

阿里云

事件总线EventBridge  
SDK参考

文档版本：20201228

## 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<br>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<br>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<br>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<br>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><br><code>Instance_ID</code>                                              |
| [] 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. Aliyun Client SDK    | 11 |
| 1.1. SDK概述              | 11 |
| 1.2. Java SDK           | 11 |
| 1.2.1. 概述               | 11 |
| 1.2.2. 版本说明             | 12 |
| 1.2.3. 安装Java SDK       | 12 |
| 1.2.4. 快速入门             | 12 |
| 1.2.5. 事件总线             | 14 |
| 1.2.5.1. 事件总线概述         | 14 |
| 1.2.5.2. CreateEventBus | 14 |
| 1.2.5.3. GetEventBus    | 15 |
| 1.2.5.4. DeleteEventBus | 16 |
| 1.2.5.5. ListEventBuses | 17 |
| 1.2.6. 事件规则             | 18 |
| 1.2.6.1. 事件规则概述         | 18 |
| 1.2.6.2. CreateRule     | 19 |
| 1.2.6.3. GetRule        | 20 |
| 1.2.6.4. EnableRule     | 21 |
| 1.2.6.5. DisableRule    | 22 |
| 1.2.6.6. UpdateRule     | 23 |
| 1.2.6.7. ListRules      | 25 |
| 1.2.6.8. DeleteRule     | 26 |
| 1.2.7. 事件               | 27 |
| 1.2.7.1. 事件概述           | 27 |
| 1.2.7.2. PutEvents      | 27 |
| 1.2.8. 事件目标             | 28 |

|                         |    |
|-------------------------|----|
| 1.2.8.1. 事件目标概述         | 28 |
| 1.2.8.2. CreateTargets  | 28 |
| 1.2.8.3. ListTargets    | 30 |
| 1.2.8.4. DeleteTargets  | 31 |
| 1.3. Go SDK             | 32 |
| 1.3.1. 概述               | 33 |
| 1.3.2. 版本说明             | 33 |
| 1.3.3. 安装Go SDK         | 33 |
| 1.3.4. 快速入门             | 34 |
| 1.3.5. 事件总线             | 36 |
| 1.3.5.1. 事件总线概述         | 36 |
| 1.3.5.2. CreateEventBus | 36 |
| 1.3.5.3. GetEventBus    | 38 |
| 1.3.5.4. DeleteEventBus | 40 |
| 1.3.5.5. ListEventBuses | 41 |
| 1.3.6. 事件规则             | 43 |
| 1.3.6.1. 事件规则概述         | 43 |
| 1.3.6.2. CreateRule     | 43 |
| 1.3.6.3. GetRule        | 45 |
| 1.3.6.4. EnableRule     | 47 |
| 1.3.6.5. DisableRule    | 48 |
| 1.3.6.6. UpdateRule     | 50 |
| 1.3.6.7. ListRules      | 52 |
| 1.3.6.8. DeleteRule     | 53 |
| 1.3.7. 事件               | 55 |
| 1.3.7.1. 事件概述           | 55 |
| 1.3.7.2. PutEvents      | 55 |
| 1.3.8. 事件目标             | 57 |

|                         |    |
|-------------------------|----|
| 1.3.8.1. 事件目标概述         | 57 |
| 1.3.8.2. CreateTargets  | 57 |
| 1.3.8.3. ListTargets    | 59 |
| 1.3.8.4. DeleteTargets  | 61 |
| 1.4. Python SDK         | 63 |
| 1.4.1. 概述               | 63 |
| 1.4.2. 版本说明             | 63 |
| 1.4.3. 安装Python SDK     | 63 |
| 1.4.4. 快速入门             | 64 |
| 1.4.5. 事件总线             | 66 |
| 1.4.5.1. 事件总线概述         | 66 |
| 1.4.5.2. CreateEventBus | 66 |
| 1.4.5.3. GetEventBus    | 68 |
| 1.4.5.4. DeleteEventBus | 70 |
| 1.4.5.5. ListEventBuses | 72 |
| 1.4.6. 事件规则             | 74 |
| 1.4.6.1. 事件规则概述         | 74 |
| 1.4.6.2. CreateRule     | 74 |
| 1.4.6.3. GetRule        | 75 |
| 1.4.6.4. EnableRule     | 77 |
| 1.4.6.5. DisableRule    | 79 |
| 1.4.6.6. UpdateRule     | 81 |
| 1.4.6.7. ListRules      | 83 |
| 1.4.6.8. DeleteRule     | 84 |
| 1.4.7. 事件               | 87 |
| 1.4.7.1. 事件概述           | 87 |
| 1.4.7.2. PutEvents      | 87 |
| 1.4.8. 事件目标             | 88 |

|                         |     |
|-------------------------|-----|
| 1.4.8.1. 事件目标概述         | 88  |
| 1.4.8.2. CreateTargets  | 88  |
| 1.4.8.3. ListTargets    | 90  |
| 1.4.8.4. DeleteTargets  | 92  |
| 1.5. PHP SDK            | 93  |
| 1.5.1. 概述               | 93  |
| 1.5.2. 版本说明             | 93  |
| 1.5.3. 安装PHP SDK        | 94  |
| 1.5.4. 快速入门             | 94  |
| 1.5.5. 事件总线             | 96  |
| 1.5.5.1. 事件总线概述         | 96  |
| 1.5.5.2. CreateEventBus | 97  |
| 1.5.5.3. GetEventBus    | 98  |
| 1.5.5.4. DeleteEventBus | 100 |
| 1.5.5.5. ListEventBuses | 101 |
| 1.5.6. 事件规则             | 103 |
| 1.5.6.1. 事件规则概述         | 103 |
| 1.5.6.2. CreateRule     | 103 |
| 1.5.6.3. GetRule        | 105 |
| 1.5.6.4. EnableRule     | 106 |
| 1.5.6.5. DisableRule    | 108 |
| 1.5.6.6. UpdateRule     | 110 |
| 1.5.6.7. ListRules      | 111 |
| 1.5.6.8. DeleteRule     | 113 |
| 1.5.7. 事件               | 114 |
| 1.5.7.1. 事件概述           | 114 |
| 1.5.7.2. PutEvents      | 114 |
| 1.5.8. 事件目标             | 116 |

|                         |     |
|-------------------------|-----|
| 1.5.8.1. 事件目标概述         | 116 |
| 1.5.8.2. CreateTargets  | 117 |
| 1.5.8.3. ListTargets    | 118 |
| 1.5.8.4. DeleteTargets  | 120 |
| 1.6. C# SDK             | 122 |
| 1.6.1. 概述               | 122 |
| 1.6.2. 版本说明             | 122 |
| 1.6.3. 安装C# SDK         | 122 |
| 1.6.4. 快速入门             | 123 |
| 1.6.5. 事件总线             | 125 |
| 1.6.5.1. 事件总线概述         | 125 |
| 1.6.5.2. CreateEventBus | 125 |
| 1.6.5.3. GetEventBus    | 127 |
| 1.6.5.4. DeleteEventBus | 128 |
| 1.6.5.5. ListEventBuses | 130 |
| 1.6.6. 事件规则             | 131 |
| 1.6.6.1. 事件规则概述         | 131 |
| 1.6.6.2. CreateRule     | 131 |
| 1.6.6.3. GetRule        | 133 |
| 1.6.6.4. EnableRule     | 135 |
| 1.6.6.5. DisableRule    | 136 |
| 1.6.6.6. UpdateRule     | 138 |
| 1.6.6.7. ListRules      | 139 |
| 1.6.6.8. DeleteRule     | 141 |
| 1.6.7. 事件               | 142 |
| 1.6.7.1. 事件概述           | 142 |
| 1.6.7.2. PutEvents      | 142 |
| 1.6.8. 事件目标             | 144 |

|                         |     |
|-------------------------|-----|
| 1.6.8.1. 事件目标概述         | 144 |
| 1.6.8.2. CreateTargets  | 144 |
| 1.6.8.3. ListTargets    | 146 |
| 1.6.8.4. DeleteTargets  | 148 |
| 1.7. TypeScript SDK     | 149 |
| 1.7.1. 概述               | 149 |
| 1.7.2. 版本说明             | 150 |
| 1.7.3. 安装TypeScript SDK | 150 |
| 1.7.4. 快速入门             | 151 |
| 1.7.5. 事件总线             | 152 |
| 1.7.5.1. 事件总线概述         | 152 |
| 1.7.5.2. CreateEventBus | 152 |
| 1.7.5.3. GetEventBus    | 153 |
| 1.7.5.4. DeleteEventBus | 154 |
| 1.7.5.5. ListEventBuses | 155 |
| 1.7.6. 事件规则             | 157 |
| 1.7.6.1. 事件规则概述         | 157 |
| 1.7.6.2. CreateRule     | 157 |
| 1.7.6.3. GetRule        | 159 |
| 1.7.6.4. EnableRule     | 160 |
| 1.7.6.5. DisableRule    | 161 |
| 1.7.6.6. UpdateRule     | 163 |
| 1.7.6.7. ListRules      | 165 |
| 1.7.6.8. DeleteRule     | 166 |
| 1.7.7. 事件               | 167 |
| 1.7.7.1. 事件概述           | 168 |
| 1.7.7.2. PutEvents      | 168 |
| 1.7.8. 事件目标             | 169 |

---

|                          |     |
|--------------------------|-----|
| 1.7.8.1. 事件目标概述          | 169 |
| 1.7.8.2. CreateTargets   | 169 |
| 1.7.8.3. ListTargets     | 171 |
| 1.7.8.4. DeleteTargets   | 172 |
| 2.CloudEvents Client SDK | 175 |

# 1. Aliyun Client SDK

## 1.1. SDK概述

本文介绍事件总线EventBridge提供的多语言SDK。

### SDK

事件总线EventBridge提供以下语言的SDK源码和参考文档。

| 语言         | SDK源码                                  | 参考文档                         |
|------------|----------------------------------------|------------------------------|
| Java       | <a href="#">eventbrige-java-sdk</a>    | <a href="#">Java文档</a>       |
| Go         | <a href="#">eventbridge-go-sdk</a>     | <a href="#">Go文档</a>         |
| Python     | <a href="#">eventbridge-python-sdk</a> | <a href="#">Python文档</a>     |
| PHP        | <a href="#">eventbridge-php-sdk</a>    | <a href="#">PHP文档</a>        |
| C#         | <a href="#">eventbridge-c#-sdk</a>     | <a href="#">C#文档</a>         |
| TypeScript | <a href="#">eventbridge-ts-sdk</a>     | <a href="#">TypeScript文档</a> |

## 1.2. Java SDK

### 1.2.1. 概述

本文介绍事件总线EventBridge Java SDK中包含的内容。

#### 内容介绍

事件总线EventBridge Java SDK提供下表中丰富的文档内容，您在实际操作过程中可以参见这些内容。

| 文档                         | 说明                      |
|----------------------------|-------------------------|
| <a href="#">版本说明</a>       | 介绍Java SDK的版本更新记录。      |
| <a href="#">安装Java SDK</a> | 介绍如何安装Java SDK。         |
| <a href="#">快速入门</a>       | 介绍如何快速使用Java SDK完成常见操作。 |
| <a href="#">事件总线</a>       | 介绍如何使用Java SDK管理事件总线。   |
| <a href="#">事件规则</a>       | 介绍如何使用Java SDK管理事件规则。   |
| <a href="#">事件</a>         | 介绍如何使用Java SDK发布事件。     |
| <a href="#">事件目标</a>       | 介绍如何使用Java SDK管理事件目标。   |

## 1.2.2. 版本说明

本文介绍Java SDK的版本信息，包含下载链接、发布时间、更新说明等，以便您按需获取适用的Java SDK。

### Version 1.2.x

| 发布时间       | 下载                                                | 更新说明                                                                                                                                       |
|------------|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 2020-12-18 | <a href="#">SDK下载</a><br><a href="#">sample下载</a> | <ul style="list-style-type: none"><li>Event格式兼容CloudEvents生态标准。</li><li>签名算法采用Aliyun Cloud Service通用签名算法。</li><li>客户端代码采用自动生成方式。</li></ul> |

## 1.2.3. 安装Java SDK

本文介绍如何安装事件总线Event Bridge Java SDK。

### 环境准备

- 环境要求
  - 安装JDK。更多信息，请参见[安装JDK](#)。
  - 安装Maven。更多信息，请参见[安装Maven](#)。
- 查看语言版本

执行命令 `java -version` 查看Java语言版本。

### 安装SDK

在`pom.xml`中添加以下依赖。

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>eventbridge-client</artifactId>
  <version>1.2.6</version>
</dependency>
```

## 1.2.4. 快速入门

本文以发布事件到事件总线Event Bridge为例介绍如何快速使用Java SDK调用API。

### 前提条件

您已完成以下操作：

- [开通事件总线Event Bridge服务](#)
- [创建AccessKey](#)

- 安装Java SDK

## 发布事件

示例代码如下：

```
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;
import java.net.URI;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
public class PutEventsSample {
    private 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<>();
        cloudEventList.add(EventBuilder.builder()
            .withId("a5074581-7e74-4e4c-868f-47e7afdf****")
            .withSource(URI.create("acs.oss"))
            .withType("oss:ObjectCreated:PostObject")
            .withSubject("acs:oss:cn-hangzhou:{yourAccountId}:xls-papk/game_apk/123.jpg")
            .withTime(new Date())
            .withJsonStringData("{\"E-Mail\": \"{email}\"}")
            .withAliyunEventBus("demo-bus")
            .build());
        PutEventsResponse putEventsResponse = eventBridgeClient.putEvents(cloudEventList);
        System.out.println(new Gson().toJson(putEventsResponse));
    }
    public static void main(String[] args){
        PutEventsSample sample = new PutEventsSample();
        sample.putEventsSample();
    }
}
```

```
try {
    sample.putEventsSample();
} catch (Throwable e) {
    e.printStackTrace();
}
}
```

## 1.2.5. 事件总线

### 1.2.5.1. 事件总线概述

您可以通过Java SDK调用事件总线接口创建总线、删除总线、获取总线详情和查询事件总线。

Java SDK提供以下事件总线接口：

- [CreateEventBus](#)
- [GetEventBus](#)
- [DeleteEventBus](#)
- [ListEventBuses](#)

### 1.2.5.2. CreateEventBus

本文介绍如何使用Java SDK创建一个事件总线。

示例代码如下：

```
import com.aliyun.eventbridge.EventBridge;
import com.aliyun.eventbridge.EventBridgeClient;
import com.aliyun.eventbridge.models.Config;
import com.aliyun.eventbridge.models.CreateEventBusRequest;
import com.aliyun.eventbridge.models.CreateEventBusResponse;
public class createEventBusSample {
    private EventBridge eventBridgeClient;
    public createEventBusSample() {
        Config authConfig = new Config();
        authConfig.accessKeyId = "{accessKeyId}";
        authConfig.accessKeySecret = "{accessKeySecret}";
        authConfig.endpoint = "{endpoint}";
        eventBridgeClient = new EventBridgeClient(authConfig);
    }
    public void createEventBusSample() {
        try {
            CreateEventBusRequest createEventBusRequest = new CreateEventBusRequest();
            createEventBusRequest.setEventBusName("mybus");
            createEventBusRequest.setDescription("my first bus on eventbridge!");
            CreateEventBusResponse response = eventBridgeClient.createEventBus(createEventBusRequest);
            System.out.println("create bus success : " + response);
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
    public static void main(String[] args) {
        createEventBusSample sample = new createEventBusSample();
        try {
            sample.createEventBusSample();
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
}
```

### 1.2.5.3. GetEventBus

本文介绍如何使用Java SDK获取一个事件总线的详细信息。

示例代码如下：

```
import com.aliyun.eventbridge.EventBridge;
import com.aliyun.eventbridge.EventBridgeClient;
import com.aliyun.eventbridge.models.Config;
import com.aliyun.eventbridge.models.GetEventBusRequest;
import com.aliyun.eventbridge.models.GetEventBusResponse;
public class getEventBusSample {
    private EventBridge eventBridgeClient;
    public getEventBusSample() {
        Config authConfig = new Config();
        authConfig.accessKeyId = "{accessKeyId}";
        authConfig.accessKeySecret = "{accessKeySecret}";
        authConfig.endpoint = "{endpoint}";
        eventBridgeClient = new EventBridgeClient(authConfig);
    }
    public void getEventBusSample() {
        try {
            GetEventBusRequest describeEventBusRequest = new GetEventBusRequest();
            describeEventBusRequest.setEventBusName("mybus");
            GetEventBusResponse deScribeResponse = eventBridgeClient.getEventBus(describeEventBusRequest
);
            System.out.println("describe bus success : " + deScribeResponse);
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
    public static void main(String[] args) {
        getEventBusSample sample = new getEventBusSample();
        try {
            sample.getEventBusSample();
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
}
```

### 1.2.5.4. DeleteEventBus

本文介绍如何使用Java SDK删除一个事件总线。

示例代码如下：

```
import com.aliyun.eventbridge.EventBridge;
import com.aliyun.eventbridge.EventBridgeClient;
import com.aliyun.eventbridge.models.Config;
import com.aliyun.eventbridge.models.DeleteEventBusRequest;
public class deleteEventBusSample {
    private EventBridge eventBridgeClient;
    public deleteEventBusSample() {
        Config authConfig = new Config();
        authConfig.accessKeyId = "{accessKeyId}";
        authConfig.accessKeySecret = "{accessKeySecret}";
        authConfig.endpoint = "{endpoint}";
        eventBridgeClient = new EventBridgeClient(authConfig);
    }
    public void deleteEventBusSample() {
        try {
            DeleteEventBusRequest deleteEventBusRequest = new DeleteEventBusRequest();
            deleteEventBusRequest.setEventBusName("mybus");
            eventBridgeClient.deleteEventBus(deleteEventBusRequest);
            System.out.println("delete bus success : " + deleteEventBusRequest.getEventBusName());
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
    public static void main(String[] args) {
        deleteEventBusSample sample = new deleteEventBusSample();
        try {
            sample.deleteEventBusSample();
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
}
```

### 1.2.5.5. ListEventBuses

本文介绍如何使用Java SDK查询所有事件总线。

示例代码如下：

```
import com.aliyun.eventbridge.EventBridge;
import com.aliyun.eventbridge.EventBridgeClient;
import com.aliyun.eventbridge.models.Config;
import com.aliyun.eventbridge.models.ListEventBusesRequest;
import com.aliyun.eventbridge.models.ListEventBusesResponse;
public class listEventBusSample {
    private EventBridge eventBridgeClient;
    public listEventBusSample() {
        Config authConfig = new Config();
        authConfig.accessKeyId = "{accessKeyId}";
        authConfig.accessKeySecret = "{accessKeySecret}";
        authConfig.endpoint = "{endpoint}";
        eventBridgeClient = new EventBridgeClient(authConfig);
    }
    public void listEventBusSample() {
        try {
            ListEventBusesRequest listEventBusesRequest = new ListEventBusesRequest();
            ListEventBusesResponse response = eventBridgeClient.listEventBuses(listEventBusesRequest);
            System.out.println("list bus success : " + response);
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
    public static void main(String[] args) {
        listEventBusSample sample = new listEventBusSample();
        try {
            sample.listEventBusSample();
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
}
```

## 1.2.6. 事件规则

### 1.2.6.1. 事件规则概述

您可以通过Java SDK调用事件规则接口创建规则、禁用规则、启用规则、删除规则等。

Java SDK提供以下事件规则接口：

- [CreateRule](#)
- [Get Rule](#)
- [EnableRule](#)
- [DisableRule](#)
- [UpdateRule](#)
- [List Rules](#)
- [DeleteRule](#)

## 1.2.6.2. CreateRule

本文介绍如何使用Java SDK创建一个事件规则。

示例代码如下：

```
import java.util.Arrays;
import java.util.List;
import com.aliyun.eventbridge.EventBridge;
import com.aliyun.eventbridge.EventBridgeClient;
import com.aliyun.eventbridge.models.Config;
import com.aliyun.eventbridge.models.CreateRuleRequest;
import com.aliyun.eventbridge.models.CreateRuleResponse;
import com.aliyun.eventbridge.models.EBTargetParam;
import com.aliyun.eventbridge.models.TargetEntry;
import com.aliyun.eventbridge.util.CommonStatus;
import com.google.common.collect.Lists;
public class createEventRuleSample {
    private EventBridge eventBridgeClient;
    public createEventRuleSample() {
        Config authConfig = new Config();
        authConfig.accessKeyId = "{accessKeyId}";
        authConfig.accessKeySecret = "{accessKeySecret}";
        authConfig.endpoint = "{endpoint}";
        eventBridgeClient = new EventBridgeClient(authConfig);
    }
    public void createEventRuleSample() {
        try {
            CreateRuleRequest createEventRuleRequest = new CreateRuleRequest();
            EBTargetParam param1 = new EBTargetParam();
            param1.setResourceKey("URL");
            param1.setForm("CONSTANT");
            param1.setValue("https://oapi.dingtalk"
                + ".com/robot/send?access_token=1019d4a19e2ef6b2f*****396fc5e94814ed8460");
            EBTargetParam param2 = new EBTargetParam();
```

