentTemplate | | as\_template. InUser | InUser | | as\_template. InDate | InDate | | as\_template. LastEditUser | 当前访问用户域短名 | | as\_template. LastEditDate | 当前日期 | |  |  |  1. 更新成功后给出提示消息 | | |
|  |  | | |

1. 邮件发送服务API

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | Alert Service Server | **Customer Requirement#** |  |
| **Use Case ID** | *PRJ13061-RUC-003* | | |
| **Description** | 提供一个API给MQ调用，然后取出在界面编辑好的模板和参数，和MQ发送过来的数据合并，构建成邮件发送出去。 | | |
| **Output** |  | | |
| **Functions List** |  | | |
| 1 | 提供一个API，供MQ调用，POST请求，Body必须符合如下规范：   |  | | --- | | {  "Key": "SalesChannel-JET-002200",//调用客户端赋值，界面定义的Key  "Server": "10.16.234.41",//调用客户端取到的机器名  "Date": "2016-01-01",//调用客户端取到的日期  "Data": {  //这里请随意  }  } |   API GDEV地址是：10.16.230.41:809/api/AlertService  MQ地址是： | | |
| 2 | API会从Kango上面根据传过来的key取邮件模板相关参数，Kango地址是10.16.75.24:3000/datastore/v1/esdtest/templates，后面两个参数分别是store名和Collection名，可以在Portal中修改。访问需要加自定义头：x-cloud-pwd：4esduse0nly  返回的数据格式应该符合下面规范：   |  | | --- | | {  "Key": "SalesChannel-JET-002200",  "Template": "",  "Subject": "(Info)ESD Third Sales Channel - Order Validate Job Exception ",  "System": "SalesChannel",  "Category": "JET",  "Description": "description",  "From": "ESDGroup@newegg.com",  "To": "Neal.Y.SHI@newegg.com",  "CC": "Tomas.F.Yang@newegg.com",  "Action": null  } |   这些信息都是通过界面维护。 | | |
| 3 | 界面邮件模板编辑请参考《模板编辑说明.docx》  固定系统变量使用$$ 举例如下:   |  |  | | --- | --- | | XXXX on $$server$$ | XXXX on 10.16.234.41 | | Sent at $$Date$$ | Sent at 06/22/2016 13:30:12 | | | |
| 4 | 调用API如果出现异常，会记录日志，日志地址在：  D:\Application\ESD\ESDConfiguration\Log\AlertService | | |
| 5 | ~~如果调用时Body为空，返回调用失败“Json content can not be empty.”；~~  如果调用时传入的Key找不到，返回调用失败“"Key:XXX does not exist in kango.”；  调用异常会调用平台会发送异常邮件给配置好的人。 | | |

1. 发送MQ数据的DLL文件

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | Alert Service Client Lib | **Customer Requirement#** |  |
| **Use Case ID** | *PRJ13061-RUC-004* | | |
| **Description** | 提供一个Dll，此Dll支持两个接口，实现将调用者的数据发送到MQ队列中。 | | |
| **Output** |  | | |
| **Functions List** |  | | |
| 1 | MQ提供API接收数据，调用此API需使用http post方法，RequestBody必须符合以下格式。   |  | | --- | | {  "MessageName": "ESDMessage",//固定不变  "MessageBody": “{  \"Date\":Getdate(), //取当前时间  \"Server\":HostName(),//取本机名  \"Key\":key, //调用方传入的参数key  \"Data\":message //调用方传入的对象序列化为json字符串  }”,  "ContentType": "application/json",//固定不变  "InvokeType": "Message",//固定不变  "MessageHeaders": [  {  "HeaderName": "Authorization",//固定不变  "HeaderValue": "102c2064fcefe3952a2d15322e79b042%2613ae4484480c68d" //此值为gedv环境  },  {  "HeaderName": "Content-Type"",//固定不变  "HeaderValue": "application/json"//固定不变  }  ]  }  MQ API 地址：  DEV：<http://10.16.75.24:3000/framework/v1/enterprise-messaging/message>  GQC：<http://10.1.24.145:3000/framework/v1/enterprise-messaging/message>  PRD：<http://apis.newegg.org/framework/v1/enterprise-messaging/message> | | | |
| 2 | Dll文件提供两个接口供外部调用   1. SendAlertMail (string key, Object notifyObject)   功能：将传入的notifyObject序列化为json字符串，之后将此字符串赋值给RequestBody中的[MessageBody.Data]，将参数key赋值给RequestBody中的[MessageBody.key]。之后用post方法将生成的RequestBody发送到MQ API。   1. SendAlertMail(string key, object notifyObject, params string[] notifyProperties)   功能： 将notifyObject序列化为json字符串，而notifyProperties表示在序列化时保留哪些节点。之后与接口1一样赋值给RequestBody并post到MQ API。 | | |
| 3 | 将数据发送给MQ API时如果出现异常则会记录到日志文件中。  文件名：ALertServiceError\_+日期.txt ，例如ALertServiceError\_2016-06-15.txt  文件路径：当前目录下  只有发生异常时才记录；若文件已存在则在文件中追加内容；若文件不存在则创建。  内容格式：   |  | | --- | | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Exception {当前时间}\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  异常信息 |   例如：   |  | | --- | | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Exception 2016/6/15 15:39:12\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  System.Net.WebException: The remote server returned an error: (502) Bad Gateway.  at System.Net.HttpWebRequest.GetResponse()  at Newegg.ESD.AlertService.ClientLib.AlertAssistant.SendToMQ(String requestBody) in D:\git\ESDSendMessageToMQ\ClientLib\AlertAssistant.cs:line 168  at Newegg.ESD.AlertService.ClientLib.AlertAssistant.RetrySend(MQRequestBody mqRequestBody) in D:\git\ESDSendMessageToMQ\ClientLib\AlertAssistant.cs:line 140 | | | |
| 4. | 若调用MQAPI成功，可以在Web Console上查看到已上传到MQ的数据。  Web Console地址  GDEV： <http://10.16.75.26:8080/kafkawebconsole/main.action>  GQC：<http://10.1.24.174:8180/kafkawebconsole/main.action>  PRD：<http://10.1.54.120:8200/kafkawebconsole/main.action> | | |