

ALIBABA CLOUD

阿里云

事件总线
最佳实践

文档版本：20201125

 阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置>网络>设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.云视频会议事件	05
2.自定义应用事件	08
3.计算PutEvents事件大小	14

1.云视频会议事件

本教程以云视频会议开始或结束事件为例，说明事件总线EventBridge如何根据事件规则将云视频会议事件路由到指定的HTTP网关，从而实现事件驱动架构。

前提条件

您已完成以下操作：

1. [开通事件总线EventBridge服务](#)
2. [开通云视频会议](#)

背景信息

- 事件驱动架构：以事件驱动架构设计和管理您的服务可以很好的实现服务模块之间解耦，提高服务整体的稳定性和灵活性。所有的业务数据都可以映射到事件上来，所有的业务事件按照业务领域划分为多个事件类型。详情请参见[事件驱动架构（Event Driven Architecture, EDA）](#)。
- 云视频会议：云视频会议是阿里云提供的开放、可靠、智能的会议PaaS服务，基于阿里集团数年视频会议技术及应用积累、全球部署的网络节点覆盖及领先的AI技术，您可快速构建多端互通与简单易用的会议协作应用。

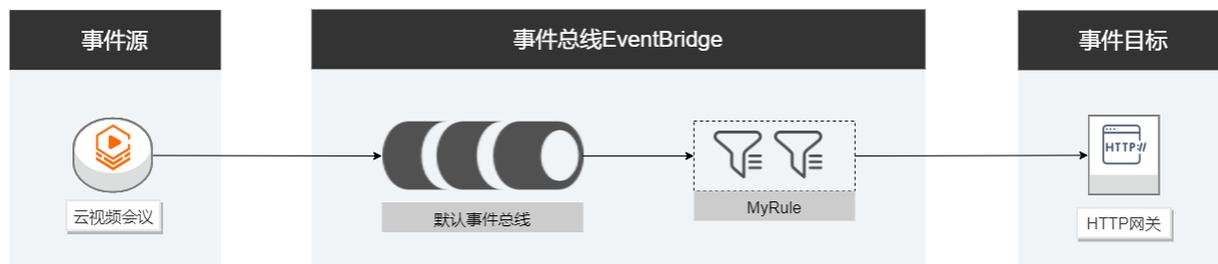
云视频会议定义的事件类型如下表所示。

事件类型	事件类型描述	事件触发场景
aliyuncvc:MeetingEvent:MeetingStateEvent	会议开始或结束事件	<ul style="list-style-type: none">○ 会议开始，即从第一个人进入会议，推送“时间、MeetingUUID、MeetingName、action”。○ 会议结束，即从最后一个人离开会议（不论是主动还是被动），推送“时间、MeetingUUID、MeetingName、action”。
aliyuncvc:MeetingEvent:MemberStateEvent	会议成员状态事件	<ul style="list-style-type: none">○ 成员入会，推送“时间、MeetingUUID、MeetingName、userID、groupId、action”。○ 成员离会，推送“时间、MeetingUUID、MeetingName、userID、groupId、action”。

事件类型	事件类型描述	事件触发场景
aliyuncvc:MeetingEvent:Member Operate	会议成员会中操作事件	<ul style="list-style-type: none"> 静音或取消静音，推送“时间、MeetingUUID、MeetingName、userID、groupId、action”。 打开或关闭扬声器，推送“时间、MeetingUUID、MeetingName、userID、groupId、action”。 打开或关闭摄像头，推送“时间、MeetingUUID、MeetingName、userID、groupId、action”。 打开或关闭共享屏幕，推送“时间、MeetingUUID、MeetingName、userID、groupId、action”。
aliyuncvc:MeetingEvent:Meeting Statistic	会议统计数据发布事件	每天推送一次最高并发。