```

EBTargetParam param2 = new EBTargetParam();
param2.setResourceKey("SecretKey");
param2.setForm("CONSTANT");
param2.setValue("SEC121a71ff304a65b4f7c*****1f4d9f6c1ca514300d15234");
EBTargetParam param3 = new EBTargetParam();
param3.setResourceKey("Body");
param3.setForm("TEMPLATE");
param3.setValue("{\"key\": \"$.source\", \"value\": \"$.data\"}");
param3.setTemplate("{\"msgtype\": \"text\", \"text\": \" \"content\": \"Hello: ${key}\"}");
List<EBTargetParam> paramList = Lists.newArrayList(param1, param2, param3);
TargetEntry targetEntry = new TargetEntry();
targetEntry.setId("dingtalk.target");
targetEntry.setType("acs.dingtalk");
targetEntry.setParamList(paramList);
targetEntry.setEndpoint("https://oapi.dingtalk"
    + ".com/robot/send?access_token=1019d4a19e2ef6b2f*****396fc5e94814ed8460");
List<TargetEntry> targetEntryList = Arrays.asList(targetEntry);
createEventRuleRequest.setRuleName("myRule");
createEventRuleRequest.setEventBusName("mybus");
createEventRuleRequest.setFilterPattern(
    "{\"source\": [\"crmabc.newsletter\"], \"type\": [\"UserSignUp\", \"UserLogin\"]}");
createEventRuleRequest.setStatus(CommonStatus.ENABLE.getKey());
createEventRuleRequest.setTargets(targetEntryList);
CreateRuleResponse response = eventBridgeClient.createRule(createEventRuleRequest);
System.out.println("create rule success : " + response);
} catch (Throwable e) {
    e.printStackTrace();
}
}

public static void main(String[] args) {
    createEventRuleSample eventRuleSamples = new createEventRuleSample();
    try {
        eventRuleSamples.createEventRuleSample();
    } catch (Throwable e) {
        e.printStackTrace();
    }
}
}

```

### 1.2.6.3. GetRule

本文介绍如何使用Java SDK获取一个事件规则的详细信息。

示例代码如下：

```
import com.aliyun.eventbridge.EventBridge;
import com.aliyun.eventbridge.EventBridgeClient;
import com.aliyun.eventbridge.models.Config;
import com.aliyun.eventbridge.models.GetRuleRequest;
import com.aliyun.eventbridge.models.GetRuleResponse;
public class getEventRuleSample {
    private EventBridge eventBridgeClient;
    public getEventRuleSample() {
        Config authConfig = new Config();
        authConfig.accessKeyId = "{accessKeyId}";
        authConfig.accessKeySecret = "{accessKeySecret}";
        authConfig.endpoint = "{endpoint}";
        eventBridgeClient = new EventBridgeClient(authConfig);
    }
    public void getEventRuleSample() {
        try {
            GetRuleRequest describeEventRuleRequest = new GetRuleRequest();
            describeEventRuleRequest.setRuleName("myRule");
            describeEventRuleRequest.setEventBusName("mybus");
            GetRuleResponse response = eventBridgeClient.getRule(describeEventRuleRequest);
            System.out.println("describe rule success : " + response);
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
    public static void main(String[] args) {
        getEventRuleSample eventRuleSamples = new getEventRuleSample();
        try {
            eventRuleSamples.getEventRuleSample();
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
}
```

## 1.2.6.4. EnableRule

本文介绍如何使用Java SDK启用一个事件规则。

示例代码如下：

```
import com.aliyun.eventbridge.EventBridge;
import com.aliyun.eventbridge.EventBridgeClient;
import com.aliyun.eventbridge.models.Config;
import com.aliyun.eventbridge.models.EnableRuleRequest;
public class enableEventRuleSample {
    private EventBridge eventBridgeClient;
    public enableEventRuleSample() {
        Config authConfig = new Config();
        authConfig.accessKeyId = "{accessKeyId}";
        authConfig.accessKeySecret = "{accessKeySecret}";
        authConfig.endpoint = "{endpoint}";
        eventBridgeClient = new EventBridgeClient(authConfig);
    }
    public void enableEventRuleSample() {
        try {
            EnableRuleRequest enableEventRuleRequest = new EnableRuleRequest();
            enableEventRuleRequest.setRuleName("myRule");
            enableEventRuleRequest.setEventBusName("mybus");
            eventBridgeClient.enableRule(enableEventRuleRequest);
            System.out.println("enable rule success : " + enableEventRuleRequest.getEventBusName() + "/"
                + enableEventRuleRequest.getRuleName());
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
    public static void main(String[] args) {
        enableEventRuleSample eventRuleSamples = new enableEventRuleSample();
        try {
            eventRuleSamples.enableEventRuleSample();
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
}
```

## 1.2.6.5. DisableRule

本文介绍如何使用Java SDK禁用一个事件规则。

示例代码如下：

```
import com.aliyun.eventbridge.EventBridge;
import com.aliyun.eventbridge.EventBridgeClient;
import com.aliyun.eventbridge.models.Config;
import com.aliyun.eventbridge.models.DisableRuleRequest;

public class disableEventRuleSample {
    private EventBridge eventBridgeClient;

    public disableEventRuleSample() {
        Config authConfig = new Config();
        authConfig.accessKeyId = "{accessKeyId}";
        authConfig.accessKeySecret = "{accessKeySecret}";
        authConfig.endpoint = "{endpoint}";
        eventBridgeClient = new EventBridgeClient(authConfig);
    }

    public void disableEventRuleSample() {
        try {
            DisableRuleRequest disableEventRuleRequest = new DisableRuleRequest();
            disableEventRuleRequest.setRuleName("myRule");
            disableEventRuleRequest.setEventBusName("mybus");
            eventBridgeClient.disableRule(disableEventRuleRequest);
            System.out.println("disable rule success : " + disableEventRuleRequest.getEventBusName() + "/"
                + disableEventRuleRequest.getRuleName());
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        disableEventRuleSample eventRuleSamples = new disableEventRuleSample();
        try {
            eventRuleSamples.disableEventRuleSample();
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
}
```

## 1.2.6.6. UpdateRule

本文介绍如何使用Java SDK更新一个事件规则的配置。

示例代码如下：

```
import com.aliyun.eventbridge.EventBridge;
import com.aliyun.eventbridge.EventBridgeClient;
import com.aliyun.eventbridge.models.Config;
import com.aliyun.eventbridge.models.UpdateRuleRequest;
public class updateEventRuleSample {
    private EventBridge eventBridgeClient;
    public updateEventRuleSample() {
        Config authConfig = new Config();
        authConfig.accessKeyId = "{accessKeyId}";
        authConfig.accessKeySecret = "{accessKeySecret}";
        authConfig.endpoint = "{endpoint}";
        eventBridgeClient = new EventBridgeClient(authConfig);
    }
    public void updateEventRuleSample() {
        try {
            UpdateRuleRequest updateEventRuleRequest = new UpdateRuleRequest();
            updateEventRuleRequest.setRuleName("myRule");
            updateEventRuleRequest.setEventBusName("mybus");
            updateEventRuleRequest.setFilterPattern("{\"source\":[\"crmabc.newsletter\"],\"type\":[\"UserSignUp\"]}");
            eventBridgeClient.updateRule(updateEventRuleRequest);
            System.out.println("update rule success! ");
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
    public static void main(String[] args) {
        updateEventRuleSample eventRuleSamples = new updateEventRuleSample();
        try {
            eventRuleSamples.updateEventRuleSample();
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
}
```

## 1.2.6.7. ListRules

本文介绍如何使用Java SDK查询一个事件总线的所有事件规则。

示例代码如下：

```
import com.aliyun.eventbridge.EventBridge;
import com.aliyun.eventbridge.EventBridgeClient;
import com.aliyun.eventbridge.models.Config;
import com.aliyun.eventbridge.models.ListRulesRequest;
import com.aliyun.eventbridge.models.ListRulesResponse;
public class listEventRuleSample {
    private EventBridge eventBridgeClient;
    public listEventRuleSample() {
        Config authConfig = new Config();
        authConfig.accessKeyId = "{accessKeyId}";
        authConfig.accessKeySecret = "{accessKeySecret}";
        authConfig.endpoint = "{endpoint}";
        eventBridgeClient = new EventBridgeClient(authConfig);
    }
    public void listEventRuleSample() {
        try {
            ListRulesRequest listEventRulesRequest = new ListRulesRequest();
            listEventRulesRequest.setEventBusName("mybus");
            ListRulesResponse response = eventBridgeClient.listRules(listEventRulesRequest);
            System.out.println("list rule success : " + response);
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
    public static void main(String[] args) {
        listEventRuleSample eventRuleSamples = new listEventRuleSample();
        try {
            eventRuleSamples.listEventRuleSample();
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
}
```

## 1.2.6.8. DeleteRule

本文介绍如何使用Java SDK删除一个事件规则。

示例代码如下：

```
import com.aliyun.eventbridge.EventBridge;
import com.aliyun.eventbridge.EventBridgeClient;
import com.aliyun.eventbridge.models.Config;
import com.aliyun.eventbridge.models.DeleteRuleRequest;

public class deleteEventRuleSample {
    private EventBridge eventBridgeClient;

    public deleteEventRuleSample() {
        Config authConfig = new Config();
        authConfig.accessKeyId = "{accessKeyId}";
        authConfig.accessKeySecret = "{accessKeySecret}";
        authConfig.endpoint = "{endpoint}";
        eventBridgeClient = new EventBridgeClient(authConfig);
    }

    public void deleteEventRuleSample() {
        try {
            DeleteRuleRequest deleteEventRuleRequest = new DeleteRuleRequest();
            deleteEventRuleRequest.setRuleName("myRule");
            deleteEventRuleRequest.setEventBusName("mybus");
            eventBridgeClient.deleteRule(deleteEventRuleRequest);
            System.out.println("delete rule success : " + deleteEventRuleRequest.getEventBusName() + "/"
                + deleteEventRuleRequest.getRuleName());
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        deleteEventRuleSample eventRuleSamples = new deleteEventRuleSample();
        try {
            eventRuleSamples.deleteEventRuleSample();
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
}
```

## 1.2.7. 事件

### 1.2.7.1. 事件概述

您可以通过Java SDK调用事件接口发布事件。

Java SDK提供以下事件接口：

- [PutEvents](#)

### 1.2.7.2. PutEvents

本文介绍如何使用Java SDK发布一个或多个事件。

示例代码如下：

```
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;
import java.net.URI;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

public class PutEventsSample {
    private 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<>();
        cloudEventList.add(EventBuilder.builder()
            .withId("a5074581-7e74-4e4c-868f-47e7afdf****")
            .withSource(URI.create("acs.oss"))
            .withType("oss:ObjectCreated:PostObject")
            .withSubject("acs:oss:cn-hangzhou:{yourAccountId}:xls-papk/game_apk/123.jpg")
            .withTime(new Date())
        );
    }
}
```

```
        .withJsonStringData("{\"E-Mail\": \"{email}\"}")
        .withAliyunEventBus("demo-bus")
        .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();
    }
}
```

## 1.2.8. 事件目标

### 1.2.8.1. 事件目标概述

您可以通过Java SDK调用事件目标接口创建事件目标、删除事件目标和查询事件目标。

Java SDK提供以下事件目标接口：

- [CreateTargets](#)
- [ListTargets](#)
- [DeleteTargets](#)

### 1.2.8.2. CreateTargets

本文介绍如何使用Java SDK为一个事件规则创建一个或多个投递目标。

示例代码如下：

```
import java.util.Arrays;
import java.util.List;
import com.aliyun.eventbridge.EventBridge;
import com.aliyun.eventbridge.EventBridgeClient;
import com.aliyun.eventbridge.models.Config;
import com.aliyun.eventbridge.models.CreateTargetsRequest;
import com.aliyun.eventbridge.models.CreateTargetsResponse;
import com.aliyun.eventbridge.models.EBTargetParam;
import com.aliyun.eventbridge.models.TargetEntry;
```

```
import com.google.common.collect.Lists;

public class createTargetsSample {
    private EventBridge eventBridgeClient;

    public createTargetsSample() {
        Config authConfig = new Config();
        authConfig.accessKeyId = "{accessKeyId}";
        authConfig.accessKeySecret = "{accessKeySecret}";
        authConfig.endpoint = "{endpoint}";
        eventBridgeClient = new EventBridgeClient(authConfig);
    }

    public void createTargetsSample() {
        try {
            TargetEntry targetEntry = new TargetEntry();
            targetEntry.setId("dingtalk.target");
            targetEntry.setType("acs.dingtalk");
            targetEntry.setEndpoint("https://oapi.dingtalk"
                + ".com/robot/send?access_token=1019d4a19e2ef6b2f*****396fc5e94814ed8460");
            EBTargetParam param1 = new EBTargetParam();
            param1.setResourceKey("URL");
            param1.setForm("CONSTANT");
            param1.setValue("https://oapi.dingtalk"
                + ".com/robot/send?access_token=1019d4a19e2ef6b2f*****396fc5e94814ed8460");
            EBTargetParam param2 = new EBTargetParam();
            param2.setResourceKey("SecretKey");
            param2.setForm("CONSTANT");
            param2.setValue("SEC121a71ff304a65b4f7c*****1f4d9f6c1ca514300d15234");
            EBTargetParam param3 = new EBTargetParam();
            param3.setResourceKey("Body");
            param3.setForm("TEMPLATE");
            param3.setValue("{\"n\" + \"  \\\"key\\\":\\\"$.source\\\",\\n\" + \"  \\\"value\\\":\\\"$.data\\\"\\n\" + \"}\"");
            param3.setTemplate("{\"msgtype\\\": \"text\\\",\\\"text\\\": \" + \"{\\\"content\\\": \\\"Hello: ${key}\\\"}\"}");
            List<EBTargetParam> paramList = Lists.newArrayList(param1, param2, param3);
            targetEntry.setParamList(paramList);
            CreateTargetsRequest createTargetsRequest = new CreateTargetsRequest();
            createTargetsRequest.setRuleName("myRule");
            createTargetsRequest.setEventBusName("mybus");
            createTargetsRequest.setTargets(Arrays.asList(targetEntry));
            CreateTargetsResponse response = eventBridgeClient.createTargets(createTargetsRequest);
            System.out.println("add targets success : " + response);
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
}
```

```
    }  
}  
public static void main(String[] args) {  
    createTargetsSample eventRuleSamples = new createTargetsSample();  
    try {  
        eventRuleSamples.createTargetsSample();  
    } catch (Throwable e) {  
        e.printStackTrace();  
    }  
}
```

### 1.2.8.3. ListTargets

本文介绍如何使用Java SDK查询一个事件规则的所有投递目标。

示例代码如下：

```
import com.aliyun.eventbridge.EventBridge;
import com.aliyun.eventbridge.EventBridgeClient;
import com.aliyun.eventbridge.models.Config;
import com.aliyun.eventbridge.models.ListTargetsRequest;
import com.aliyun.eventbridge.models.ListTargetsResponse;
public class listTargetsSample {
    private EventBridge eventBridgeClient;
    public listTargetsSample() {
        Config authConfig = new Config();
        authConfig.accessKeyId = "{accessKeyId}";
        authConfig.accessKeySecret = "{accessKeySecret}";
        authConfig.endpoint = "{endpoint}";
        eventBridgeClient = new EventBridgeClient(authConfig);
    }
    public void listTargetsSample() {
        try {
            ListTargetsRequest listTargetsRequest = new ListTargetsRequest();
            listTargetsRequest.setRuleName("myRule");
            listTargetsRequest.setEventBusName("mybus");
            ListTargetsResponse response = eventBridgeClient.listTargets(listTargetsRequest);
            System.out.println("list targets success : " + response);
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
    public static void main(String[] args) {
        listTargetsSample eventRuleSamples = new listTargetsSample();
        try {
            eventRuleSamples.listTargetsSample();
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
}
```

## 1.2.8.4. DeleteTargets

本文介绍如何使用Java SDK删除一个事件规则的一个或多个投递目标。

示例代码如下：

```
import java.util.Arrays;
import com.aliyun.eventbridge.EventBridge;
import com.aliyun.eventbridge.EventBridgeClient;
import com.aliyun.eventbridge.models.Config;
import com.aliyun.eventbridge.models.DeleteTargetsRequest;
import com.aliyun.eventbridge.models.DeleteTargetsResponse;
public class deleteTargetsSample {
    private EventBridge eventBridgeClient;
    public deleteTargetsSample() {
        Config authConfig = new Config();
        authConfig.accessKeyId = "{accessKeyId}";
        authConfig.accessKeySecret = "{accessKeySecret}";
        authConfig.endpoint = "{endpoint}";
        eventBridgeClient = new EventBridgeClient(authConfig);
    }
    public void deleteTargetsSample() {
        try {
            DeleteTargetsRequest deleteTargetsRequest = new DeleteTargetsRequest();
            deleteTargetsRequest.setRuleName("myRule");
            deleteTargetsRequest.setEventBusName("mybus");
            deleteTargetsRequest.setTargetIds(Arrays.asList("httpTarget2"));
            DeleteTargetsResponse response = eventBridgeClient.deleteTargets(deleteTargetsRequest);
            System.out.println("remove targets success : " + response);
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
    public static void main(String[] args) {
        deleteTargetsSample eventRuleSamples = new deleteTargetsSample();
        try {
            eventRuleSamples.deleteTargetsSample();
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
}
```

## 1.3. Go SDK

## 1.3.1. 概述

本文介绍事件总线EventBridge Go SDK中包含的内容。

### 内容介绍

事件总线EventBridge Go SDK提供下表中丰富的文档内容，您在实际操作过程中可以参见这些内容。

| 文档                       | 说明                    |
|--------------------------|-----------------------|
| <a href="#">版本说明</a>     | 介绍Go SDK的版本更新记录。      |
| <a href="#">安装Go SDK</a> | 介绍如何安装Go SDK。         |
| <a href="#">快速入门</a>     | 介绍如何快速使用Go SDK完成常见操作。 |
| <a href="#">事件总线</a>     | 介绍如何使用Go SDK管理事件总线。   |
| <a href="#">事件规则</a>     | 介绍如何使用Go SDK管理事件规则。   |
| <a href="#">事件</a>       | 介绍如何使用Go SDK发布事件。     |
| <a href="#">事件目标</a>     | 介绍如何使用Go SDK管理事件目标。   |

## 1.3.2. 版本说明

本文介绍Go SDK的版本信息，包含下载链接、发布时间、更新说明等，以便您按需获取适用的Go SDK。

### Version 1.2.x

| 发布时间       | 下载                                                | 更新说明                                                                                                                                       |
|------------|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 2020-10-10 | <a href="#">SDK下载</a><br><a href="#">sample下载</a> | <ul style="list-style-type: none"><li>Event格式兼容CloudEvents生态标准。</li><li>签名算法采用Aliyun Cloud Service通用签名算法。</li><li>客户端代码采用自动生成方式。</li></ul> |

## 1.3.3. 安装Go SDK

本文介绍如何安装事件总线EventBridge Go SDK。

### 环境准备

- 环境要求  
安装Golang1.12.0以上的版本。更多信息，请参见[安装Golang](#)。
- 查看语言版本  
执行命令 `go version` 查看Go语言版本。

## 安装SDK

1. 执行以下命令启用Go mod。

```
go env -w GO111MODULE=on
```

2. 执行以下命令设置Go mod代理。

```
go env -w GOPROXY=https://goproxy.cn,direct
```

3. 执行以下命令进行初始化，生成`go.mod`文件。

```
go mod init
```

4. 执行以下命令安装Go SDK。

```
go get github.com/alibabacloud-go/eventbridge-sdk
```

5. 执行以下命令安装打印端口。

```
go get github.com/alibabacloud/tea-console
```

### 1.3.4. 快速入门

本文以发布事件到事件总线Event Bridge为例介绍如何快速使用Go SDK调用API。

#### 前提条件

- [开通事件总线Event Bridge服务](#)
- [创建AccessKey](#)
- [安装Go SDK](#)

#### 发布事件

示例代码如下：

```
package main
import (
    eventbridge "github.com/alibabacloud-go/eventbridge-sdk/eventbridge"
    console "github.com/alibabacloud-go/tea-console/client"
    util "github.com/alibabacloud-go/tea-utils/service"
    "github.com/alibabacloud-go/tea/tea"
)
/**
 * Create client初始化公共请求参数。
 */
func CreateClient() (_result *eventbridge.Client, _err error) {
    config := &eventbridge.Config{}
    // 您的AccessKey ID。
    config.AccessKeyId = tea.String("<accessKeyId>")
```

```

    config.AccessKeyId = tea.String("<accessKeyId>")
    // 您的AccessKey Secret。
    config.AccessKeySecret = tea.String("<accessKeySecret>")
    // 您的接入点。
    config.Endpoint = tea.String("<endpoint>")
    _result = &eventbridge.Client{}
    _result, _err = eventbridge.NewClient(config)
    return _result, _err
}
/**
 * PutEvents
 */
func PutEvents(client *eventbridge.Client) (_err error) {
    event := &eventbridge.CloudEvent{}
    event.Datacontenttype = tea.String("application/json")
    event.Data = util.ToBytes(tea.String("test"))
    event.Id = tea.String("a5074581-7e74-4e4c-868f-47e7afdf8445")
    event.Source = tea.String("acs.oss")
    event.Specversion = tea.String("1.0")
    event.Type = tea.String("oss:ObjectCreated:PostObject")
    event.Time = tea.String("2020-08-24T13:54:05.965Asia/Shanghai")
    event.Subject = tea.String("1.0")
    event.Type = tea.String("acs:oss:cn-hangzhou:1234567:xls-papk/game_apk/123.jpg")
    event.Extensions = map[string]interface{}{
        "aliyuneventbusname": "demo-bus",
    }
    tryErr := func() (_e error) {
        defer func() {
            if r := tea.Recover(recover()); r != nil {
                _e = r
            }
        }()
        resp, _err := client.PutEvents([]*eventbridge.CloudEvent{event})
        if _err != nil {
            return _err
        }
        console.Log(tea.String("-----Publish event to the aliyun EventBus-----"))
        console.Log(util.ToJSONString(tea.ToMap(resp)))
        return nil
    }()
    _err = tryErr
}

```

```
if tryErr != nil {
    var error = &tea.SDKError{}
    if _t, ok := tryErr.(*tea.SDKError); ok {
        error = _t
    } else {
        error.SetErrMsg(tryErr.Error())
    }
    console.Log(error.Message)
}
return _err
}

func main() {
    client, _err := CreateClient()
    if _err != nil {
        panic(_err)
    }
    _err = PutEvents(client)
    if _err != nil {
        panic(_err)
    }
}
```

## 1.3.5. 事件总线

### 1.3.5.1. 事件总线概述

您可以通过Go SDK调用事件总线接口创建总线、删除总线、获取总线详情和查询事件总线。

Go SDK提供以下事件总线接口：

- [CreateEventBus](#)
- [GetEventBus](#)
- [DeleteEventBus](#)
- [ListEventBuses](#)

### 1.3.5.2. CreateEventBus

本文介绍如何使用Go SDK创建一个事件总线。

示例代码如下：

```
package main

import (
    eventbridge "github.com/alibabacloud-go/eventbridge-sdk/eventbridge"
```

```

    console "github.com/alibabacloud-go/tea-console/client"
    util "github.com/alibabacloud-go/tea-utils/service"
    "github.com/alibabacloud-go/tea/tea"
)
/**
 * Create client初始化公共请求参数。
 */
func CreateClient() (_result *eventbridge.Client, _err error) {
    config := &eventbridge.Config{}
    // 您的AccessKey ID。
    config.AccessKeyId = tea.String("<accessKeyId>")
    // 您的AccessKey Secret。
    config.AccessKeySecret = tea.String("<accessKeySecret>")
    // 您的接入点。
    config.Endpoint = tea.String("<endpoint>")
    _result = &eventbridge.Client{}
    _result, _err = eventbridge.NewClient(config)
    return _result, _err
}

func CreateEventBusSample(client *eventbridge.Client) (_err error) {
    tryErr := func() (_e error) {
        defer func() {
            if r := tea.Recover(recover()); r != nil {
                _e = r
            }
        }()
        createEventBusRequest := &eventbridge.CreateEventBusRequest{}
        createEventBusRequest.EventBusName = tea.String("demo-bus")
        response, _err := client.CreateEventBus(createEventBusRequest)
        if _err != nil {
            return _err
        }
        console.Log(tea.String("-----Create bus success -----"))
        console.Log(util.ToJSONString(tea.ToMap(response)))
        return nil
    }()
    _err = tryErr
    if tryErr != nil {
        var error = &tea.SDKError{}
        if _t, ok := tryErr.(*tea.SDKError); ok {
            error = *

```

```
        error = _err
    } else {
        error.SetErrMsg(tryErr.Error())
    }
    console.Log(error.Message)
}
return _err
}
func main() {
    client, _err := CreateClient()
    if _err != nil {
        panic(_err)
    }
    _err = CreateEventBusSample(client)
    if _err != nil {
        panic(_err)
    }
}
```

### 1.3.5.3. GetEventBus

本文介绍如何使用Go SDK获取一个事件总线的详细信息。

示例代码如下：

```
package main
import (
    eventbridge "github.com/alibabacloud-go/eventbridge-sdk/eventbridge"
    console "github.com/alibabacloud-go/tea-console/client"
    util "github.com/alibabacloud-go/tea-utils/service"
    "github.com/alibabacloud-go/tea/tea"
)
/**
 * Create client初始化公共请求参数。
 */
func CreateClient() (_result *eventbridge.Client, _err error) {
    config := &eventbridge.Config{}
    // 您的AccessKey ID。
    config.AccessKeyId = tea.String("<accessKeyId>")
    // 您的AccessKey Secret。
    config.AccessKeySecret = tea.String("<accessKeySecret>")
    // 您的接入点。
```

```
config.Endpoint = tea.String("<endpoint>")
_result = &eventbridge.Client{}
_result, _err = eventbridge.NewClient(config)
return _result, _err
}

func GetEventBusSample(client *eventbridge.Client) (_err error) {
    tryErr := func() (_e error) {
        defer func() {
            if r := tea.Recover(recover()); r != nil {
                _e = r
            }
        }()
        describeEventBusRequest := &eventbridge.GetEventBusRequest{}
        describeEventBusRequest.EventBusName = tea.String("demo-bus")
        deScribeResponse, _err := client.GetEventBus(describeEventBusRequest)
        if _err != nil {
            return _err
        }
        console.Log(tea.String("-----describe bus success -----"))
        console.Log(util.ToJSONString(tea.ToMap(deScribeResponse)))
        return nil
    }()
    _err = tryErr
    if tryErr != nil {
        var error = &tea.SDKError{}
        if _t, ok := tryErr.(*tea.SDKError); ok {
            error = _t
        } else {
            error.SetErrMsg(tryErr.Error())
        }
        console.Log(error.Message)
    }
    return _err
}

func main() {
    client, _err := CreateClient()
    if _err != nil {
        panic(_err)
    }
    _err = GetEventBusSample(client)
    if _err != nil {
```

```
    panic(_err)
  }
}
```

### 1.3.5.4. DeleteEventBus

本文介绍如何使用Go SDK删除一个事件总线。

示例代码如下：

```
package main
import (
    eventbridge "github.com/alibabacloud-go/eventbridge-sdk/eventbridge"
    console "github.com/alibabacloud-go/tea-console/client"
    "github.com/alibabacloud-go/tea/tea"
)
/**
 * Create client初始化公共请求参数。
 */
func CreateClient() (_result *eventbridge.Client, _err error) {
    config := &eventbridge.Config{}
    // 您的AccessKey ID。
    config.AccessKeyId = tea.String("<accessKeyId>")
    // 您的AccessKey Secret。
    config.AccessKeySecret = tea.String("<accessKeySecret>")
    // 您的接入点。
    config.Endpoint = tea.String("<endpoint>")
    _result = &eventbridge.Client{}
    _result, _err = eventbridge.NewClient(config)
    return _result, _err
}
func DeleteEventBusSample(client *eventbridge.Client) (_err error) {
    tryErr := func() (_e error) {
        defer func() {
            if r := tea.Recover(recover()); r != nil {
                _e = r
            }
        }()
        //delete
        deleteEventBusRequest := &eventbridge.DeleteEventBusRequest{}
        deleteEventBusRequest.EventBusName = tea.String("demo-bus22")
    }
}
```

```
_, _err = client.DeleteEventBus(deleteEventBusRequest)
if _err != nil {
    return _err
}
console.Log(tea.String("-----Delete bus success -----"))
return nil
}()
_err = tryErr
if tryErr != nil {
    var error = &tea.SDKError{}
    if _t, ok := tryErr.(*tea.SDKError); ok {
        error = _t
    } else {
        error.SetErrMsg(tryErr.Error())
    }
    console.Log(error.Message)
}
return _err
}
func main() {
    client, _err := CreateClient()
    if _err != nil {
        panic(_err)
    }
    _err = DeleteEventBusSample(client)
    if _err != nil {
        panic(_err)
    }
}
```

### 1.3.5.5. ListEventBuses

本文介绍如何使用Go SDK查询所有事件总线。

示例代码如下：

```
package main
import (
    "fmt"
    eventbridge "github.com/alibabacloud-go/eventbridge-sdk/eventbridge"
    console "github.com/alibabacloud-go/tea-console/client"
    "github.com/alibabacloud-go/tea/tea"
)
```

```

)
/**
 * Create client初始化公共请求参数。
 */
func CreateClient() (_result *eventbridge.Client, _err error) {
    config := &eventbridge.Config{}
    // 您的AccessKey ID。
    config.AccessKeyId = tea.String("<accessKeyId>")
    // 您的AccessKey Secret。
    config.AccessKeySecret = tea.String("<accessKeySecret>")
    // 您的接入点。
    config.Endpoint = tea.String("<endpoint>")
    _result = &eventbridge.Client{}
    _result, _err = eventbridge.NewClient(config)
    return _result, _err
}

func ListEventRuleSample(client *eventbridge.Client) (_err error) {
    tryErr := func() (_e error) {
        defer func() {
            if r := tea.Recover(recover()); r != nil {
                _e = r
            }
        }()
        listEventRulesRequest := &eventbridge.ListRulesRequest{}
        listEventRulesRequest.EventBusName = tea.String("demo-bus")
        resp, _err := client.ListRules(listEventRulesRequest)
        if _err != nil {
            return _err
        }
        fmt.Println(resp)
        console.Log(tea.String("-----list rule success-----"))
        return nil
    }()
    _err = tryErr
    if tryErr != nil {
        var error = &tea.SDKError{}
        if _t, ok := tryErr.(*tea.SDKError); ok {
            error = _t
        } else {
            error.SetErrMsg(tryErr.Error())
        }
    }
}

```

```
}
    console.Log(error.Message)
}
return _err
}
func main() {
    client, _err := CreateClient()
    if _err != nil {
        panic(_err)
    }
    _err = ListEventRuleSample(client)
    if _err != nil {
        panic(_err)
    }
}
```

## 1.3.6. 事件规则

### 1.3.6.1. 事件规则概述

您可以通过Go SDK调用事件规则接口创建规则、禁用规则、启用规则、删除规则等。

Go SDK提供以下事件规则接口：

- [CreateRule](#)
- [GetRule](#)
- [EnableRule](#)
- [DisableRule](#)
- [UpdateRule](#)
- [ListRules](#)
- [DeleteRule](#)

### 1.3.6.2. CreateRule

本文介绍如何使用Go SDK创建一个事件规则。

示例代码如下：

```
package main
import (
    eventbridge "github.com/alibabacloud-go/eventbridge-sdk/eventbridge"
    console "github.com/alibabacloud-go/tea-console/client"
    util "github.com/alibabacloud-go/tea-utils/service"
    "github.com/alibabacloud-go/tea/tea"
```

```

)
/**
 * Create client初始化公共请求参数。
 */
func CreateClient() (_result *eventbridge.Client, _err error) {
    config := &eventbridge.Config{}
    // 您的AccessKey ID。
    config.AccessKeyId = tea.String("<accessKeyId>")
    // 您的AccessKey Secret。
    config.AccessKeySecret = tea.String("<accessKeySecret>")
    // 您的接入点。
    config.Endpoint = tea.String("<endpoint>")
    _result = &eventbridge.Client{}
    _result, _err = eventbridge.NewClient(config)
    return _result, _err
}

func CreateEventRuleSample(client *eventbridge.Client) (_err error) {
    tryErr := func() (_e error) {
        defer func() {
            if r := tea.Recover(recover()); r != nil {
                _e = r
            }
        }()
        createEventRuleRequest := &eventbridge.CreateRuleRequest{}
        targetEntry := &eventbridge.TargetEntry{}
        targetEntry.Id = tea.String("1234")
        targetEntry.Endpoint = tea.String("http://www.baidu.com")
        targetEntryList := []*eventbridge.TargetEntry{targetEntry}
        //targetEntryList[0] = targetEntry;
        createEventRuleRequest.RuleName = tea.String("myRule")
        createEventRuleRequest.EventBusName = tea.String("demo-bus")
        createEventRuleRequest.FilterPattern = tea.String("{\"source\":\"acs.oss\",\"type\":\"oss:ObjectCreated:UploadPart\"}")
        createEventRuleRequest.Status = tea.String("enable")
        createEventRuleRequest.Targets = targetEntryList
        response, _err := client.CreateRule(createEventRuleRequest)
        if _err != nil {
            return _err
        }
        console.Log(tea.String("-----create rule success-----"))
        console.Log(util.ToJSONString(tea.ToMap(response)))
    }
}

```

```
    return nil
}()
_err = tryErr
if tryErr != nil {
    var error = &tea.SDKError{}
    if _t, ok := tryErr.(*tea.SDKError); ok {
        error = _t
    } else {
        error.SetErrMsg(tryErr.Error())
    }
    console.Log(error.Message)
}
return _err
}
func main() {
    client, _err := CreateClient()
    if _err != nil {
        panic(_err)
    }
    _err = CreateEventRuleSample(client)
    if _err != nil {
        panic(_err)
    }
}
```

### 1.3.6.3. GetRule

本文介绍如何使用Go SDK获取一个事件规则的详细信息。

示例代码如下：

```
package main
import (
    eventbridge "github.com/alibabacloud-go/eventbridge-sdk/eventbridge"
    console "github.com/alibabacloud-go/tea-console/client"
    "github.com/alibabacloud-go/tea/tea"
)
/**
 * Create client初始化公共请求参数。
 */
func CreateClient() (_result *eventbridge.Client, _err error) {
```

```

config := &eventbridge.Config{}
// 您的AccessKey ID。
config.AccessKeyId = tea.String("<accessKeyId>")
// 您的AccessKey Secret。
config.AccessKeySecret = tea.String("<accessKeySecret>")
// 您的接入点。
config.Endpoint = tea.String("<endpoint>")
_result = &eventbridge.Client{}
_result, _err = eventbridge.NewClient(config)
return _result, _err
}

func GetEventRuleSample(client *eventbridge.Client) (_err error) {
    tryErr := func() (_e error) {
        defer func() {
            if r := tea.Recover(recover()); r != nil {
                _e = r
            }
        }()
        getEventRuleRequest := &eventbridge.GetRuleRequest{}
        getEventRuleRequest.RuleName = tea.String("myRule")
        getEventRuleRequest.EventBusName = tea.String("demo-bus")
        _, _err = client.GetRule(getEventRuleRequest)
        if _err != nil {
            return _err
        }
        console.Log(tea.String("-----get rule success-----"))
        return nil
    }()
    _err = tryErr
    if tryErr != nil {
        var error = &tea.SDKError{}
        if _t, ok := tryErr.(*tea.SDKError); ok {
            error = _t
        } else {
            error.SetErrMsg(tryErr.Error())
        }
        console.Log(error.Message)
    }
    return _err
}

func main() {

```

```
client, _err := CreateClient()
if _err != nil {
    panic(_err)
}
_err = GetEventRuleSample(client)
if _err != nil {
    panic(_err)
}
}
```

### 1.3.6.4. EnableRule

本文介绍如何使用Go SDK启用一个事件规则。

示例代码如下：

```
package main
import (
    eventbridge "github.com/alibabacloud-go/eventbridge-sdk/eventbridge"
    console "github.com/alibabacloud-go/tea-console/client"
    "github.com/alibabacloud-go/tea/tea"
)
/**
 * Create client初始化公共请求参数。
 */
func CreateClient() (_result *eventbridge.Client, _err error) {
    config := &eventbridge.Config{}
    // 您的AccessKey ID。
    config.AccessKeyId = tea.String("<accessKeyId>")
    // 您的AccessKey Secret。
    config.AccessKeySecret = tea.String("<accessKeySecret>")
    // 您的接入点。
    config.Endpoint = tea.String("<endpoint>")
    _result = &eventbridge.Client{}
    _result, _err = eventbridge.NewClient(config)
    return _result, _err
}
func EnableEventRuleSample(client *eventbridge.Client) (_err error) {
    tryErr := func() (_e error) {
        defer func() {
            if r := tea.Recover(recover()); r != nil {
                e = r
            }
        }()
    }
}
```

```

    }
}()
enableEventRuleRequest := &eventbridge.EnableRuleRequest{}
enableEventRuleRequest.RuleName = tea.String("myRule")
enableEventRuleRequest.EventBusName = tea.String("demo-bus")
_, _err = client.EnableRule(enableEventRuleRequest)
if _err != nil {
    return _err
}
console.Log(tea.String("-----enable rule success-----"))
return nil
}()
_err = tryErr
if tryErr != nil {
    var error = &tea.SDKError{}
    if _t, ok := tryErr.(*tea.SDKError); ok {
        error = _t
    } else {
        error.SetErrMsg(tryErr.Error())
    }
    console.Log(error.Message)
}
return _err
}
func main() {
    client, _err := CreateClient()
    if _err != nil {
        panic(_err)
    }
    _err = EnableEventRuleSample(client)
    if _err != nil {
        panic(_err)
    }
}
}

```

### 1.3.6.5. DisableRule

本文介绍如何使用Go SDK禁用一个事件规则。

示例代码如下：

```
package main
import (
    eventbridge "github.com/alibabacloud-go/eventbridge-sdk/eventbridge"
    console "github.com/alibabacloud-go/tea-console/client"
    "github.com/alibabacloud-go/tea/tea"
)
/**
 * Create client初始化公共请求参数。
 */
func CreateClient() (_result *eventbridge.Client, _err error) {
    config := &eventbridge.Config{}
    // 您的AccessKey ID。
    config.AccessKeyId = tea.String("<accessKeyId>")
    // 您的AccessKey Secret。
    config.AccessKeySecret = tea.String("<accessKeySecret>")
    // 您的接入点。
    config.Endpoint = tea.String("<endpoint>")
    _result = &eventbridge.Client{}
    _result, _err = eventbridge.NewClient(config)
    return _result, _err
}

func DisableEventRuleSample(client *eventbridge.Client) (_err error) {
    tryErr := func() (_e error) {
        defer func() {
            if r := tea.Recover(recover()); r != nil {
                _e = r
            }
        }()
        disableEventRuleRequest := &eventbridge.DisableRuleRequest{}
        disableEventRuleRequest.RuleName = tea.String("myRule")
        disableEventRuleRequest.EventBusName = tea.String("demo-bus")
        _, _err = client.DisableRule(disableEventRuleRequest)
        if _err != nil {
            return _err
        }
        console.Log(tea.String("-----disable rule success-----"))
        return nil
    }()
    _err = tryErr
    if tryErr != nil {
        var error = &tea.SDKError{}
    }
}
```

```

    if _t, ok := tryErr.(*tea.SDKError); ok {
        error = _t
    } else {
        error.SetErrMsg(tryErr.Error())
    }
    console.Log(error.Message)
}
return _err
}
func main() {
    client, _err := CreateClient()
    if _err != nil {
        panic(_err)
    }
    _err = DisableEventRuleSample(client)
    if _err != nil {
        panic(_err)
    }
}

```

### 1.3.6.6. UpdateRule

本文介绍如何使用Go SDK更新一个事件规则的配置。

示例代码如下：

```

package main
import (
    eventbridge "github.com/alibabacloud-go/eventbridge-sdk/eventbridge"
    console "github.com/alibabacloud-go/tea-console/client"
    "github.com/alibabacloud-go/tea/tea"
)
/**
 * Create client初始化公共请求参数。
 */
func CreateClient() (_result *eventbridge.Client, _err error) {
    config := &eventbridge.Config{}
    // 您的AccessKey ID。
    config.AccessKeyId = tea.String("<accessKeyId>")
    // 您的AccessKey Secret。
    config.AccessKeySecret = tea.String("<accessKeySecret>")
    // 初始化、上

```

```
// 您的接入点。
config.Endpoint = tea.String("<endpoint>")
_result = &eventbridge.Client{}
_result, _err = eventbridge.NewClient(config)
return _result, _err
}

func UpdateEventRuleSample(client *eventbridge.Client) (_err error) {
    tryErr := func() (_e error) {
        defer func() {
            if r := tea.Recover(recover()); r != nil {
                _e = r
            }
        }()
        updateEventRuleRequest := &eventbridge.UpdateRuleRequest{}
        updateEventRuleRequest.EventBusName = tea.String("demo-bus")
        updateEventRuleRequest.RuleName = tea.String("myRule")
        updateEventRuleRequest.FilterPattern = tea.String("{\"source\":\"acs.oss\",\"type\":\"oss:BucketQueried:GetBucketStat\"}")
        _, _err = client.UpdateRule(updateEventRuleRequest)
        if _err != nil {
            return _err
        }
        console.Log(tea.String("-----update rule success-----"))
        return nil
    }()
    _err = tryErr
    if tryErr != nil {
        var error = &tea.SDKError{}
        if _t, ok := tryErr.(*tea.SDKError); ok {
            error = _t
        } else {
            error.SetErrMsg(tryErr.Error())
        }
        console.Log(error.Message)
    }
    return _err
}

func main() {
    client, _err := CreateClient()
    if _err != nil {
        panic(_err)
    }
}
```

```
}
_err = UpdateEventRuleSample(client)
if _err != nil {
    panic(_err)
}
}
```

### 1.3.6.7. ListRules

本文介绍如何使用Go SDK查询一个事件总线的所有事件规则。

示例代码如下：

```
package main
import (
    "fmt"
    eventbridge "github.com/alibabacloud-go/eventbridge-sdk/eventbridge"
    console "github.com/alibabacloud-go/tea-console/client"
    "github.com/alibabacloud-go/tea/tea"
)
/**
 * Create client初始化公共请求参数。
 */
func CreateClient() (_result *eventbridge.Client, _err error) {
    config := &eventbridge.Config{}
    // 您的AccessKey ID。
    config.AccessKeyId = tea.String("<accessKeyId>")
    // 您的AccessKey Secret。
    config.AccessKeySecret = tea.String("<accessKeySecret>")
    // 您的接入点。
    config.Endpoint = tea.String("<endpoint>")
    _result = &eventbridge.Client{}
    _result, _err = eventbridge.NewClient(config)
    return _result, _err
}
func ListEventRuleSample(client *eventbridge.Client) (_err error) {
    tryErr := func() (_e error) {
        defer func() {
            if r := tea.Recover(recover()); r != nil {
                _e = r
            }
        }()
    }()
```