云视频会议事件通知事件流

本文以云视频会议为例，为您介绍阿里云服务接入事件总线EventBridge的流程。



步骤一：创建事件规则

云视频会议事件会发布到云服务专用总线。事件总线EventBridge为您内置了云服务专用总线，您无需创建事件总线。为云服务专用总线创建事件规则的步骤如下：

1. 登录[事件总线EventBridge控制台](#)。
2. 在顶部菜单栏，选择地域。
3. 在左侧导航栏，单击[云服务专用总线](#)。
4. 在规则管理页面，单击创建规则。
5. 在创建规则页面，完成以下操作。
 - i. 在配置基本信息页面，在名称文本框输入 *MyRule*，在描述文本框输入 *将云视频会议开始或结束事件路由到指定HTTP网关。*，然后单击下一步。

- ii. 在配置事件模式页面，事件模式类型选择阿里云服务专用事件模式，从阿里云服务列表选择云视频会议，从事件类型列表选择会议状态，然后单击下一步。
- iii. 在配置事件目标页面，选择目标服务类型。
 - 服务类型：单击HTTP。
 - URL：输入地址 `http://api.xtingke.com/eventBridge/processEvent`。
 - Body：单击完整事件。

 注意 目标服务和事件规则必须处于同一地域。

- iv. 单击创建。

步骤二：使用HTTP网关接收事件

1. 使用HTTP网关接收事件的示例代码如下。

```
import lombok.extern.slf4j.Slf4j;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.ResponseBody;
@Controller
@SpringBootApplication
@Slf4j
public class EventProcessingApplication {
    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
    @PostMapping("/eventBridge/processEvent")
    @ResponseBody
    public String reviewMessage(@RequestBody String data) {
        log.info("reviewEvent");
        log.info(data);
        return "recieved";
    }
}
```

2. 自定义应用事件

您可以使用事件总线EventBridge的自定义总线接收您自己的应用生产的事件，并将其路由至目标阿里云服务。本教程以客户关系管理CRM（Customer Relation Management）系统为例，说明如何将营销相关事件发布到事件总线EventBridge，然后利用事件规则过滤出需要关心的用户注册或登录事件，并将其路由至消息服务MNS的队列。

前提条件

您已完成以下操作：

1. [开通事件总线EventBridge服务](#)
2. [开通MNS服务](#)
3. [创建队列](#)

背景信息

本文以CRM系统为例介绍自定义事件如何接入事件总线EventBridge，从而构造事件驱动架构。

- CRM系统：客户信息收集、管理、分析和利用的信息系统。该系统记录了企业在市场营销过程中和客户发生的各种交互行为，以及各类有关活动的状态。如需接入事件总线EventBridge，该CRM系统需要根据[CloudEvents 1.0](#)协议对营销相关事件的属性和数据进行描述。CRM系统的营销相关事件定义示例如下：

名称	示例值	描述
source	crmabc.newsletter	事件源。长度最大128字节。
type	UserSignUp, UserLogin	事件类型。长度最大64字节。取值： <ul style="list-style-type: none"> ◦ <i>UserPayOff</i>: 用户支付。 ◦ <i>UserLogin</i>: 用户登录。 ◦ <i>UserSignUp</i>: 用户注册。
subject	crmabc/users/1234345	事件主题。长度最大128字节。
data	{ "E-Mail": "{\$email}" }	事件内容。自定义事件的事件附加数据。

- 事件驱动架构：以事件驱动架构设计和管理您的服务可以很好的实现服务模块之间解耦，提高服务整体的稳定性和灵活性。所有的业务数据都可以映射到事件上来，所有的业务事件按照业务领域划分为多个事件类型。详情请参见[事件驱动架构（Event Driven Architecture, EDA）](#)。

步骤一：创建自定义总线

创建自定义总线marketing，用于接收CRM系统的营销相关事件。

1. 登录[事件总线EventBridge控制台](#)。
2. 在顶部菜单栏，选择地域。
3. 在左侧导航栏，单击自定义总线。
4. 在自定义总线页面，单击创建自定义总线。
5. 在创建事件总线面板，在名称文本框输入marketing，在描述文本框输入用于接收CRM系统的营销相关事件。，然后单击确定。