```
//
listEventRulesRequest := &eventbridge.ListRulesRequest{}
listEventRulesRequest.EventBusName = tea.String("demo-bus")
resp, _err := client.ListRules(listEventRulesRequest)
if _err != nil {
    return _err
}
fmt.Println(resp)
console.Log(tea.String("-----list rule success-----"))
return nil
}()
_err = tryErr
if tryErr != nil {
    var error = &tea.SDKError{}
    if _t, ok := tryErr.(*tea.SDKError); ok {
        error = _t
    } else {
        error.SetErrMsg(tryErr.Error())
    }
    console.Log(error.Message)
}
return _err
}
func main() {
    client, _err := CreateClient()
    if _err != nil {
        panic(_err)
    }
    _err = ListEventRuleSample(client)
    if _err != nil {
        panic(_err)
    }
}
```

### 1.3.6.8. DeleteRule

本文介绍如何使用Go SDK删除一个事件规则。

示例代码如下：

```
package main
import (
```

```

eventbridge "github.com/alibabacloud-go/eventbridge-sdk/eventbridge"
console "github.com/alibabacloud-go/tea-console/client"
"github.com/alibabacloud-go/tea/tea"
)
/**
 * Create client初始化公共请求参数。
 */
func CreateClient() (_result *eventbridge.Client, _err error) {
    config := &eventbridge.Config{}
    // 您的AccessKey ID。
    config.AccessKeyId = tea.String("<accessKeyId>")
    // 您的AccessKey Secret。
    config.AccessKeySecret = tea.String("<accessKeySecret>")
    // 您的接入点。
    config.Endpoint = tea.String("<endpoint>")
    _result = &eventbridge.Client{}
    _result, _err = eventbridge.NewClient(config)
    return _result, _err
}

func DeleteEventRuleSample(client *eventbridge.Client) (_err error) {
    tryErr := func() (_e error) {
        defer func() {
            if r := tea.Recover(recover()); r != nil {
                _e = r
            }
        }()
        deleteRuleRequest := &eventbridge.DeleteRuleRequest{}
        deleteRuleRequest.RuleName = tea.String("myRule")
        deleteRuleRequest.EventBusName = tea.String("demo-bus")
        _, _err = client.DeleteRule(deleteRuleRequest)
        if _err != nil {
            return _err
        }
        console.Log(tea.String("-----delete rule success-----"))
        return nil
    }()
    _err = tryErr
    if tryErr != nil {
        var error = &tea.SDKError{}
        if _t, ok := tryErr.(*tea.SDKError); ok {
            error = _t
        }
    }
}

```

```
    } else {
        error.SetErrMsg(tryErr.Error())
    }
    console.Log(error.Message)
}
return _err
}
func main() {
    client, _err := CreateClient()
    if _err != nil {
        panic(_err)
    }
    _err = DeleteEventRuleSample(client)
    if _err != nil {
        panic(_err)
    }
}
```

## 1.3.7. 事件

### 1.3.7.1. 事件概述

您可以通过Go SDK调用事件接口发布事件。

Go SDK提供以下事件接口：

- [PutEvents](#)

### 1.3.7.2. PutEvents

本文介绍如何使用Go SDK发布一个或多个事件。

示例代码如下：

```
package main
import (
    eventbridge "github.com/alibabacloud-go/eventbridge-sdk/eventbridge"
    console "github.com/alibabacloud-go/tea-console/client"
    util "github.com/alibabacloud-go/tea-utils/service"
    "github.com/alibabacloud-go/tea/tea"
)
/**
 * Create client初始化公共请求参数。
 */
```

```

/
func CreateClient() (_result *eventbridge.Client, _err error) {
    config := &eventbridge.Config{}
    // 您的AccessKey ID。
    config.AccessKeyId = tea.String("<accessKeyId>")
    // 您的AccessKey Secret。
    config.AccessKeySecret = tea.String("<accessKeySecret>")
    // 您的接入点。
    config.Endpoint = tea.String("<endpoint>")
    _result = &eventbridge.Client{}
    _result, _err = eventbridge.NewClient(config)
    return _result, _err
}
/**
 * PutEvents
 */
func PutEvents(client *eventbridge.Client) (_err error) {
    event := &eventbridge.CloudEvent{}
    event.Datacontenttype = tea.String("application/json")
    event.Data = util.ToBytes(tea.String("test"))
    event.Id = tea.String("a5074581-7e74-4e4c-868f-47e7afdf8445")
    event.Source = tea.String("acs.oss")
    event.Specversion = tea.String("1.0")
    event.Type = tea.String("oss:ObjectCreated:PostObject")
    event.Time = tea.String("2020-08-24T13:54:05.965Asia/Shanghai")
    event.Subject = tea.String("1.0")
    event.Type = tea.String("acs:oss:cn-hangzhou:1234567:xls-papk/game_apk/123.jpg")
    event.Extensions = map[string]interface{}{
        "aliyuneventbusname": "demo-bus",
    }
    tryErr := func() (_e error) {
        defer func() {
            if r := tea.Recover(recover()); r != nil {
                _e = r
            }
        }()
        resp, _err := client.PutEvents([]*eventbridge.CloudEvent{event})
        if _err != nil {
            return _err
        }
        console.Log(tea.String("-----Publish event to the aliyun EventBus-----"))
    }
}

```

```
    console.Log(util.ToJSONString(tea.ToMap(resp)))
    return nil
}()
_err = tryErr
if tryErr != nil {
    var error = &tea.SDKError{}
    if _t, ok := tryErr.(*tea.SDKError); ok {
        error = _t
    } else {
        error.SetErrMsg(tryErr.Error())
    }
    console.Log(error.Message)
}
return _err
}
func main() {
    client, _err := CreateClient()
    if _err != nil {
        panic(_err)
    }
    _err = PutEvents(client)
    if _err != nil {
        panic(_err)
    }
}
```

## 1.3.8. 事件目标

### 1.3.8.1. 事件目标概述

您可以通过Go SDK调用事件目标接口创建事件目标、删除事件目标和查询事件目标。

Go SDK提供以下事件目标接口：

- [CreateTargets](#)
- [ListTargets](#)
- [DeleteTargets](#)

### 1.3.8.2. CreateTargets

本文介绍如何使用Go SDK为一个事件规则创建一个或多个投递目标。

示例代码如下：

```

package main

import (
    eventbridge "github.com/alibabacloud-go/eventbridge-sdk/eventbridge"
    console "github.com/alibabacloud-go/tea-console/client"
    util "github.com/alibabacloud-go/tea-utils/service"
    "github.com/alibabacloud-go/tea/tea"
)

/**
 * Create client初始化公共请求参数。
 */

func CreateClient() (_result *eventbridge.Client, _err error) {
    config := &eventbridge.Config{}
    // 您的AccessKey ID。
    config.AccessKeyId = tea.String("<accessKeyId>")
    // 您的AccessKey Secret。
    config.AccessKeySecret = tea.String("<accessKeySecret>")
    // 您的接入点。
    config.Endpoint = tea.String("<endpoint>")
    _result = &eventbridge.Client{}
    _result, _err = eventbridge.NewClient(config)
    return _result, _err
}

func CreateTargetsSample(client *eventbridge.Client) (_err error) {
    tryErr := func() (_e error) {
        defer func() {
            if r := tea.Recover(recover()); r != nil {
                _e = r
            }
        }()
        createTargetsRequest := &eventbridge.CreateTargetsRequest{}
        createTargetsRequest.EventBusName = tea.String("demo-bus")
        createTargetsRequest.RuleName = tea.String("myRule3")
        targetEntry := &eventbridge.TargetEntry{}
        targetEntry.Id = tea.String("1234")
        targetEntry.Endpoint = tea.String("http://www.baidu.com")
        list := []*eventbridge.TargetEntry{targetEntry}
        createTargetsRequest.Targets = list
        response, _err := client.CreateTargets(createTargetsRequest)
        if _err != nil {
            return _err
        }
    }()
    return _err
}

```

```
}
    console.Log(tea.String("-----create targets success-----"))
    console.Log(util.ToJSONString(tea.ToMap(response)))
    return nil
}()
_err = tryErr
if tryErr != nil {
    var error = &tea.SDKError{}
    if _t, ok := tryErr.(*tea.SDKError); ok {
        error = _t
    } else {
        error.SetErrMsg(tryErr.Error())
    }
    console.Log(error.Message)
}
return _err
}
func main() {
    client, _err := CreateClient()
    if _err != nil {
        panic(_err)
    }
    _err = CreateTargetsSample(client)
    if _err != nil {
        panic(_err)
    }
}
```

### 1.3.8.3. ListTargets

本文介绍如何使用Go SDK查询一个事件规则的所有投递目标。

示例代码如下：

```
package main
import (
    eventbridge "github.com/alibabacloud-go/eventbridge-sdk/eventbridge"
    console "github.com/alibabacloud-go/tea-console/client"
    util "github.com/alibabacloud-go/tea-utils/service"
    "github.com/alibabacloud-go/tea/tea"
)
/**
```

```

* Create client初始化公共请求参数。
*/
func CreateClient() (_result *eventbridge.Client, _err error) {
    config := &eventbridge.Config{}
    // 您的AccessKey ID。
    config.AccessKeyId = tea.String("<accessKeyId>")
    // 您的AccessKey Secret。
    config.AccessKeySecret = tea.String("<accessKeySecret>")
    // 您的接入点。
    config.Endpoint = tea.String("<endpoint>")
    _result = &eventbridge.Client{}
    _result, _err = eventbridge.NewClient(config)
    return _result, _err
}

func ListTargetsSample(client *eventbridge.Client) (_err error) {
    tryErr := func() (_e error) {
        defer func() {
            if r := tea.Recover(recover()); r != nil {
                _e = r
            }
        }()
        listTargetsRequest := &eventbridge.ListTargetsRequest{}
        listTargetsRequest.EventBusName = tea.String("demo-bus")
        listTargetsRequest.RuleName = tea.String("myRule")
        response, _err := client.ListTargets(listTargetsRequest)
        if _err != nil {
            return _err
        }
        console.Log(tea.String("-----list targets success-----"))
        console.Log(util.ToJSONString(tea.ToMap(response)))
        return nil
    }()
    _err = tryErr
    if tryErr != nil {
        var error = &tea.SDKError{}
        if _t, ok := tryErr.(*tea.SDKError); ok {
            error = _t
        } else {
            error.SetErrMsg(tryErr.Error())
        }
        console.Log(error.Message)
    }
}

```

```
    }
    return _err
}
func main() {
    client, _err := CreateClient()
    if _err != nil {
        panic(_err)
    }
    _err = ListTargetsSample(client)
    if _err != nil {
        panic(_err)
    }
}
```

### 1.3.8.4. DeleteTargets

本文介绍如何使用Go SDK删除一个事件规则的一个或多个投递目标。

示例代码如下：

```
package main
import (
    eventbridge "github.com/alibabacloud-go/eventbridge-sdk/eventbridge"
    console "github.com/alibabacloud-go/tea-console/client"
    util "github.com/alibabacloud-go/tea-utils/service"
    "github.com/alibabacloud-go/tea/tea"
)
/**
 * Create client初始化公共请求参数。
 */
func CreateClient() (_result *eventbridge.Client, _err error) {
    config := &eventbridge.Config{}
    // 您的AccessKey ID。
    config.AccessKeyId = tea.String("<accessKeyId>")
    // 您的AccessKey Secret。
    config.AccessKeySecret = tea.String("<accessKeySecret>")
    // 您的接入点。
    config.Endpoint = tea.String("<endpoint>")
    _result = &eventbridge.Client{}
    _result, _err = eventbridge.NewClient(config)
    return _result, _err
}
```

```

}

func DeleteTargetsSample(client *eventbridge.Client) (_err error) {
    tryErr := func() (_e error) {
        defer func() {
            if r := tea.Recover(recover()); r != nil {
                _e = r
            }
        }()
        deleteTargetsResponse := &eventbridge.DeleteTargetsRequest{}
        deleteTargetsResponse.EventBusName = tea.String("demo-bus")
        deleteTargetsResponse.RuleName = tea.String("myRule")
        list := []*string{tea.String("1234")}
        deleteTargetsResponse.TargetIds = list
        response, _err := client.DeleteTargets(deleteTargetsResponse)
        if _err != nil {
            return _err
        }
        console.Log(tea.String("-----delete targets success-----"))
        console.Log(util.ToJSONString(tea.ToMap(response)))
        return nil
    }()
    _err = tryErr
    if tryErr != nil {
        var error = &tea.SDKError{}
        if _t, ok := tryErr.(*tea.SDKError); ok {
            error = _t
        } else {
            error.SetErrMsg(tryErr.Error())
        }
        console.Log(error.Message)
    }
    return _err
}

func main() {
    client, _err := CreateClient()
    if _err != nil {
        panic(_err)
    }
    _err = DeleteTargetsSample(client)
    if _err != nil {
        panic(_err)
    }
}

```

```
}  
}
```

## 1.4. Python SDK

### 1.4.1. 概述

本文介绍事件总线EventBridge Python SDK中包含的内容。

#### 内容介绍

事件总线EventBridge Python SDK提供下表中丰富的文档内容，您在实际操作过程中可以参见这些内容。

| 文档                           | 说明                        |
|------------------------------|---------------------------|
| <a href="#">版本说明</a>         | 介绍Python SDK的版本更新记录。      |
| <a href="#">安装Python SDK</a> | 介绍如何安装Python SDK。         |
| <a href="#">快速入门</a>         | 介绍如何快速使用Python SDK完成常见操作。 |
| <a href="#">事件总线</a>         | 介绍如何使用Python SDK管理事件总线。   |
| <a href="#">事件规则</a>         | 介绍如何使用Python SDK管理事件规则。   |
| <a href="#">事件</a>           | 介绍如何使用Python SDK发布事件。     |
| <a href="#">事件目标</a>         | 介绍如何使用Python SDK管理事件目标。   |

### 1.4.2. 版本说明

本文介绍Python SDK的版本信息，包含下载链接、发布时间、更新说明等，以便您按需获取适用的Python SDK。

#### Version 1.2.x

| 发布时间       | 下载                                                | 更新说明                                                                                                                                       |
|------------|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 2020-10-10 | <a href="#">SDK下载</a><br><a href="#">sample下载</a> | <ul style="list-style-type: none"><li>Event格式兼容CloudEvents生态标准。</li><li>签名算法采用Aliyun Cloud Service通用签名算法。</li><li>客户端代码采用自动生成方式。</li></ul> |

### 1.4.3. 安装Python SDK

本文介绍如何安装事件总线EventBridge Python SDK。

## 环境准备

- 环境要求

安装Python 3.4以上版本。更多信息，请参见[安装Python](#)。

 说明 Python 3.4以上版本自带pip。

- 查看语言版本

执行命令 `python -V` 查看Python语言版本。

## 安装SDK

1. 执行以下命令安装Python SDK。

```
pip install alibabacloud_eventbridge
```

2. 执行以下命令安装打印端口。

```
pip install alibabacloud_tea_console
```

## 1.4.4. 快速入门

本文以发布事件到事件总线Event Bridge为例介绍如何快速使用Python SDK调用API。

### 前提条件

您已完成以下操作：

- [开通事件总线Event Bridge服务](#)
- [创建AccessKey](#)
- [安装Python SDK](#)

### 发布事件

示例代码如下：

```
# -*- coding: utf-8 -*-
from alibabacloud_eventbridge.client import Client as EventBridgeClient
from alibabacloud_eventbridge import models as event_bridge_models
from alibabacloud_tea_console.client import Client as ConsoleClient
from alibabacloud_tea_util.client import Client as UtilClient

class put_events(object):
    def __init__(self):
        pass

    @staticmethod
    def create_client():
        """
        Create client初始化公共请求参数。
        """
```

```

"""
config = event_bridge_models.Config(
)
# 您的AccessKey ID。
config.access_key_id = "<accessKeyId>"
# 您的AccessKey Secret。
config.access_key_secret = "<accessKeySecret>"
# 您的接入点。
config.endpoint = "<endpoint>"
return EventBridgeClient(config)

@staticmethod
def put_events(client):
    """
    PutEvents
    """
    event = event_bridge_models.CloudEvent(
    )
    event.datacontenttype = "application/json"
    event.data = UtilClient.to_bytes("test")
    event.id = "a5074581-7e74-4e4c-868f-47e7afdf*****"
    event.source = "acs.oss"
    event.specversion = "1.0"
    event.type = "oss:ObjectCreated:PostObject"
    event.time = "2020-08-24T13:54:05.965Asia/Shanghai"
    event.subject = "1.0"
    event.type = "acs:oss:cn-hangzhou:1234567:xls-papk/game_apk/123.jpg"
    event.extensions = {
        "aliyuneventbusname": "demo-bus"
    }
    try:
        resp = client.put_events([
            event
        ])
        ConsoleClient.log("-----Publish event to the aliyun EventBus-----")
        ConsoleClient.log(UtilClient.to_jsonstring(resp.to_map()))
    except Exception as error:
        ConsoleClient.log(error.message)

@staticmethod
def main(args):
    client = put_events.create_client()
    put_events.put_events(client)

```

```
put_events.main("")
```

## 1.4.5. 事件总线

### 1.4.5.1. 事件总线概述

您可以通过Python SDK调用事件总线接口创建总线、删除总线、获取总线详情和查询事件总线。

Python SDK提供以下事件总线接口：

- [CreateEventBus](#)
- [GetEventBus](#)
- [DeleteEventBus](#)
- [ListEventBuses](#)

### 1.4.5.2. CreateEventBus

本文介绍如何使用Python SDK创建一个事件总线。

示例代码如下：

```
# -*- coding: utf-8 -*-
from alibabacloud_eventbridge.client import Client as EventBridgeClient
from alibabacloud_eventbridge import models as event_bridge_models
from alibabacloud_tea_console.client import Client as ConsoleClient
from alibabacloud_tea_util.client import Client as UtilClient

class create_event_bus_sample(object):
    def __init__(self):
        pass
    @staticmethod
    def create_client():
        """
        Create client初始化公共请求参数。
        """
        config = event_bridge_models.Config(
        )
        # 您的AccessKey ID。
        config.access_key_id = "<accessKeyId>"
        # 您的AccessKey Secret。
        config.access_key_secret = "<accessKeySecret>"
        # 您的接入点。
        config.endpoint = "<endpoint>"
        return EventBridgeClient(config)
    @staticmethod
    def create_event_bus_sample(client):
        try:
            create_event_bus_request = event_bridge_models.CreateEventBusRequest(
            )
            create_event_bus_request.event_bus_name = "demo-bus"
            response = client.create_event_bus(create_event_bus_request)
            ConsoleClient.log("-----Create bus success -----")
            ConsoleClient.log(UtilClient.to_jsonstring(response.to_map()))
        except Exception as error:
            ConsoleClient.log(error.message)
    @staticmethod
    def main(args):
        client = create_event_bus_sample.create_client()
        create_event_bus_sample.create_event_bus_sample(client)
        create_event_bus_sample.main("")
```

### 1.4.5.3. GetEventBus

本文介绍如何使用Python SDK获取一个事件总线的详细信息。

示例代码如下：

```
# -*- coding: utf-8 -*-
from alibabacloud_eventbridge.client import Client as EventBridgeClient
from alibabacloud_eventbridge import models as event_bridge_models
from alibabacloud_tea_console.client import Client as ConsoleClient
from alibabacloud_tea_util.client import Client as UtilClient

class get_event_bus_sample(object):
    def __init__(self):
        pass
    @staticmethod
    def create_client():
        """
        Create client初始化公共请求参数。
        """
        config = event_bridge_models.Config(
        )
        # 您的AccessKey ID。
        config.access_key_id = "<accessKeyId>"
        # 您的AccessKey Secret。
        config.access_key_secret = "<accessKeySecret>"
        # 您的接入点。
        config.endpoint = "<endpoint>"
        return EventBridgeClient(config)
    @staticmethod
    def get_event_bus_sample(client):
        try:
            describe_event_bus_request = event_bridge_models.GetEventBusRequest(
            )
            describe_event_bus_request.event_bus_name = "demo-bus"
            de_scribe_response = client.get_event_bus(describe_event_bus_request)
            ConsoleClient.log("-----describe bus success -----")
            ConsoleClient.log(UtilClient.to_jsonstring(de_scribe_response.to_map()))
        except Exception as error:
            ConsoleClient.log(error.message)
    @staticmethod
    def main(args):
        client = get_event_bus_sample.create_client()
        get_event_bus_sample.get_event_bus_sample(client)
get_event_bus_sample.main("")
```

## 1.4.5.4. DeleteEventBus

本文介绍如何使用Python SDK删除一个事件总线。

示例代码如下：

```
# -*- coding: utf-8 -*-
from alibabacloud_eventbridge.client import Client as EventBridgeClient
from alibabacloud_eventbridge import models as event_bridge_models
from alibabacloud_tea_console.client import Client as ConsoleClient
from alibabacloud_tea_util.client import Client as UtilClient

class delete_event_bus_sample(object):
    def __init__(self):
        pass
    @staticmethod
    def create_client():
        """
        Create client初始化公共请求参数。
        """
        config = event_bridge_models.Config(
        )
        # 您的AccessKey ID。
        config.access_key_id = "<accessKeyId>"
        # 您的AccessKey Secret。
        config.access_key_secret = "<accessKeySecret>"
        # 您的接入点。
        config.endpoint = "<endpoint>"
        return EventBridgeClient(config)
    @staticmethod
    def delete_event_bus_sample(client):
        try:
            # delete
            delete_event_bus_request = event_bridge_models.DeleteEventBusRequest(
            )
            delete_event_bus_request.event_bus_name = "demo-bus"
            client.delete_event_bus(delete_event_bus_request)
            ConsoleClient.log("-----Delete bus success -----")
        except Exception as error:
            ConsoleClient.log(error.message)
    @staticmethod
    def main(args):
        client = delete_event_bus_sample.create_client()
        delete_event_bus_sample.delete_event_bus_sample(client)
delete_event_bus_sample.main("")
```

## 1.4.5.5. ListEventBuses

本文介绍如何使用Python SDK查询所有事件总线。

示例代码如下：

```
# -*- coding: utf-8 -*-
from alibabacloud_eventbridge.client import Client as EventBridgeClient
from alibabacloud_eventbridge import models as event_bridge_models
from alibabacloud_tea_console.client import Client as ConsoleClient
from alibabacloud_tea_util.client import Client as UtilClient

class list_event_bus_sample(object):
    def __init__(self):
        pass
    @staticmethod
    def create_client():
        """
        Create client初始化公共请求参数。
        """
        config = event_bridge_models.Config(
        )
        # 您的AccessKey ID。
        config.access_key_id = "<accessKeyId>"
        # 您的AccessKey Secret。
        config.access_key_secret = "<accessKeySecret>"
        # 您的接入点。
        config.endpoint = "<endpoint>"
        return EventBridgeClient(config)
    @staticmethod
    def list_event_bus_sample(client):
        try:
            list_event_buses_request = event_bridge_models.ListEventBusesRequest(
            )
            list_event_buses_request.limit = 100
            response = client.list_event_buses(list_event_buses_request)
            ConsoleClient.log("-----list bus success -----")
            ConsoleClient.log(UtilClient.to_jsonstring(response.to_map()))
        except Exception as error:
            ConsoleClient.log(error.message)
    @staticmethod
    def main(args):
        client = list_event_bus_sample.create_client()
        list_event_bus_sample.list_event_bus_sample(client)
list_event_bus_sample.main("")
```

## 1.4.6. 事件规则

### 1.4.6.1. 事件规则概述

您可以通过Python SDK调用事件规则接口创建规则、禁用规则、启用规则、删除规则等。

Python SDK提供以下事件规则接口：

- [CreateRule](#)
- [GetRule](#)
- [EnableRule](#)
- [DisableRule](#)
- [UpdateRule](#)
- [ListRules](#)
- [DeleteRule](#)

### 1.4.6.2. CreateRule

本文介绍如何使用Python SDK创建一个事件规则。

示例代码如下：

```
# -*- coding: utf-8 -*-
from alibabacloud_eventbridge.client import Client as EventBridgeClient
from alibabacloud_eventbridge import models as event_bridge_models
from alibabacloud_tea_console.client import Client as ConsoleClient
from alibabacloud_tea_util.client import Client as UtilClient

class create_event_rule_sample(object):
    def __init__(self):
        pass

    @staticmethod
    def create_client():
        """
        Create client初始化公共请求参数。
        """
        config = event_bridge_models.Config(
        )
        # 您的AccessKey ID。
        config.access_key_id = "<accessKeyId>"
        # 您的AccessKey Secret。
        config.access_key_secret = "<accessKeySecret>"
        # 您的接入点。
        config.endpoint = "<endpoint>"
        return EventBridgeClient(config)
```

```
@staticmethod
def create_event_rule_sample(client):
    try:
        create_event_rule_request = event_bridge_models.CreateRuleRequest(
        )
        target_entry = event_bridge_models.TargetEntry(
        )
        target_entry.id = "1234"
        target_entry.endpoint = "http://www.baidu.com"
        target_entry_list = [
            target_entry
        ]
        # targetEntryList[0]=targetEntry;
        create_event_rule_request.rule_name = "myRule"
        create_event_rule_request.event_bus_name = "demo-bus"
        create_event_rule_request.filter_pattern = "{\"source\":[\"acs.oss\"],\"type\":[\"oss:ObjectCreated:Up
loadPart\"]}"
        create_event_rule_request.status = "enable"
        create_event_rule_request.targets = target_entry_list
        response = client.create_rule(create_event_rule_request)
        ConsoleClient.log("-----create rule success-----")
        ConsoleClient.log(UtilClient.to_jsonstring(response.to_map()))
    except Exception as error:
        ConsoleClient.log(error.message)

@staticmethod
def main(args):
    client = create_event_rule_sample.create_client()
    create_event_rule_sample.create_event_rule_sample(client)
create_event_rule_sample.main("")
```

### 1.4.6.3. GetRule

本文介绍如何使用Python SDK获取一个事件规则的详细信息。

示例代码如下：

```
# -*- coding: utf-8 -*-
from alibabacloud_eventbridge.client import Client as EventBridgeClient
from alibabacloud_eventbridge import models as event_bridge_models
from alibabacloud_tea_console.client import Client as ConsoleClient
from alibabacloud_tea_util.client import Client as UtilClient

class get_event_rule_sample(object):
    def __init__(self):
        pass
    @staticmethod
    def create_client():
        """
        Create client初始化公共请求参数。
        """
        config = event_bridge_models.Config(
        )
        # 您的AccessKey ID。
        config.access_key_id = "<accessKeyId>"
        # 您的AccessKey Secret。
        config.access_key_secret = "<accessKeySecret>"
        # 您的接入点。
        config.endpoint = "<endpoint>"
        return EventBridgeClient(config)
    @staticmethod
    def get_event_rule_sample(client):
        try:
            get_event_rule_request = event_bridge_models.GetRuleRequest(
            )
            get_event_rule_request.rule_name = "myRule"
            get_event_rule_request.event_bus_name = "demo-bus"
            client.get_rule(get_event_rule_request)
            ConsoleClient.log("-----describe rule success-----")
        except Exception as error:
            ConsoleClient.log(error.message)
    @staticmethod
    def main(args):
        client = get_event_rule_sample.create_client()
        get_event_rule_sample.get_event_rule_sample(client)
get_event_rule_sample.main("")
```

## 1.4.6.4. EnableRule

本文介绍如何使用Python SDK启用一个事件规则。

示例代码如下：

```
# -*- coding: utf-8 -*-
from alibabacloud_eventbridge.client import Client as EventBridgeClient
from alibabacloud_eventbridge import models as event_bridge_models
from alibabacloud_tea_console.client import Client as ConsoleClient
from alibabacloud_tea_util.client import Client as UtilClient

class enable_event_rule_sample(object):
    def __init__(self):
        pass
    @staticmethod
    def create_client():
        """
        Create client初始化公共请求参数。
        """
        config = event_bridge_models.Config(
        )
        # 您的AccessKey ID。
        config.access_key_id = "<accessKeyId>"
        # 您的AccessKey Secret。
        config.access_key_secret = "<accessKeySecret>"
        # 您的接入点。
        config.endpoint = "<endpoint>"
        return EventBridgeClient(config)
    @staticmethod
    def enable_event_rule_sample(client):
        try:
            enable_event_rule_request = event_bridge_models.EnableRuleRequest(
            )
            enable_event_rule_request.rule_name = "myRule"
            enable_event_rule_request.event_bus_name = "demo-bus"
            client.enable_rule(enable_event_rule_request)
            ConsoleClient.log("-----enable rule success-----")
        except Exception as error:
            ConsoleClient.log(error.message)
    @staticmethod
    def main(args):
        client = enable_event_rule_sample.create_client()
        enable_event_rule_sample.enable_event_rule_sample(client)
enable_event_rule_sample.main("")
```

## 1.4.6.5. DisableRule

本文介绍如何使用Python SDK禁用一个事件规则。

示例代码如下：

```
# -*- coding: utf-8 -*-
from alibabacloud_eventbridge.client import Client as EventBridgeClient
from alibabacloud_eventbridge import models as event_bridge_models
from alibabacloud_tea_console.client import Client as ConsoleClient
from alibabacloud_tea_util.client import Client as UtilClient

class disable_event_rule_sample(object):
    def __init__(self):
        pass
    @staticmethod
    def create_client():
        """
        Create client初始化公共请求参数。
        """
        config = event_bridge_models.Config(
        )
        # 您的AccessKey ID。
        config.access_key_id = "<accessKeyId>"
        # 您的AccessKey Secret。
        config.access_key_secret = "<accessKeySecret>"
        # 您的接入点。
        config.endpoint = "<endpoint>"
        return EventBridgeClient(config)
    @staticmethod
    def disable_event_rule_sample(client):
        try:
            disable_event_rule_request = event_bridge_models.DisableRuleRequest(
            )
            disable_event_rule_request.rule_name = "myRule"
            disable_event_rule_request.event_bus_name = "demo-bus"
            client.disable_rule(disable_event_rule_request)
            ConsoleClient.log("-----disable rule success-----")
        except Exception as error:
            ConsoleClient.log(error.message)
    @staticmethod
    def main(args):
        client = disable_event_rule_sample.create_client()
        disable_event_rule_sample.disable_event_rule_sample(client)
        disable_event_rule_sample.main("")
```

## 1.4.6.6. UpdateRule

本文介绍如何使用Python SDK更新一个事件规则的配置。

示例代码如下：

```

# -*- coding: utf-8 -*-
from alibabacloud_eventbridge.client import Client as EventBridgeClient
from alibabacloud_eventbridge import models as event_bridge_models
from alibabacloud_tea_console.client import Client as ConsoleClient
from alibabacloud_tea_util.client import Client as UtilClient

class update_event_rule_sample(object):
    def __init__(self):
        pass
    @staticmethod
    def create_client():
        """
        Create client初始化公共请求参数。
        """
        config = event_bridge_models.Config(
        )
        # 您的AccessKey ID。
        config.access_key_id = "<accessKeyId>"
        # 您的AccessKey Secret。
        config.access_key_secret = "<accessKeySecret>"
        # 您的接入点。
        config.endpoint = "<endpoint>"
        return EventBridgeClient(config)
    @staticmethod
    def update_event_rule_sample(client):
        try:
            update_event_rule_request = event_bridge_models.UpdateRuleRequest(
            )
            update_event_rule_request.event_bus_name = "demo-bus"
            update_event_rule_request.rule_name = "myRule"
            update_event_rule_request.filter_pattern = "{\"source\": \"acs.oss\", \"type\": \"oss:BucketQueried:GetBucketStat\"}"
            client.update_rule(update_event_rule_request)
            ConsoleClient.log("-----update rule success-----")
        except Exception as error:
            ConsoleClient.log(error.message)
    @staticmethod
    def main(args):
        client = update_event_rule_sample.create_client()
        update_event_rule_sample.update_event_rule_sample(client)
        update_event_rule_sample.main("")

```

## 1.4.6.7. ListRules

本文介绍如何使用Python SDK查询一个事件总线的所有规则。

示例代码如下：

```

# -*- coding: utf-8 -*-
from alibabacloud_eventbridge.client import Client as EventBridgeClient
from alibabacloud_eventbridge import models as event_bridge_models
from alibabacloud_tea_console.client import Client as ConsoleClient
from alibabacloud_tea_util.client import Client as UtilClient

class list_event_rule_sample(object):
    def __init__(self):
        pass
    @staticmethod
    def create_client():
        """
        Create client初始化公共请求参数。
        """
        config = event_bridge_models.Config(
        )
        # 您的AccessKey ID。
        config.access_key_id = "<accessKeyId>"
        # 您的AccessKey Secret。
        config.access_key_secret = "<accessKeySecret>"
        # 您的接入点。
        config.endpoint = "<endpoint>"
        return EventBridgeClient(config)
    @staticmethod
    def list_event_rule_sample(client):
        try:
            list_event_rules_request = event_bridge_models.ListRulesRequest(
            )
            list_event_rules_request.event_bus_name = "demo-bus"
            response = client.list_rules(list_event_rules_request)
            ConsoleClient.log("-----listRules rule success-----")
        except Exception as error:
            ConsoleClient.log(error.message)
    @staticmethod
    def main(args):
        client = list_event_rule_sample.create_client()
        list_event_rule_sample.list_event_rule_sample(client)
list_event_rule_sample.main("")

```

## 1.4.6.8. DeleteRule

本文介绍如何使用Python SDK删除一个事件规则。

示例代码如下：

```
# -*- coding: utf-8 -*-
from alibabacloud_eventbridge.client import Client as EventBridgeClient
from alibabacloud_eventbridge import models as event_bridge_models
from alibabacloud_tea_console.client import Client as ConsoleClient
from alibabacloud_tea_util.client import Client as UtilClient

class delete_event_rule_sample(object):
    def __init__(self):
        pass
    @staticmethod
    def create_client():
        """
        Create client初始化公共请求参数。
        """
        config = event_bridge_models.Config(
        )
        # 您的AccessKey ID。
        config.access_key_id = "<accessKeyId>"
        # 您的AccessKey Secret。
        config.access_key_secret = "<accessKeySecret>"
        # 您的接入点。
        config.endpoint = "<endpoint>"
        return EventBridgeClient(config)
    @staticmethod
    def delete_event_rule_sample(client):
        try:
            delete_rule_request = event_bridge_models.DeleteRuleRequest(
            )
            delete_rule_request.rule_name = "myRule"
            delete_rule_request.event_bus_name = "demo-bus"
            client.delete_rule(delete_rule_request)
            ConsoleClient.log("-----delete rule success-----")
        except Exception as error:
            ConsoleClient.log(error.message)
    @staticmethod
    def main(args):
        client = delete_event_rule_sample.create_client()
        delete_event_rule_sample.delete_event_rule_sample(client)
delete_event_rule_sample.main("")
```

## 1.4.7. 事件

### 1.4.7.1. 事件概述

您可以通过Python SDK调用事件接口发布事件。

Python SDK提供以下事件接口：

- [PutEvents](#)

### 1.4.7.2. PutEvents

本文介绍如何使用Python SDK发布一个或多个事件。

示例代码如下：

```
# -*- coding: utf-8 -*-
from alibabacloud_eventbridge.client import Client as EventBridgeClient
from alibabacloud_eventbridge import models as event_bridge_models
from alibabacloud_tea_console.client import Client as ConsoleClient
from alibabacloud_tea_util.client import Client as UtilClient

class put_events(object):
    def __init__(self):
        pass
    @staticmethod
    def create_client():
        """
        Create client初始化公共请求参数。
        """
        config = event_bridge_models.Config(
        )
        # 您的AccessKey ID。
        config.access_key_id = "<accessKeyId>"
        # 您的AccessKey Secret。
        config.access_key_secret = "<accessKeySecret>"
        # 您的接入点。
        config.endpoint = "<endpoint>"
        return EventBridgeClient(config)
    @staticmethod
    def put_events(client):
        """
        PutEvents
        """
        event = event_bridge_models.CloudEvent(
```

```
)
event.datacontenttype = "application/json"
event.data = UtilClient.to_bytes("test")
event.id = "a5074581-7e74-4e4c-868f-47e7afdf*****"
event.source = "acs.oss"
event.specversion = "1.0"
event.type = "oss:ObjectCreated:PostObject"
event.time = "2020-08-24T13:54:05.965Asia/Shanghai"
event.subject = "1.0"
event.type = "acs:oss:cn-hangzhou:1234567:xls-papk/game_apk/123.jpg"
event.extensions = {
    "aliyuneventbusname": "demo-bus"
}
try:
    resp = client.put_events([
        event
    ])
    ConsoleClient.log("-----Publish event to the aliyun EventBus-----")
    ConsoleClient.log(UtilClient.to_jsonstring(resp.to_map()))
except Exception as error:
    ConsoleClient.log(error.message)
@staticmethod
def main(args):
    client = put_events.create_client()
    put_events.put_events(client)
put_events.main("")
```

## 1.4.8. 事件目标

### 1.4.8.1. 事件目标概述

您可以通过Python SDK调用事件目标接口创建事件目标、删除事件目标和查询事件目标。

Python SDK提供以下事件目标接口：

- [CreateTargets](#)
- [ListTargets](#)
- [DeleteTargets](#)

### 1.4.8.2. CreateTargets

本文介绍如何使用Python SDK为一个事件规则创建一个或多个投递目标。

示例代码如下：

```
# -*- coding: utf-8 -*-
from alibabacloud_eventbridge.client import Client as EventBridgeClient
from alibabacloud_eventbridge import models as event_bridge_models
from alibabacloud_tea_console.client import Client as ConsoleClient
from alibabacloud_tea_util.client import Client as UtilClient

class create_targets_sample(object):
    def __init__(self):
        pass

    @staticmethod
    def create_client():
        """
        Create client初始化公共请求参数。
        """
        config = event_bridge_models.Config(
        )
        # 您的AccessKey ID。
        config.access_key_id = "<accessKeyId>"
        # 您的AccessKey Secret。
        config.access_key_secret = "<accessKeySecret>"
        # 您的接入点。
        config.endpoint = "<endpoint>"
        return EventBridgeClient(config)

    @staticmethod
    def create_targets_sample(client):
        try:
            create_targets_request = event_bridge_models.CreateTargetsRequest(
            )
            create_targets_request.event_bus_name = "demo-bus"
            create_targets_request.rule_name = "myRule"
            target_entry = event_bridge_models.TargetEntry(
            )
            target_entry.id = "1234"
            target_entry.endpoint = "http://www.baidu.com"
            list = [
                target_entry
            ]
            create_targets_request.targets = list
            response = client.create_targets(create_targets_request)
            ConsoleClient.log("-----create targets success-----")
```

```
        ConsoleClient.log(UtilClient.to_jsonstring(response.to_map()))
    except Exception as error:
        ConsoleClient.log(error.message)
    @staticmethod
    def main(args):
        client = create_targets_sample.create_client()
        create_targets_sample.create_targets_sample(client)
    create_targets_sample.main("")
```

### 1.4.8.3. ListTargets

本文介绍如何使用Python SDK查询一个事件规则的所有投递目标。

示例代码如下：

```
# -*- coding: utf-8 -*-
from alibabacloud_eventbridge.client import Client as EventBridgeClient
from alibabacloud_eventbridge import models as event_bridge_models
from alibabacloud_tea_console.client import Client as ConsoleClient
from alibabacloud_tea_util.client import Client as UtilClient

class list_targets_sample(object):
    def __init__(self):
        pass
    @staticmethod
    def create_client():
        """
        Create client初始化公共请求参数。
        """
        config = event_bridge_models.Config(
        )
        # 您的AccessKey ID。
        config.access_key_id = "<accessKeyId>"
        # 您的AccessKey Secret。
        config.access_key_secret = "<accessKeySecret>"
        # 您的接入点。
        config.endpoint = "<endpoint>"
        return EventBridgeClient(config)
    @staticmethod
    def list_targets_sample(client):
        try:
            list_targets_request = event_bridge_models.ListTargetsRequest(
            )
            list_targets_request.event_bus_name = "demo-bus"
            list_targets_request.rule_name = "myRule"
            response = client.list_targets(list_targets_request)
            ConsoleClient.log("-----list targets success-----")
            ConsoleClient.log(UtilClient.to_jsonstring(response.to_map()))
        except Exception as error:
            ConsoleClient.log(error.message)
    @staticmethod
    def main(args):
        client = list_targets_sample.create_client()
        list_targets_sample.list_targets_sample(client)
list_targets_sample.main("")
```

## 1.4.8.4. DeleteTargets

本文介绍如何使用Python SDK删除一个事件规则下的一个或多个投递目标。

示例代码如下：

```
# -*- coding: utf-8 -*-
from alibabacloud_eventbridge.client import Client as EventBridgeClient
from alibabacloud_eventbridge import models as event_bridge_models
from alibabacloud_tea_console.client import Client as ConsoleClient
from alibabacloud_tea_util.client import Client as UtilClient

class delete_targets_sample(object):
    def __init__(self):
        pass
    @staticmethod
    def create_client():
        """
        Create client初始化公共请求参数。
        """
        config = event_bridge_models.Config(
        )
        # 您的AccessKey ID。
        config.access_key_id = "<accessKeyId>"
        # 您的AccessKey Secret。
        config.access_key_secret = "<accessKeySecret>"
        # 您的接入点。
        config.endpoint = "<endpoint>"
        return EventBridgeClient(config)
    @staticmethod
    def delete_targets_sample(client):
        try:
            delete_targets_response = event_bridge_models.DeleteTargetsRequest(
            )
            delete_targets_response.event_bus_name = "demo-bus"
            delete_targets_response.rule_name = "myRule"
            list = [
                "1234"
            ]
            delete_targets_response.target_ids = list
            response = client.delete_targets(delete_targets_response)
            ConsoleClient.log("-----delete targets success-----")
            ConsoleClient.log(UtilClient.to_jsonstring(response.to_map()))
```