步骤二：使用SDK发布事件

使用SDK将CRM系统的营销相关事件发布到事件总线Event Bridge。

1. 添加Maven依赖。Maven依赖的示例代码如下：

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>eventbridge-client</artifactId>
  <version>1.2.3</version>
</dependency>
```

2. 发布事件。发布事件的示例代码如下：

```
import java.net.URI;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import com.aliyun.eventbridge.EventBridge;
import com.aliyun.eventbridge.EventBridgeClient;
import com.aliyun.eventbridge.models.CloudEvent;
import com.aliyun.eventbridge.models.Config;
import com.aliyun.eventbridge.models.PutEventsResponse;
import com.aliyun.eventbridge.util.EventBuilder;
import com.google.gson.Gson;
public class PutEventsSample {
  private final EventBridge eventBridgeClient;
  public PutEventsSample() {
    Config authConfig = new Config();
    authConfig.accessKeyId = "{accessKeyId}";
    authConfig.accessKeySecret = "{accessKeySecret}";
    authConfig.endpoint = "{endpoint}";
    eventBridgeClient = new EventBridgeClient(authConfig);
  }
  public void putEventsSample() {
    List<CloudEvent> cloudEventList = new ArrayList<CloudEvent>();
    cloudEventList.add(EventBuilder.builder()
      .withId("9e9b433c-a89b-4918-896b-7e1b7221****")
      .withSource(URI.create("crmabc.newsletter"))
      .withType("UserSignUp")
      .withSubject("crmabc/users/1234345")
      .withTime(new Date())
      .withJsonStringData("{\"E-Mail\": \"{email}\"}")
      .withAliyunEventBus("marketing")
```

```

        .build());
    PutEventsResponse putEventsResponse = eventBridgeClient.putEvents(cloudEventList);
    System.out.println(new Gson().toJson(putEventsResponse));
}
public static void main(String[] args){
    PutEventsSample sample = new PutEventsSample();
    try {
        sample.putEventsSample();
    }
    catch (Throwable e) {
        e.printStackTrace();
    }
}
}
}

```

参数	描述
accessKeyId	阿里云账号的AccessKey ID。
accessKeySecret	阿里云账号的AccessKey Secret。
endpoint	事件总线EventBridge服务接入点，在控制台的自定义总线页面右上角单击查看接入点获取。

步骤三：创建事件规则

创建事件规则过滤并路由您关心的营销相关事件。

1. 在左侧导航栏，单击自定义总线。
2. 在自定义总线页面，找到目标总线，在其右侧操作列单击规则管理。
3. 在规则管理页面，单击创建规则。
4. 在创建规则页面，完成以下操作。
 - i. 在配置基本信息页面，在名称文本框输入 *MyCustomRule*，在描述文本框输入 *将CRM系统的营销相关事件路由到消息服务MNS。*，然后单击下一步。

- ii. 在配置事件模式页面，事件模式类型选择自定义事件模式，在事件模式内容代码框输入事件模式，然后单击下一步。过滤出需要关心的用户注册或登录事件的事件模式内容示例如下所示。

```
{
  "source": [
    "crmabc.newsletter"
  ],
  "type": [
    "UserSignUp", "UserLogin"
  ]
}
```

- iii. 在配置事件目标页面，选择目标服务类型。
 - 服务类型：单击消息服务。
 - 队列：选择MyQueue。
 - 消息内容：单击完整事件。
- iv. 单击创建。

结果验证

您可以在消息服务MNS控制台确认是否接收到关心的营销相关事件。

1. 登录[消息服务MNS控制台](#)。
2. 在顶部菜单栏，选择地域。
3. 在左侧导航栏，单击队列。
4. 在队列页面，在队列名称列找到目标队列MyQueue，在其右侧操作列单击接收消息。
5. 在接收消息对话框，清空Base64解码，然后单击接收消息。

接收消息 ✕

目的队列名称: MyQueue

消息句柄 [?]: 8-1zuT42PL4zuz8zcEz0z8Nmza8RntkvxqET 复制

消息创建时间 [?]: 2020-06-04 20:07:13

被消费次数 [?]: 1

首次消费时间 [?]: 2020-06-04 20:57:37

下次消费时间 [?]: 2020-06-04 20:58:07

消息内容 [?]:

```
{ "eventId": "9e9b433c-a89b-4918-896b-7e1b72210a69", "publishTime": 1591272433527, "Message": { "data": { "E-Mail": { "email": "" }, "id": "9e9b433c-a89b-4918-896b-7e1b72210a69", "source": "crmabc.newsletter", "specversion": "1.0", "type": "UserSignUp", "subject": "crmabc/users/1234345", "time": "2020-06-04T20:07:13.000Z" } } }
```

 自动删除已接收消息 Base64解码

告警

如果当前操作的队列正被您的应用使用，请谨慎操作，以防您的应用消费异常。

接收消息 取消

接收到的消息内容示例如下：

```
{
  "eventId":"9e9b433c-a89b-4918-896b-7e1b7221****",
  "publishTime":1591272433527,
  "Message":{
    "data":{"E-Mail":{"email}}",
    "id":"9e9b433c-a89b-4918-896b-7e1b7221****",
    "source":"crmabc.newsletter",
    "specversion":"1.0",
    "type":"UserSignUp",
    "subject":"crmabc/users/1234345",
    "time":"2020-06-04T12:07:11.851Z"
  },
  "eventBusName":"marketing",
  "eventBusOwner":"<yourAccountId>",
  "ruleName":"MyCustomRule",
  "eventBusInvoker":"<yourAccountId>",
  "MessageMD5":"D0256972C35F85409E38D176B7E7****"
}
```

3. 计算PutEvents事件大小

本文介绍如何计算PutEvents事件大小，将多个事件添加到一个请求中以提高效率。

使用场景

您可以使用PutEvents操作添加自定义事件到事件总线Event Bridge中。您可以使用PutEvents添加多个事件，通过预先计算事件大小，使所有事件总大小不超过256 KB，然后将这些事件添加到一个请求中以提高效率。

以下提供了计算事件大小的方式。

计算方式

CloudEvent 大小计算方式如下：

- time：大小一般为36字节。
- specversion：以字符串UTF-8编码形式的字节数来度量。
- id：以字符串UTF-8编码形式的字节数来度量。
- type：以字符串UTF-8编码形式的字节数来度量。
- source：以字符串UTF-8编码形式的字节数来度量。
- subject：以字符串UTF-8编码形式的字节数来度量。
- dataschema：以字符串UTF-8编码形式的字节数来度量。
- datacontenttype：以字符串UTF-8编码形式的字节数来度量。
- data：按照byte[]的长度来度量。

示例代码

以下为计算单个CloudEvent 大小的Java示例代码：

```
int getSize(CloudEvent event) {
    int size = 0;
    if (event.getTime() != null) {
        size += 36;
    }
    size += event.getSpecversion().getBytes(StandardCharsets.UTF_8).length;
    size += event.getId().getBytes(StandardCharsets.UTF_8).length;
    size += event.getType().getBytes(StandardCharsets.UTF_8).length;
    size += event.getSource().toString().getBytes(StandardCharsets.UTF_8).length;
    size += event.getSubject().getBytes(StandardCharsets.UTF_8).length;
    size += event.getDataschema().toString().getBytes(StandardCharsets.UTF_8).length;
    size += event.getDatacontenttype().getBytes(StandardCharsets.UTF_8).length;
    size += event.getData().length;
    return size;
}
```