```
except Exception as error:
    ConsoleClient.log(error.message)
@staticmethod
def main(args):
    client = delete_targets_sample.create_client()
    delete_targets_sample.delete_targets_sample(client)
delete_targets_sample.main("")
```

# 1.5. PHP SDK

## 1.5.1. 概述

本文介绍事件总线EventBridge PHP SDK中包含的内容。

### 内容介绍

事件总线EventBridge PHP SDK提供下表中丰富的文档内容，您在实际操作过程中可以参见这些内容。

| 文档        | 说明                     |
|-----------|------------------------|
| 版本说明      | 介绍PHP SDK的版本更新记录。      |
| 安装PHP SDK | 介绍如何安装PHP SDK。         |
| 快速入门      | 介绍如何快速使用PHP SDK完成常见操作。 |
| 事件总线      | 介绍如何快速使用PHP SDK管理事件总线。 |
| 事件规则      | 介绍如何快速使用PHP SDK管理事件规则。 |
| 事件        | 介绍如何快速使用PHP SDK发布事件。   |
| 事件目标      | 介绍如何快速使用PHP SDK管理事件目标。 |

## 1.5.2. 版本说明

本文介绍PHP SDK的版本信息，包含下载链接、发布时间、更新说明等，以便您按需获取适用的PHP SDK。

### Version 1.2.x

| 发布时间 | 下载 | 更新说明 |
|------|----|------|
|------|----|------|

| 发布时间       | 下载                                                | 更新说明                                                                                                                                           |
|------------|---------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| 2020-10-10 | <a href="#">SDK下载</a><br><a href="#">sample下载</a> | <ul style="list-style-type: none"> <li>Event格式兼容CloudEvents生态标准。</li> <li>签名算法采用Aliyun Cloud Service通用签名算法。</li> <li>客户端代码采用自动生成方式。</li> </ul> |

## 1.5.3. 安装PHP SDK

本文介绍如何安装事件总线EventBridge PHP SDK。

### 环境准备

- 环境要求
  - 安装PHP。更多信息，请参见[安装PHP](#)。
  - 安装Composer。更多信息，请参见[安装Composer](#)。

- 查看语言版本

执行命令 `php -v` 查看PHP语言版本。

### 安装SDK

- 执行以下命令安装PHP SDK。

```
composer require alibabacloud/eventbridge
```

- 执行以下命令安装打印端口。

```
composer require alibabacloud/tea-console
```

## 1.5.4. 快速入门

本文以发布事件到事件总线EventBridge为例介绍如何快速使用PHP SDK调用API。

### 前提条件

您已完成以下操作：

- [开通事件总线EventBridge服务](#)
- [创建AccessKey](#)
- [安装PHP SDK](#)

### 发布事件

示例代码如下：

```
<?php
namespace Alibabacloud\Sample;
use AlibabaCloud\SDK\EventBridge\Eventbridge;
```

```
use AlibabaCloud\SDK\EventBridge\EventBridge;
use AlibabaCloud\SDK\EventBridge\Models\CloudEvent;
use AlibabaCloud\SDK\EventBridge\Models\Config;
use AlibabaCloud\Tea\Console\Console;
use AlibabaCloud\Tea\Exception\TeaError;
use AlibabaCloud\Tea\Utils\Utils;
use Exception;
class Client
{
    /**
     * Create client初始化公共请求参数。
     *
     * @return Eventbridge
     */
    public static function createClient()
    {
        $config = new Config([]);
        $config->accessKeyId = ('accessKeyId');
        $config->accessKeySecret = ('ACCESS_KEY_SECRET');
        $config->endpoint = ('EVENT_BRIDGE_ENDPOINT');
        return new Eventbridge($config);
    }
    /**
     * PutEvents
     *
     * @param Eventbridge $client
     *
     * @return void
     */
    public static function PutEvents($client)
    {
        $event = new CloudEvent([]);
        $event->datacontenttype = 'application/json';
        $event->data = Utils::toBytes('test');
        $event->id = 'a5074581-7e74-4e4c-868f-47e7afdf****';
        $event->source = 'acs.oss';
        $event->specversion = '1.0';
        $event->type = 'oss:ObjectCreated:PostObject';
        $event->time = '2020-08-24T13:54:05.965Asia/Shanghai';
        $event->subject = '1.0';
        $event->type = 'acs:oss:cn-hangzhou:1234567:xls-papk/game_apk/123.jpg';
    }
}
```

```

$event->extensions = [
    'aliyuneventbusname' => 'demo-bus',
];
try {
    $resp = $client->putEvents([
        $event,
    ]);
    Console::log('-----Publish event to the aliyun EventBus-----');
    Console::log(Utils::toJSONString($resp->toMap()));
} catch (Exception $error) {
    if (!$error instanceof TeaError) {
        $error = new TeaError([], $error->getMessage(), $error->getCode(), $error);
    }
    Console::log($error->message);
}
}
/**
 * @param string[] $args
 *
 * @return void
 */
public static function main($args)
{
    $client = self::createClient();
    self::PutEvents($client);
}
}
require '../vendor/autoload.php';
Client::main([]);

```

## 1.5.5. 事件总线

### 1.5.5.1. 事件总线概述

您可以通过PHP SDK调用事件总线接口创建总线、删除总线、获取总线详情和查询事件总线。

PHP SDK提供以下事件总线接口：

- [CreateEventBus](#)
- [GetEventBus](#)
- [DeleteEventBus](#)

- List Event Buses

## 1.5.5.2. CreateEventBus

本文介绍如何使用PHP SDK创建一个事件总线。

示例代码如下：

```
<?php
namespace Alibabacloud\Sample;
use AlibabaCloud\SDK\EventBridge\Eventbridge;
use AlibabaCloud\SDK\EventBridge\Models\Config;
use AlibabaCloud\SDK\EventBridge\Models\CreateEventBusRequest;
use AlibabaCloud\Tea\Console\Console;
use AlibabaCloud\Tea\Exception\TeaError;
use AlibabaCloud\Tea\Utils\Utils;
use Exception;
class Client
{
    /**
     * Create client初始化公共请求参数。
     *
     * @return Eventbridge
     */
    public static function createClient()
    {
        $config = new Config([]);
        $config->accessKeyId = ('accessKeyId');
        $config->accessKeySecret = ('ACCESS_KEY_SECRET');
        $config->endpoint = ('EVENT_BRIDGE_ENDPOINT');
        return new Eventbridge($config);
    }
    /**
     * @param Eventbridge $client
     *
     * @return void
     */
    public static function createEventBusSample($client)
    {
        try {
            $createEventBusRequest = new CreateEventBusRequest([]);
            $createEventBusRequest->eventBusName = 'demo-bus';
            $response = $client->createEventBus($createEventBusRequest);
```

```

        $response = $client->createEventBus($createEventBusRequest);
        Console::log('-----Create bus success -----');
        Console::log(Utils::toJSONString($response->toMap()));
    } catch (Exception $error) {
        if (!($error instanceof TeaError)) {
            $error = new TeaError([], $error->getMessage(), $error->getCode(), $error);
        }
        Console::log($error->message);
    }
}
/**
 * @param string[] $args
 *
 * @return void
 */
public static function main($args)
{
    $client = self::createClient();
    self::createEventBusSample($client);
}
}
require '../vendor/autoload.php';
Client::main([]);

```

### 1.5.5.3. GetEventBus

本文介绍如何使用PHP SDK获取一个事件总线的详细信息。

示例代码如下：

```

<?php
namespace Alibabacloud\Sample;
use AlibabaCloud\SDK\EventBridge\Eventbridge;
use AlibabaCloud\SDK\EventBridge\Models\Config;
use AlibabaCloud\SDK\EventBridge\Models\GetEventBusRequest;
use AlibabaCloud\Tea\Console\Console;
use AlibabaCloud\Tea\Exception\TeaError;
use AlibabaCloud\Tea\Utils\Utils;
use Exception;
class Client
{
    /**

```

```
* Create client 初始化公共请求参数。
*
* @return Eventbridge
*/
public static function createClient()
{
    $config = new Config([]);
    $config->accessKeyId = ('accessKeyId');
    $config->accessKeySecret = ('ACCESS_KEY_SECRET');
    $config->endpoint = ('EVENT_BRIDGE_ENDPOINT');
    return new Eventbridge($config);
}
/**
 * @param Eventbridge $client
 *
 * @return void
 */
public static function getEventBusSample($client)
{
    try {
        $describeEventBusRequest = new GetEventBusRequest([]);
        $describeEventBusRequest->eventBusName = 'demo-bus';
        $describeResponse = $client->getEventBus($describeEventBusRequest);
        Console::log('-----get bus success -----');
        Console::log(Utils::toJSONString($describeResponse->toMap()));
    } catch (Exception $error) {
        if (!$error instanceof TeaError) {
            $error = new TeaError([], $error->getMessage(), $error->getCode(), $error);
        }
        Console::log($error->message);
    }
}
/**
 * @param string[] $args
 *
 * @return void
 */
public static function main($args)
{
    $client = self::createClient();
    self::getEventBusSample($client);
}
```

```
}  
}  
require '../vendor/autoload.php';  
Client::main([]);
```

### 1.5.5.4. DeleteEventBus

本文介绍如何使用PHP SDK删除一个事件总线。

示例代码如下：

```
<?php  
namespace Alibabacloud\Sample;  
use AlibabaCloud\SDK\EventBridge\Eventbridge;  
use AlibabaCloud\SDK\EventBridge\Models\Config;  
use AlibabaCloud\SDK\EventBridge\Models>DeleteEventBusRequest;  
use AlibabaCloud\Tea\Console\Console;  
use AlibabaCloud\Tea\Exception\TeaError;  
use Exception;  
class Client  
{  
    /**  
     * Create client 初始化公共请求参数。  
     *  
     * @return Eventbridge  
     */  
    public static function createClient()  
    {  
        $config = new Config([]);  
        $config->accessKeyId = ('accessKeyId');  
        $config->accessKeySecret = ('ACCESS_KEY_SECRET');  
        $config->endpoint = ('EVENT_BRIDGE_ENDPOINT');  
        return new Eventbridge($config);  
    }  
    /**  
     * @param Eventbridge $client  
     *  
     * @return void  
     */  
    public static function deleteEventBusSample($client)  
    {
```

```

try {
    // delete
    $deleteEventBusRequest = new DeleteEventBusRequest([]);
    $deleteEventBusRequest->eventBusName = 'demo-bus';
    $client->deleteEventBus($deleteEventBusRequest);
    Console::log('-----Delete bus success -----');
} catch (Exception $error) {
    if (!$error instanceof TeaError) {
        $error = new TeaError([], $error->getMessage(), $error->getCode(), $error);
    }
    Console::log($error->message);
}
}
/**
 * @param string[] $args
 *
 * @return void
 */
public static function main($args)
{
    $client = self::createClient();
    self::deleteEventBusSample($client);
}
}
require '../vendor/autoload.php';
Client::main([]);

```

### 1.5.5.5. ListEventBuses

本文介绍如何使用PHP SDK查询所有事件总线。

示例代码如下：

```

<?php
namespace Alibabacloud\Sample;
use AlibabaCloud\SDK\EventBridge\Eventbridge;
use AlibabaCloud\SDK\EventBridge\Models\Config;
use AlibabaCloud\SDK\EventBridge\Models>ListEventBusesRequest;
use AlibabaCloud\Tea\Console\Console;
use AlibabaCloud\Tea\Exception\TeaError;
use AlibabaCloud\Tea\Utils\Utils;
use Exception;

```

```
class Client
{
    /**
     * Create client 初始化公共请求参数。
     *
     * @return Eventbridge
     */
    public static function createClient()
    {
        $config = new Config([]);
        $config->accessKeyId = ('accessKeyId');
        $config->accessKeySecret = ('ACCESS_KEY_SECRET');
        $config->endpoint = ('EVENT_BRIDGE_ENDPOINT');
        return new Eventbridge($config);
    }
    /**
     * @param Eventbridge $client
     *
     * @return void
     */
    public static function listEventBusSample($client)
    {
        try {
            $listEventBusesRequest = new ListEventBusesRequest([]);
            $listEventBusesRequest->limit = 100;
            $response = $client->listEventBuses($listEventBusesRequest);
            Console::log('-----list bus success -----');
            Console::log(Utils::toJSONString($response->toMap()));
        } catch (Exception $error) {
            if (!($error instanceof TeaError)) {
                $error = new TeaError([], $error->getMessage(), $error->getCode(), $error);
            }
            Console::log($error->message);
        }
    }
    /**
     * @param string[] $args
     *
     * @return void
     */
    public static function main($args)
```

```
public static function main($args)
{
    $client = self::createClient();
    self::listEventBusSample($client);
}
}
require '../vendor/autoload.php';
Client::main([]);
```

## 1.5.6. 事件规则

### 1.5.6.1. 事件规则概述

您可以通过PHP SDK调用事件规则接口创建规则、禁用规则、启用规则、删除规则等。

PHP SDK提供以下事件规则接口：

- [CreateRule](#)
- [GetRule](#)
- [EnableRule](#)
- [DisableRule](#)
- [UpdateRule](#)
- [ListRules](#)
- [DeleteRule](#)

### 1.5.6.2. CreateRule

本文介绍如何使用PHP SDK创建一个事件规则。

示例代码如下：

```
<?php
namespace Alibabacloud\Sample;
use AlibabaCloud\SDK\EventBridge\Eventbridge;
use AlibabaCloud\SDK\EventBridge\Models\Config;
use AlibabaCloud\SDK\EventBridge\Models>CreateRuleRequest;
use AlibabaCloud\SDK\EventBridge\Models\TargetEntry;
use AlibabaCloud\Tea\Console\Console;
use AlibabaCloud\Tea\Exception\TeaError;
use AlibabaCloud\Tea\Utils\Utils;
use Exception;
class Client
{
    /**
```

```

* Create client 初始化公共请求参数。
*
* @return Eventbridge
*/
public static function createClient()
{
    $config = new Config([]);
    $config->accessKeyId = ('accessKeyId');
    $config->accessKeySecret = ('ACCESS_KEY_SECRET');
    $config->endpoint = ('EVENT_BRIDGE_ENDPOINT');
    return new Eventbridge($config);
}
/**
* @param Eventbridge $client
*
* @return void
*/
public static function createEventRuleSample($client)
{
    try {
        $createEventRuleRequest = new CreateRuleRequest([]);
        $targetEntry = new TargetEntry([]);
        $targetEntry->id = '1234';
        $targetEntry->endpoint = 'http://www.baidu.com';
        $targetEntryList = [
            $targetEntry,
        ];
        // targetEntryList[0] = targetEntry;
        $createEventRuleRequest->ruleName = 'myRule';
        $createEventRuleRequest->eventBusName = 'demo-bus';
        $createEventRuleRequest->filterPattern = '{"source":["acs.oss"],"type":["oss:ObjectCreated:UploadPart"]}';
        $createEventRuleRequest->status = 'enable';
        $createEventRuleRequest->targets = $targetEntryList;
        $response = $client->createRule($createEventRuleRequest);
        Console::log('-----create rule success-----');
        Console::log(Utils::toJSONString($response->toMap()));
    } catch (Exception $error) {
        if (!$error instanceof TeaError) {
            $error = new TeaError([], $error->getMessage(), $error->getCode(), $error);
        }
    }
}

```

```
        Console::log($error->message);
    }
}
/**
 * @param string[] $args
 *
 * @return void
 */
public static function main($args)
{
    $client = self::createClient();
    self::createEventRuleSample($client);
}
}
require '../vendor/autoload.php';
Client::main([]);
```

### 1.5.6.3. GetRule

本文介绍如何使用PHP SDK获取一个事件规则的详细信息。

示例代码如下：

```
<?php
namespace Alibabacloud\Sample;
use AlibabaCloud\SDK\EventBridge\Eventbridge;
use AlibabaCloud\SDK\EventBridge\Models\Config;
use AlibabaCloud\SDK\EventBridge\Models\GetRuleRequest;
use AlibabaCloud\Tea\Console\Console;
use AlibabaCloud\Tea\Exception\TeaError;
use Exception;
class Client
{
    /**
     * Create client 初始化公共请求参数。
     *
     * @return Eventbridge
     */
    public static function createClient()
    {
        $config = new Config([]);
```

```

    $config->accessKeyId = ('accessKeyId');
    $config->accessKeySecret = ('ACCESS_KEY_SECRET');
    $config->endpoint = ('EVENT_BRIDGE_ENDPOINT');
    return new Eventbridge($config);
}
/**
 * @param Eventbridge $client
 *
 * @return void
 */
public static function getEventRuleSample($client)
{
    try {
        $describeEventRuleRequest = new GetRuleRequest([]);
        $describeEventRuleRequest->ruleName = 'myRule';
        $describeEventRuleRequest->eventBusName = 'demo-bus';
        $client->getRule($describeEventRuleRequest);
        Console::log('-----describe rule success-----');
    } catch (Exception $error) {
        if (!$error instanceof TeaError) {
            $error = new TeaError([], $error->getMessage(), $error->getCode(), $error);
        }
        Console::log($error->message);
    }
}
/**
 * @param string[] $args
 *
 * @return void
 */
public static function main($args)
{
    $client = self::createClient();
    self::getEventRuleSample($client);
}
}
require '../vendor/autoload.php';
Client::main([]);

```

### 1.5.6.4. EnableRule

本文介绍如何使用PHP SDK启用一个事件规则。

示例代码如下：

```
<?php
namespace Alibabacloud\Sample;
use AlibabaCloud\SDK\EventBridge\Eventbridge;
use AlibabaCloud\SDK\EventBridge\Models\Config;
use AlibabaCloud\SDK\EventBridge\Models\EnableRuleRequest;
use AlibabaCloud\Tea\Console\Console;
use AlibabaCloud\Tea\Exception\TeaError;
use Exception;
class Client
{
    /**
     * Create client 初始化公共请求参数。
     *
     * @return Eventbridge
     */
    public static function createClient()
    {
        $config = new Config([]);
        $config->accessKeyId = ('accessKeyId');
        $config->accessKeySecret = ('ACCESS_KEY_SECRET');
        $config->endpoint = ('EVENT_BRIDGE_ENDPOINT');
        return new Eventbridge($config);
    }
    /**
     * @param Eventbridge $client
     *
     * @return void
     */
    public static function enableEventRuleSample($client)
    {
        try {
            $enableEventRuleRequest = new EnableRuleRequest([]);
            $enableEventRuleRequest->ruleName = 'myRule';
            $enableEventRuleRequest->eventBusName = 'demo-bus';
            $client->enableRule($enableEventRuleRequest);
            Console::log('-----enable rule success-----');
        } catch (Exception $error) {
            if (!($error instanceof TeaError)) {
```

```

        $error = new TeaError([], $error->getMessage(), $error->getCode(), $error);
    }
    Console::log($error->message);
}
}
/**
 * @param string[] $args
 *
 * @return void
 */
public static function main($args)
{
    $client = self::createClient();
    self::enableEventRuleSample($client);
}
}
require '../vendor/autoload.php';
Client::main([]);

```

### 1.5.6.5. DisableRule

本文介绍如何使用PHP SDK禁用一个事件规则。

示例代码如下：

```

<?php
namespace Alibabacloud\Sample;
use AlibabaCloud\SDK\EventBridge\Eventbridge;
use AlibabaCloud\SDK\EventBridge\Models\Config;
use AlibabaCloud\SDK\EventBridge\Models\DisableRuleRequest;
use AlibabaCloud\Tea\Console\Console;
use AlibabaCloud\Tea\Exception\TeaError;
use Exception;
class Client
{
    /**
     * Create client 初始化公共请求参数。
     *
     * @return Eventbridge
     */
    public static function createClient()
    {

```

```
    {
        $config = new Config([]);
        $config->accessKeyId = ('accessKeyId');
        $config->accessKeySecret = ('ACCESS_KEY_SECRET');
        $config->endpoint = ('EVENT_BRIDGE_ENDPOINT');
        return new Eventbridge($config);
    }
    /**
     * @param Eventbridge $client
     *
     * @return void
     */
    public static function disableEventRuleSample($client)
    {
        try {
            $disableEventRuleRequest = new DisableRuleRequest([]);
            $disableEventRuleRequest->ruleName = 'myRule';
            $disableEventRuleRequest->eventBusName = 'demo-bus';
            $client->disableRule($disableEventRuleRequest);
            Console::log('-----disable rule success-----');
        } catch (Exception $error) {
            if (!$error instanceof TeaError) {
                $error = new TeaError([], $error->getMessage(), $error->getCode(), $error);
            }
            Console::log($error->message);
        }
    }
    /**
     * @param string[] $args
     *
     * @return void
     */
    public static function main($args)
    {
        $client = self::createClient();
        self::disableEventRuleSample($client);
    }
}
require '../vendor/autoload.php';
Client::main([]);
```

## 1.5.6.6. UpdateRule

本文介绍如何使用PHP SDK更新一个事件规则的配置。

示例代码如下：

```
<?php
namespace Alibabacloud\Sample;
use AlibabaCloud\SDK\EventBridge\Eventbridge;
use AlibabaCloud\SDK\EventBridge\Models\Config;
use AlibabaCloud\SDK\EventBridge\Models\UpdateRuleRequest;
use AlibabaCloud\Tea\Console\Console;
use AlibabaCloud\Tea\Exception\TeaError;
use Exception;
class Client
{
    /**
     * Create client 初始化公共请求参数。
     *
     * @return Eventbridge
     */
    public static function createClient()
    {
        $config = new Config([]);
        $config->accessKeyId = ('accessKeyId');
        $config->accessKeySecret = ('ACCESS_KEY_SECRET');
        $config->endpoint = ('EVENT_BRIDGE_ENDPOINT');
        return new Eventbridge($config);
    }
    /**
     * @param Eventbridge $client
     *
     * @return void
     */
    public static function updateEventRuleSample($client)
    {
        try {
            $updateEventRuleRequest = new UpdateRuleRequest([]);
            $updateEventRuleRequest->eventBusName = 'demo-bus';
            $updateEventRuleRequest->ruleName = 'myRule';
            $updateEventRuleRequest->filterPattern = '{"source":["acs.oss"],"type":["oss:BucketQueried:GetBucketStat"]}';
        }
    }
}
```

```

        $client->updateRule($updateEventRuleRequest);
        Console::log('-----update rule success-----');
    } catch (Exception $error) {
        if (!$error instanceof TeaError) {
            $error = new TeaError([], $error->getMessage(), $error->getCode(), $error);
        }
        Console::log($error->message);
    }
}

/**
 * @param string[] $args
 *
 * @return void
 */
public static function main($args)
{
    $client = self::createClient();
    self::updateEventRuleSample($client);
}
}

require '../vendor/autoload.php';
Client::main([]);

```

### 1.5.6.7. ListRules

本文介绍如何使用PHP SDK查询一个事件总线的所有事件规则。

示例代码如下：

```

<?php
namespace Alibabacloud\Sample;
use AlibabaCloud\SDK\EventBridge\Eventbridge;
use AlibabaCloud\SDK\EventBridge\Models\Config;
use AlibabaCloud\SDK\EventBridge\Models>ListRulesRequest;
use AlibabaCloud\Tea\Console\Console;
use AlibabaCloud\Tea\Exception\TeaError;
use Exception;
class Client
{
    /**
     * Create client 初始化公共请求参数。

```

```

*
* @return Eventbridge
*/
public static function createClient()
{
    $config = new Config([]);
    $config->accessKeyId = ('accessKeyId');
    $config->accessKeySecret = ('ACCESS_KEY_SECRET');
    $config->endpoint = ('EVENT_BRIDGE_ENDPOINT');
    return new Eventbridge($config);
}
/**
* @param Eventbridge $client
*
* @return void
*/
public static function listEventRuleSample($client)
{
    try {
        $listEventRulesRequest = new ListRulesRequest([]);
        $listEventRulesRequest->eventBusName = 'demo-bus';
        $response = $client->listRules($listEventRulesRequest);
        Console::log('-----list rule success-----');
    } catch (Exception $error) {
        if (!$error instanceof TeaError) {
            $error = new TeaError([], $error->getMessage(), $error->getCode(), $error);
        }
        Console::log($error->message);
    }
}
/**
* @param string[] $args
*
* @return void
*/
public static function main($args)
{
    $client = self::createClient();
    self::listEventRuleSample($client);
}
}

```

```
require '../vendor/autoload.php';
Client::main([]);
```

## 1.5.6.8. DeleteRule

本文介绍如何使用PHP SDK删除一个事件规则。

示例代码如下：

```
<?php
namespace Alibabacloud\Sample;
use AlibabaCloud\SDK\EventBridge\Eventbridge;
use AlibabaCloud\SDK\EventBridge\Models\Config;
use AlibabaCloud\SDK\EventBridge\Models>DeleteRuleRequest;
use AlibabaCloud\Tea\Console\Console;
use AlibabaCloud\Tea\Exception\TeaError;
use Exception;
class Client
{
    /**
     * Create client 初始化公共请求参数。
     *
     * @return Eventbridge
     */
    public static function createClient()
    {
        $config = new Config([]);
        $config->accessKeyId = ('accessKeyId');
        $config->accessKeySecret = ('ACCESS_KEY_SECRET');
        $config->endpoint = ('EVENT_BRIDGE_ENDPOINT');
        return new Eventbridge($config);
    }
    /**
     * @param Eventbridge $client
     *
     * @return void
     */
    public static function deleteEventRuleSample($client)
    {
        try {
            $deleteRuleRequest = new DeleteRuleRequest([]);
            $deleteRuleRequest->ruleName = 'myRule';
```

```

        $deleteRuleRequest->ruleName = 'myRule',
        $deleteRuleRequest->eventBusName = 'demo-bus';
        $client->deleteRule($deleteRuleRequest);
        Console::log('-----delete rule success-----');
    } catch (Exception $error) {
        if (!$error instanceof TeaError) {
            $error = new TeaError([], $error->getMessage(), $error->getCode(), $error);
        }
        Console::log($error->message);
    }
}
/**
 * @param string[] $args
 *
 * @return void
 */
public static function main($args)
{
    $client = self::createClient();
    self::deleteEventRuleSample($client);
}
}
require '../vendor/autoload.php';
Client::main([]);

```

## 1.5.7. 事件

### 1.5.7.1. 事件概述

您可以通过PHP SDK调用事件接口发布事件。

PHP SDK提供以下事件接口：

- [PutEvents](#)

### 1.5.7.2. PutEvents

本文介绍如何使用PHP SDK发布一个或多个事件。

示例代码如下：

```

<?php
namespace Alibabacloud\Sample;
use AlibabaCloud\SDK\EventBridge\Eventbridge;

```

```
use AlibabaCloud\SDK\EventBridge\Models\CloudEvent;
use AlibabaCloud\SDK\EventBridge\Models\Config;
use AlibabaCloud\Tea\Console\Console;
use AlibabaCloud\Tea\Exception\TeaError;
use AlibabaCloud\Tea\Utils\Utils;
use Exception;
class Client
{
    /**
     * Create client初始化公共请求参数。
     *
     * @return Eventbridge
     */
    public static function createClient()
    {
        $config = new Config([]);
        $config->accessKeyId = ('accessKeyId');
        $config->accessKeySecret = ('ACCESS_KEY_SECRET');
        $config->endpoint = ('EVENT_BRIDGE_ENDPOINT');
        return new Eventbridge($config);
    }
    /**
     * PutEvents
     *
     * @param Eventbridge $client
     *
     * @return void
     */
    public static function PutEvents($client)
    {
        $event = new CloudEvent([]);
        $event->datacontenttype = 'application/json';
        $event->data = Utils::toBytes('test');
        $event->id = 'a5074581-7e74-4e4c-868f-47e7afdf****';
        $event->source = 'acs.oss';
        $event->specversion = '1.0';
        $event->type = 'oss:ObjectCreated:PostObject';
        $event->time = '2020-08-24T13:54:05.965Asia/Shanghai';
        $event->subject = '1.0';
        $event->type = 'acs:oss:cn-hangzhou:1234567:xls-papk/game_apk/123.jpg';
        $event->extensions = [
```

```
'aliyuneventbusname' => 'demo-bus',
];
try {
    $resp = $client->putEvents([
        $event,
    ]);
    Console::log('-----Publish event to the aliyun EventBus-----');
    Console::log(Utils::toJSONString($resp->toMap()));
} catch (Exception $error) {
    if (!$error instanceof TeaError) {
        $error = new TeaError([], $error->getMessage(), $error->getCode(), $error);
    }
    Console::log($error->message);
}
}
/**
 * @param string[] $args
 *
 * @return void
 */
public static function main($args)
{
    $client = self::createClient();
    self::PutEvents($client);
}
}
require '../vendor/autoload.php';
Client::main([]);
```

## 1.5.8. 事件目标

### 1.5.8.1. 事件目标概述

您可以通过PHP SDK调用事件目标接口创建事件目标、删除事件目标和查询事件目标。

PHP SDK提供以下事件目标接口：

- [CreateTargets](#)
- [ListTargets](#)
- [DeleteTargets](#)

## 1.5.8.2. CreateTargets

本文介绍如何使用PHP SDK为一个事件规则创建一个或多个投递目标。

示例代码如下：

```
<?php
namespace Alibabacloud\Sample;
use AlibabaCloud\SDK\EventBridge\Eventbridge;
use AlibabaCloud\SDK\EventBridge\Models\Config;
use AlibabaCloud\SDK\EventBridge\Models\CreateTargetsRequest;
use AlibabaCloud\SDK\EventBridge\Models\TargetEntry;
use AlibabaCloud\Tea\Console\Console;
use AlibabaCloud\Tea\Exception\TeaError;
use AlibabaCloud\Tea\Utils\Utils;
use Exception;
class Client
{
    /**
     * Create client 初始化公共请求参数。
     *
     * @return Eventbridge
     */
    public static function createClient()
    {
        $config = new Config([]);
        $config->accessKeyId = ('accessKeyId');
        $config->accessKeySecret = ('ACCESS_KEY_SECRET');
        $config->endpoint = ('EVENT_BRIDGE_ENDPOINT');
        return new Eventbridge($config);
    }
    /**
     * @param Eventbridge $client
     *
     * @return void
     */
    public static function createTargetsSample($client)
    {
        try {
            $createTargetsRequest = new CreateTargetsRequest([]);
            $createTargetsRequest->eventBusName = 'demo-bus';
            $createTargetsRequest->ruleName = 'myRule';
```

```

        $targetEntry      = new TargetEntry([]);
        $targetEntry->id    = '1234';
        $targetEntry->endpoint    = 'http://www.baidu.com';
        $list              = [
            $targetEntry,
        ];
        $createTargetsRequest->targets = $list;
        $response           = $client->createTargets($createTargetsRequest);
        Console::log('-----create targets success-----');
        Console::log(Utils::toJSONString($response->toMap()));
    } catch (Exception $error) {
        if (!$error instanceof TeaError) {
            $error = new TeaError([], $error->getMessage(), $error->getCode(), $error);
        }
        Console::log($error->message);
    }
}
}
/**
 * @param string[] $args
 *
 * @return void
 */
public static function main($args)
{
    $client = self::createClient();
    self::createTargetsSample($client);
}
}
require '../vendor/autoload.php';
Client::main([]);

```

### 1.5.8.3. ListTargets

本文介绍如何使用PHP SDK查询一个事件规则的所有投递目标。

示例代码如下：

```

<?php
namespace Alibabacloud\Sample;
use AlibabaCloud\SDK\EventBridge\Eventbridge;
use AlibabaCloud\SDK\EventBridge\Models\Config;

```

```
use AlibabaCloud\SDK\EventBridge\Models\ListTargetsRequest;
use AlibabaCloud\Tea\Console\Console;
use AlibabaCloud\Tea\Exception\TeaError;
use AlibabaCloud\Tea\Utils\Utils;
use Exception;
class Client
{
    /**
     * Create client 初始化公共请求参数。
     *
     * @return Eventbridge
     */
    public static function createClient()
    {
        $config = new Config([]);
        $config->accessKeyId = ('accessKeyId');
        $config->accessKeySecret = ('ACCESS_KEY_SECRET');
        $config->endpoint = ('EVENT_BRIDGE_ENDPOINT');
        return new Eventbridge($config);
    }
    /**
     * @param Eventbridge $client
     *
     * @return void
     */
    public static function listTargetsSample($client)
    {
        try {
            $listTargetsRequest = new ListTargetsRequest([]);
            $listTargetsRequest->eventBusName = 'demo-bus';
            $listTargetsRequest->ruleName = 'myRule';
            $response = $client->listTargets($listTargetsRequest);
            Console::log('-----list targets success-----');
            Console::log(Utils::toJSONString($response->toMap()));
        } catch (Exception $error) {
            if (!$error instanceof TeaError) {
                $error = new TeaError([], $error->getMessage(), $error->getCode(), $error);
            }
            Console::log($error->message);
        }
    }
}
```

```
/**
 * @param string[] $args
 *
 * @return void
 */
public static function main($args)
{
    $client = self::createClient();
    self::listTargetsSample($client);
}
}
require '../vendor/autoload.php';
Client::main([]);
```

## 1.5.8.4. DeleteTargets

本文介绍如何使用PHP SDK删除一个事件规则的一个或多个投递目标。

示例代码如下：

```
<?php
namespace Alibabacloud\Sample;
use AlibabaCloud\SDK\EventBridge\Eventbridge;
use AlibabaCloud\SDK\EventBridge\Models\Config;
use AlibabaCloud\SDK\EventBridge\Models>DeleteTargetsRequest;
use AlibabaCloud\Tea\Console\Console;
use AlibabaCloud\Tea\Exception\TeaError;
use AlibabaCloud\Tea\Utils\Utils;
use Exception;
class Client
{
    /**
     * Create client 初始化公共请求参数。
     *
     * @return Eventbridge
     */
    public static function createClient()
    {
        $config = new Config([]);
        $config->accessKeyId = ('accessKeyId');
        $config->accessKeySecret = ('ACCESS_KEY_SECRET');
        $config->endpoint = ('EVENT_BRIDGE_ENDPOINT');
```

```

$config->endpoint = ( EVENT_BRIDGE_ENDPOINT ),
return new Eventbridge($config);
}
/**
 * @param Eventbridge $client
 *
 * @return void
 */
public static function deleteTargetsSample($client)
{
    try {
        $removeTargetsResponse = new DeleteTargetsRequest([]);
        $removeTargetsResponse->eventBusName = 'demo-bus';
        $removeTargetsResponse->ruleName = 'myRule';
        $list = [
            '1234',
        ];
        $removeTargetsResponse->targetIds = $list;
        $response = $client->deleteTargets($removeTargetsResponse);
        Console::log('-----delete targets success-----');
        Console::log(Utils::toJSONString($response->toMap()));
    } catch (Exception $error) {
        if (!$error instanceof TeaError) {
            $error = new TeaError([], $error->getMessage(), $error->getCode(), $error);
        }
        Console::log($error->message);
    }
}
/**
 * @param string[] $args
 *
 * @return void
 */
public static function main($args)
{
    $client = self::createClient();
    self::deleteTargetsSample($client);
}
}
require '../vendor/autoload.php';
Client::main([]);

```

## 1.6. C# SDK

### 1.6.1. 概述

本文介绍事件总线EventBridge C# SDK中包含的内容。

#### 内容介绍

事件总线EventBridge C# SDK提供下表中丰富的文档内容，您在实际操作过程中可以参见这些内容。

| 文档                       | 说明                    |
|--------------------------|-----------------------|
| <a href="#">版本说明</a>     | 介绍C# SDK的版本更新记录。      |
| <a href="#">安装C# SDK</a> | 介绍如何安装C# SDK。         |
| <a href="#">快速入门</a>     | 介绍如何快速使用C# SDK完成常见操作。 |
| <a href="#">事件总线</a>     | 介绍如何使用C# SDK管理事件总线。   |
| <a href="#">事件规则</a>     | 介绍如何快速使用C# SDK管理事件规则。 |
| <a href="#">事件</a>       | 介绍如何快速使用C# SDK发布事件。   |
| <a href="#">事件目标</a>     | 介绍如何快速使用C# SDK管理事件目标。 |

### 1.6.2. 版本说明

本文介绍C# SDK的版本信息，包含下载链接、发布时间、更新说明等，以便您按需获取适用的C# SDK。

#### Version 1.2.x

| 发布时间       | 下载                                                | 更新说明                                                                                                                                       |
|------------|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 2020-10-10 | <a href="#">SDK下载</a><br><a href="#">sample下载</a> | <ul style="list-style-type: none"><li>Event格式兼容CloudEvents生态标准。</li><li>签名算法采用Aliyun Cloud Service通用签名算法。</li><li>客户端代码采用自动生成方式。</li></ul> |

### 1.6.3. 安装C# SDK

本文介绍如何安装事件总线EventBridge C# SDK。

#### 环境准备

- 环境要求
  - 安装.NET 2.0以上版本。更多信息，请参见[安装.NET](#)。

- 安装Visual Studio 2010以上版本。更多信息，请参见[安装Visual Studio](#)。
- 查看语言版本  
执行命令 `dotnet --version` 查看.NET Core版本。

## 安装SDK

在Visual Studio中安装C# SDK。

1. 选择工具 > NuGet包管理器 > 程序包管理器控制台。
2. 执行以下命令安装C# SDK。

```
Install-Package AlibabaCloud.SDK.Eventbridge
```

## 1.6.4. 快速入门

本文以发布事件到事件总线EventBridge为例介绍如何快速使用C# SDK调用API。

### 前提条件

- [开通事件总线EventBridge服务](#)
- [创建AccessKey](#)
- [安装C# SDK](#)

### 发布事件

示例代码如下：

```
using System;
using System.Collections.Generic;
using Tea;
namespace Alibabacloud.Sample
{
    public class Client
    {
        /**
         * Create client初始化公共请求参数。
         */
        public static AlibabaCloud.SDK.EventBridge.EventBridgeClient CreateClient()
        {
            AlibabaCloud.SDK.EventBridge.Models.Config config = new AlibabaCloud.SDK.EventBridge.Models.Config();
            // 您的AccessKey ID。
            config.AccessKeyId = "<accessKeyId>";
            // 您的AccessKey Secret。
            config.AccessKeySecret = "<accessKeySecret>";
            // 您的接入点。
```

```

        config.Endpoint = "<endpoint>";
        return new AlibabaCloud.SDK.EventBridge.EventBridgeClient(config);
    }
    /**
     * PutEvents
     */
    public static void PutEvents(AlibabaCloud.SDK.EventBridge.EventBridgeClient client)
    {
        AlibabaCloud.SDK.EventBridge.Models.CloudEvent event_ = new AlibabaCloud.SDK.EventBridge.Models.CloudEvent();
        event_.Datacontenttype = "application/json";
        event_.Data = AlibabaCloud.TeaUtil.Common.ToBytes("test");
        event_.Id = "a5074581-7e74-4e4c-868f-47e7afdf****";
        event_.Source = "acs.oss";
        event_.Specversion = "1.0";
        event_.Type = "oss:ObjectCreated:PostObject";
        event_.Time = "2020-08-24T13:54:05.965Asia/Shanghai";
        event_.Subject = "1.0";
        event_.Type = "acs:oss:cn-hangzhou:1234567:xls-papk/game_apk/123.jpg";
        event_.Extensions = new Dictionary<string, object>
        { { "aliyuneventbusname", "demo-bus" },
        };
        try
        {
            AlibabaCloud.SDK.EventBridge.Models.PutEventsResponse resp = client.PutEvents(new List<AlibabaCloud.SDK.EventBridge.Models.CloudEvent>
            {
                event_
            });
            Console.WriteLine("-----Publish event to the aliyun EventBus-----");
            Console.WriteLine(AlibabaCloud.TeaUtil.Common.ToJSONString(resp.ToMap()));
        }
        catch (TeaException error)
        {
            Console.WriteLine(error.Message);
        }
        catch (Exception _error)
        {
            TeaException error = new TeaException(new Dictionary<string, object>
            { { "message", _error.Message }
            });
        }
    }

```

```
        Console.WriteLine(error.Message);
    }
}

static void Main(string[] args)
{
    AlibabaCloud.SDK.EventBridge.EventBridgeClient client = Client.CreateClient();
    Client.PutEvents(client);
    Console.ReadKey();
}
}
```

## 1.6.5. 事件总线

### 1.6.5.1. 事件总线概述

您可以通过C# SDK调用事件总线接口创建总线、删除总线、获取总线详情和查询事件总线。

C# SDK提供以下事件总线接口：

- [CreateEventBus](#)
- [GetEventBus](#)
- [DeleteEventBus](#)
- [ListEventBuses](#)

### 1.6.5.2. CreateEventBus

本文介绍如何使用C# SDK创建一个事件总线。

示例代码如下：

```
using System;
using System.Collections.Generic;
using Tea;
namespace Alibabacloud
{
    public class Client
    {
        /**
         * Create client初始化公共请求参数。
         */
        public static AlibabaCloud.SDK.EventBridge.EventBridgeClient CreateClient()
        {

```

```
        AlibabaCloud.SDK.EventBridge.Models.Config config = new AlibabaCloud.SDK.EventBridge.Models.Config();
        // 您的AccessKey ID。
        config.AccessKeyId = "<accessKeyId>";
        // 您的AccessKey Secret。
        config.AccessKeySecret = "<accessKeySecret>";
        // 您的接入点。
        config.Endpoint = "<endpoint>";
        return new AlibabaCloud.SDK.EventBridge.EventBridgeClient(config);
    }

    public static void CreateEventBusSample(AlibabaCloud.SDK.EventBridge.EventBridgeClient client)
    {
        try
        {
            AlibabaCloud.SDK.EventBridge.Models.CreateEventBusRequest createEventBusRequest = new AlibabaCloud.SDK.EventBridge.Models.CreateEventBusRequest();
            createEventBusRequest.EventBusName = "demo-bus";
            AlibabaCloud.SDK.EventBridge.Models.CreateEventBusResponse response = client.CreateEventBus(createEventBusRequest);
            Console.WriteLine("-----Create bus success -----");
            Console.WriteLine(AlibabaCloud.TeaUtil.Common.ToJSONString(response.ToMap()));
        }
        catch (TeaException error)
        {
            Console.WriteLine(error.Message);
        }
        catch (Exception _error)
        {
            TeaException error = new TeaException(new Dictionary<string, object>
            {
                { "message", _error.Message }
            });
            Console.WriteLine(error.Message);
        }
    }

    static void Main(string[] args)
    {
        AlibabaCloud.SDK.EventBridge.EventBridgeClient client = CreateClient();
        CreateEventBusSample(client);
        Console.ReadKey();
    }
}
```

```
}
```

### 1.6.5.3. GetEventBus

本文介绍如何使用C# SDK获取一个事件总线的详细信息。

示例代码如下：

```
using System;
using System.Collections.Generic;
using Tea;
namespace AlibabaCloud.Sample
{
    public class Client
    {
        /**
         * Create client初始化公共请求参数。
         */
        public static AlibabaCloud.SDK.EventBridge.EventBridgeClient CreateClient()
        {
            AlibabaCloud.SDK.EventBridge.Models.Config config = new AlibabaCloud.SDK.EventBridge.Models.Config();
            // 您的AccessKey ID。
            config.AccessKeyId = "<accessKeyId>";
            // 您的AccessKey Secret。
            config.AccessKeySecret = "<accessKeySecret>";
            // 您的接入点。
            config.Endpoint = "<endpoint>";
            return new AlibabaCloud.SDK.EventBridge.EventBridgeClient(config);
        }
        public static void GetEventBusSample(AlibabaCloud.SDK.EventBridge.EventBridgeClient client)
        {
            try
            {
                AlibabaCloud.SDK.EventBridge.Models.GetEventBusRequest describeEventBusRequest = new AlibabaCloud.SDK.EventBridge.Models.GetEventBusRequest();
                describeEventBusRequest.EventBusName = "demo-bus";
                AlibabaCloud.SDK.EventBridge.Models.GetEventBusResponse deScribeResponse = client.GetEventBus(describeEventBusRequest);
                Console.WriteLine("-----get bus success -----");
                Console.WriteLine(AlibabaCloud.TeaUtil.Common.ToJSONString(deScribeResponse.ToMap()));
            }
        }
    }
}
```

```
    }
    catch (TeaException error)
    {
        Console.WriteLine(error.Message);
    }
    catch (Exception _error)
    {
        TeaException error = new TeaException(new Dictionary<string, object>
        { { "message", _error.Message }
        });
        Console.WriteLine(error.Message);
    }
}
static void Main(string[] args)
{
    AlibabaCloud.SDK.EventBridge.EventBridgeClient client = CreateClient();
    GetEventBusSample(client);
    Console.ReadKey();
}
}
```

### 1.6.5.4. DeleteEventBus

本文介绍如何使用C# SDK删除一个事件总线。

示例代码如下：

```
using System;
using System.Collections.Generic;
using Tea;
namespace Alibabacloud.Sample
{
    public class Client
    {
        /**
         * Create client初始化公共请求参数。
         */
        public static AlibabaCloud.SDK.EventBridge.EventBridgeClient CreateClient()
        {
            AlibabaCloud.SDK.EventBridge.Models.Config config = new AlibabaCloud.SDK.EventBridge.Models.Con
            fig();
```

```
// 您的AccessKey ID。
config.AccessKeyId = "<accessKeyId>";
// 您的AccessKey Secret。
config.AccessKeySecret = "<accessKeySecret>";
// 您的接入点。
config.Endpoint = "<endpoint>";
return new AlibabaCloud.SDK.EventBridge.EventBridgeClient(config);
}

public static void DeleteEventBusSample(AlibabaCloud.SDK.EventBridge.EventBridgeClient client)
{
    try
    {
        //delete
        AlibabaCloud.SDK.EventBridge.Models.DeleteEventBusRequest deleteEventBusRequest = new AlibabaCloud.SDK.EventBridge.Models.DeleteEventBusRequest();
        deleteEventBusRequest.EventBusName = "demo-bus";
        client.DeleteEventBus(deleteEventBusRequest);
        Console.WriteLine("-----Delete bus success -----");
    }
    catch (TeaException error)
    {
        Console.WriteLine(error.Message);
    }
    catch (Exception _error)
    {
        TeaException error = new TeaException(new Dictionary<string, object>
        { { "message", _error.Message }
        });
        Console.WriteLine(error.Message);
    }
}

static void Main(string[] args)
{
    AlibabaCloud.SDK.EventBridge.EventBridgeClient client = Client.CreateClient();
    Client.DeleteEventBusSample(client);
    Console.ReadKey();
}
}
```

## 1.6.5.5. ListEventBuses

本文介绍如何使用C# SDK查询所有事件总线。

示例代码如下：

```
using System;
using System.Collections.Generic;
using Tea;
namespace Alibabacloud.Sample
{
    public class Client
    {
        /**
         * Create client初始化公共请求参数。
         */
        public static AlibabaCloud.SDK.EventBridge.EventBridgeClient CreateClient()
        {
            AlibabaCloud.SDK.EventBridge.Models.Config config = new AlibabaCloud.SDK.EventBridge.Models.Config();
            // 您的AccessKey ID。
            config.AccessKeyId = "<accessKeyId>";
            // 您的AccessKey Secret。
            config.AccessKeySecret = "<accessKeySecret>";
            // 您的接入点。
            config.Endpoint = "<endpoint>";
            return new AlibabaCloud.SDK.EventBridge.EventBridgeClient(config);
        }
        public static void ListEventBusSample(AlibabaCloud.SDK.EventBridge.EventBridgeClient client)
        {
            try
            {
                AlibabaCloud.SDK.EventBridge.Models.ListEventBusesRequest listEventBusesRequest = new AlibabaCloud.SDK.EventBridge.Models.ListEventBusesRequest();
                listEventBusesRequest.Limit = 100;
                AlibabaCloud.SDK.EventBridge.Models.ListEventBusesResponse response = client.ListEventBuses(listEventBusesRequest);
                Console.WriteLine("-----list bus success -----");
                Console.WriteLine(AlibabaCloud.TeaUtil.Common.ToJSONString(response.ToMap()));
            }
            catch (TeaException error)
            {
            }
        }
    }
}
```

```
        Console.WriteLine(error.Message);
    }
    catch (Exception _error)
    {
        TeaException error = new TeaException(new Dictionary<string, object>
        { { "message", _error.Message }
        });
        Console.WriteLine(error.Message);
    }
}

static void Main(string[] args)
{
    AlibabaCloud.SDK.EventBridge.EventBridgeClient client = CreateClient();
    ListEventBusSample(client);
    Console.ReadKey();
}
}
```

## 1.6.6. 事件规则

### 1.6.6.1. 事件规则概述

您可以通过C# SDK调用事件规则接口创建规则、禁用规则、启用规则、删除规则等。

C# SDK提供以下事件规则接口：

- [CreateRule](#)
- [GetRule](#)
- [EnableRule](#)
- [DisableRule](#)
- [UpdateRule](#)
- [ListRules](#)
- [DeleteRule](#)

### 1.6.6.2. CreateRule

本文介绍如何使用C# SDK创建一个事件规则。

示例代码如下：

```
using System;
using System.Collections.Generic;
```

```
using Tea;
namespace AlibabaCloud.Sample
{
    public class Client
    {
        /**
         * Create client初始化公共请求参数。
         */
        public static AlibabaCloud.SDK.EventBridge.EventBridgeClient CreateClient()
        {
            AlibabaCloud.SDK.EventBridge.Models.Config config = new AlibabaCloud.SDK.EventBridge.Models.Config();
            // 您的AccessKey ID。
            config.AccessKeyId = "<accessKeyId>";
            // 您的AccessKey Secret。
            config.AccessKeySecret = "<accessKeySecret>";
            // 您的接入点。
            config.Endpoint = "<endpoint>";
            return new AlibabaCloud.SDK.EventBridge.EventBridgeClient(config);
        }
        public static void CreateEventRuleSample(AlibabaCloud.SDK.EventBridge.EventBridgeClient client)
        {
            try
            {
                AlibabaCloud.SDK.EventBridge.Models.CreateRuleRequest createEventRuleRequest = new AlibabaCloud.SDK.EventBridge.Models.CreateRuleRequest();
                AlibabaCloud.SDK.EventBridge.Models.TargetEntry targetEntry = new AlibabaCloud.SDK.EventBridge.Models.TargetEntry();
                targetEntry.Id = "1234";
                targetEntry.Endpoint = "http://www.baidu.com";
                //targetEntry.Type = "{\"resourceKey\":\"url\",\"form\":\"JSONPAHT\",\"value\":\"http://www.baidu.com\"}";
                List<AlibabaCloud.SDK.EventBridge.Models.TargetEntry> targetEntryList = new List<AlibabaCloud.SDK.EventBridge.Models.TargetEntry>
                {
                    targetEntry
                };
                //targetEntryList[0]=targetEntry;
                createEventRuleRequest.RuleName = "myRule";
                createEventRuleRequest.EventBusName = "demo-bus";
                createEventRuleRequest.FilterPattern = "{\"source\":[\"acs.aliyuncvc\"],\"type\":[\"aliyuncvc:MeetingEvent:MeetingStateEvent\"]}";
            }
            catch (Exception e)
            {
                // Handle exception
            }
        }
    }
}
```

```
        createEventRuleRequest.Status = "enable";
        createEventRuleRequest.Targets = targetEntryList;
        AlibabaCloud.SDK.EventBridge.Models.CreateRuleResponse response = client.CreateRule(createEventRuleRequest);
        Console.WriteLine("-----create rule success-----");
        Console.WriteLine(AlibabaCloud.TeaUtil.Common.ToJSONString(response.ToMap()));
    }
    catch (TeaException error)
    {
        Console.WriteLine(error.Message);
    }
    catch (Exception _error)
    {
        TeaException error = new TeaException(new Dictionary<string, object>
        { { "message", _error.Message }
        });
        Console.WriteLine(error.Message);
    }
}

static void Main(string[] args)
{
    AlibabaCloud.SDK.EventBridge.EventBridgeClient client = Client.CreateClient();
    Client.CreateEventRuleSample(client);
    Console.ReadKey();
}
}
```

### 1.6.6.3. GetRule

本文介绍如何使用C# SDK获取一个事件规则的详细信息。

示例代码如下：

```
using System;
using System.Collections.Generic;
using Tea;
namespace Alibabacloud.Sample
{
    public class Client
    {
        /**
```

```

/...
* Create client初始化公共请求参数。
*/
public static AlibabaCloud.SDK.EventBridge.EventBridgeClient CreateClient()
{
    AlibabaCloud.SDK.EventBridge.Models.Config config = new AlibabaCloud.SDK.EventBridge.Models.Con
fig();
    // 您的AccessKey ID。
    config.AccessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.AccessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.Endpoint = "<endpoint>";
    return new AlibabaCloud.SDK.EventBridge.EventBridgeClient(config);
}

public static void GetEventRuleSample(AlibabaCloud.SDK.EventBridge.EventBridgeClient client)
{
    try
    {
        AlibabaCloud.SDK.EventBridge.Models.GetRuleRequest getEventRuleRequest = new AlibabaCloud.S
DK.EventBridge.Models.GetRuleRequest();
        getEventRuleRequest.RuleName = "myRule";
        getEventRuleRequest.EventBusName = "demo-bus";
        client.GetRule(getEventRuleRequest);
        Console.WriteLine("-----Get rule success-----");
    }
    catch (TeaException error)
    {
        Console.WriteLine(error.Message);
    }
    catch (Exception _error)
    {
        TeaException error = new TeaException(new Dictionary<string, object>
        { { "message", _error.Message }
        });
        Console.WriteLine(error.Message);
    }
}

static void Main(string[] args)
{
    AlibabaCloud.SDK.EventBridge.EventBridgeClient client = Client.CreateClient();

```

```
        Client.GetEventRuleSample(client);
        Console.ReadKey();
    }
}
```

## 1.6.6.4. EnableRule

本文介绍如何使用C# SDK启用一个事件规则。

示例代码如下：

```
using System;
using System.Collections.Generic;
using Tea;
namespace Alibabacloud.Sample
{
    public class Client
    {
        /**
         * Create client初始化公共请求参数。
         */
        public static AlibabaCloud.SDK.EventBridge.EventBridgeClient CreateClient()
        {
            AlibabaCloud.SDK.EventBridge.Models.Config config = new AlibabaCloud.SDK.EventBridge.Models.Config();
            // 您的AccessKey ID。
            config.AccessKeyId = "<accessKeyId>";
            // 您的AccessKey Secret。
            config.AccessKeySecret = "<accessKeySecret>";
            // 您的接入点。
            config.Endpoint = "<endpoint>";
            return new AlibabaCloud.SDK.EventBridge.EventBridgeClient(config);
        }
        public static void EnableEventRuleSample(AlibabaCloud.SDK.EventBridge.EventBridgeClient client)
        {
            try
            {
                AlibabaCloud.SDK.EventBridge.Models.EnableRuleRequest enableEventRuleRequest = new AlibabaCloud.SDK.EventBridge.Models.EnableRuleRequest();
                enableEventRuleRequest.RuleName = "myRule";
                enableEventRuleRequest.EventBusName = "demo-bus";
            }
        }
    }
}
```

```

        client.EnableRule(enableEventRuleRequest);
        Console.WriteLine("-----enable rule success-----");
    }
    catch (TeaException error)
    {
        Console.WriteLine(error.Message);
    }
    catch (Exception _error)
    {
        TeaException error = new TeaException(new Dictionary<string, object>
        { { "message", _error.Message }
        });
        Console.WriteLine(error.Message);
    }
}
static void Main(string[] args)
{
    AlibabaCloud.SDK.EventBridge.EventBridgeClient client = Client.CreateClient();
    Client.EnableEventRuleSample(client);
    Console.ReadKey();
}
}
}

```

### 1.6.6.5. DisableRule

本文介绍如何使用C# SDK禁用一个事件规则。

示例代码如下：

```

using System;
using System.Collections.Generic;
using Tea;
namespace Alibabacloud.Sample
{
    public class Client
    {
        /**
         * Create client初始化公共请求参数。
         */
        public static AlibabaCloud.SDK.EventBridge.EventBridgeClient CreateClient()

```

```
{
    AlibabaCloud.SDK.EventBridge.Models.Config config = new AlibabaCloud.SDK.EventBridge.Models.Con
fig();
    // 您的AccessKey ID。
    config.AccessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.AccessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.Endpoint = "<endpoint>";
    return new AlibabaCloud.SDK.EventBridge.EventBridgeClient(config);
}

public static void DisableEventRuleSample(AlibabaCloud.SDK.EventBridge.EventBridgeClient client)
{
    try
    {
        AlibabaCloud.SDK.EventBridge.Models.DisableRuleRequest disableEventRuleRequest = new Alibaba
Cloud.SDK.EventBridge.Models.DisableRuleRequest();
        disableEventRuleRequest.RuleName = "myRule";
        disableEventRuleRequest.EventBusName = "demo-bus";
        client.DisableRule(disableEventRuleRequest);
        Console.WriteLine("-----disable rule success-----");
    }
    catch (TeaException error)
    {
        Console.WriteLine(error.Message);
    }
    catch (Exception _error)
    {
        TeaException error = new TeaException(new Dictionary<string, object>
        { { "message", _error.Message }
        });
        Console.WriteLine(error.Message);
    }
}

static void Main(string[] args)
{
    AlibabaCloud.SDK.EventBridge.EventBridgeClient client = Client.CreateClient();
    Client.DisableEventRuleSample(client);
    Console.ReadKey();
}
}
```

```
}
```

### 1.6.6.6. UpdateRule

本文介绍如何使用C# SDK更新一个事件规则的配置。

示例代码如下：

```
using System;
using System.Collections.Generic;
using Tea;
namespace Alibabacloud.Sample
{
    public class Client
    {
        /**
         * Create client初始化公共请求参数。
         */
        public static AlibabaCloud.SDK.EventBridge.EventBridgeClient CreateClient()
        {
            AlibabaCloud.SDK.EventBridge.Models.Config config = new AlibabaCloud.SDK.EventBridge.Models.Config();
            // 您的AccessKey ID。
            config.AccessKeyId = "<accessKeyId>";
            // 您的AccessKey Secret。
            config.AccessKeySecret = "<accessKeySecret>";
            // 您的接入点。
            config.Endpoint = "<endpoint>";
            return new AlibabaCloud.SDK.EventBridge.EventBridgeClient(config);
        }
        public static void UpdateEventRuleSample(AlibabaCloud.SDK.EventBridge.EventBridgeClient client)
        {
            try
            {
                AlibabaCloud.SDK.EventBridge.Models.UpdateRuleRequest updateEventRuleRequest = new AlibabaCloud.SDK.EventBridge.Models.UpdateRuleRequest();
                updateEventRuleRequest.EventBusName = "demo-bus";
                updateEventRuleRequest.RuleName = "myRule";
                updateEventRuleRequest.FilterPattern = "{\"source\":\"acs.oss\"},\"type\":\"oss:BucketQueried:GetBucketStat\"}";
                client.UpdateRule(updateEventRuleRequest);
                Console.WriteLine("Update rule success.");
            }
            catch (Exception e)
            {
                Console.WriteLine(e.ToString());
            }
        }
    }
}
```

```
        Console.WriteLine("-----update rule success-----");
    }
    catch (TeaException error)
    {
        Console.WriteLine(error.Message);
    }
    catch (Exception _error)
    {
        TeaException error = new TeaException(new Dictionary<string, object>
        { { "message", _error.Message }
        });
        Console.WriteLine(error.Message);
    }
}
static void Main(string[] args)
{
    AlibabaCloud.SDK.EventBridge.EventBridgeClient client = Client.CreateClient();
    Client.UpdateEventRuleSample(client);
    Console.ReadKey();
}
}
```

### 1.6.6.7. ListRules

本文介绍如何使用C# SDK查询一个事件总线下的所有事件规则。

示例代码如下：

```
using System;
using System.Collections.Generic;
using Tea;
namespace Alibabacloud.Sample
{
    public class Client
    {
        /**
         * Create client初始化公共请求参数。
         */
        public static AlibabaCloud.SDK.EventBridge.EventBridgeClient CreateClient()
        {
            AlibabaCloud.SDK.EventBridge.Models.Config config = new AlibabaCloud.SDK.EventBridge.Models.Con
```

```
fig());
    // 您的AccessKey ID。
    config.AccessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.AccessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.Endpoint = "<endpoint>";
    return new AlibabaCloud.SDK.EventBridge.EventBridgeClient(config);
}

public static void ListEventRuleSample(AlibabaCloud.SDK.EventBridge.EventBridgeClient client)
{
    try
    {
        AlibabaCloud.SDK.EventBridge.Models.ListRulesRequest listEventRulesRequest = new AlibabaCloud.
        SDK.EventBridge.Models.ListRulesRequest();
        listEventRulesRequest.EventBusName = "demo-bus";
        AlibabaCloud.SDK.EventBridge.Models.ListRulesResponse response = client.ListRules(listEventRules
        Request);
        Console.WriteLine("-----list rule success-----");
    }
    catch (TeaException error)
    {
        Console.WriteLine(error.Message);
    }
    catch (Exception _error)
    {
        TeaException error = new TeaException(new Dictionary<string, object>
        { { "message", _error.Message }
        });
        Console.WriteLine(error.Message);
    }
}

static void Main(string[] args)
{
    AlibabaCloud.SDK.EventBridge.EventBridgeClient client = Client.CreateClient();
    Client.ListEventRuleSample(client);
    Console.ReadKey();
}
}
```

## 1.6.6.8. DeleteRule

本文介绍如何使用C# SDK删除一个事件规则。

示例代码如下：

```
using System;
using System.Collections.Generic;
using Tea;
namespace Alibabacloud.Sample
{
    public class Client
    {
        /**
         * Create client初始化公共请求参数。
         */
        public static AlibabaCloud.SDK.EventBridge.EventBridgeClient CreateClient()
        {
            AlibabaCloud.SDK.EventBridge.Models.Config config = new AlibabaCloud.SDK.EventBridge.Models.Config();
            // 您的AccessKey ID。
            config.AccessKeyId = "<accessKeyId>";
            // 您的AccessKey Secret。
            config.AccessKeySecret = "<accessKeySecret>";
            // 您的接入点。
            config.Endpoint = "<endpoint>";
            return new AlibabaCloud.SDK.EventBridge.EventBridgeClient(config);
        }
        public static void DeleteEventRuleSample(AlibabaCloud.SDK.EventBridge.EventBridgeClient client)
        {
            try
            {
                AlibabaCloud.SDK.EventBridge.Models.DeleteRuleRequest deleteRuleRequest = new AlibabaCloud.SDK.EventBridge.Models.DeleteRuleRequest();
                deleteRuleRequest.RuleName = "myRule";
                deleteRuleRequest.EventBusName = "demo-bus";
                client.DeleteRule(deleteRuleRequest);
                Console.WriteLine("-----delete rule success-----");
            }
            catch (TeaException error)
            {
                Console.WriteLine(error.Message);
            }
        }
    }
}
```

```
    }
    catch (Exception _error)
    {
        TeaException error = new TeaException(new Dictionary<string, object>
        { { "message", _error.Message }
        });
        Console.WriteLine(error.Message);
    }
}

static void Main(string[] args)
{
    AlibabaCloud.SDK.EventBridge.EventBridgeClient client = Client.CreateClient();
    Client.DeleteEventRuleSample(client);
    Console.ReadKey();
}
}
```

## 1.6.7. 事件

### 1.6.7.1. 事件概述

您可以通过C# SDK调用事件接口发布事件。

C# SDK提供以下事件接口：

- [PutEvents](#)

### 1.6.7.2. PutEvents

本文介绍如何使用C# SDK发布一个或多个事件。

示例代码如下：

```
using System;
using System.Collections.Generic;
using Tea;
namespace Alibabacloud.Sample
{
    public class Client
    {
        /**
         * Create client初始化公共请求参数。
         */
    }
}
```

```

    */
    public static AlibabaCloud.SDK.EventBridge.EventBridgeClient CreateClient()
    {
        AlibabaCloud.SDK.EventBridge.Models.Config config = new AlibabaCloud.SDK.EventBridge.Models.Config();
        // 您的AccessKey ID。
        config.AccessKeyId = "<accessKeyId>";
        // 您的AccessKey Secret。
        config.AccessKeySecret = "<accessKeySecret>";
        // 您的接入点。
        config.Endpoint = "<endpoint>";
        return new AlibabaCloud.SDK.EventBridge.EventBridgeClient(config);
    }
    /**
     * PutEvents
     */
    public static void PutEvents(AlibabaCloud.SDK.EventBridge.EventBridgeClient client)
    {
        AlibabaCloud.SDK.EventBridge.Models.CloudEvent event_ = new AlibabaCloud.SDK.EventBridge.Models.CloudEvent();
        event_.Datacontenttype = "application/json";
        event_.Data = AlibabaCloud.TeaUtil.Common.ToBytes("test");
        event_.Id = "a5074581-7e74-4e4c-868f-47e7afdf****";
        event_.Source = "acs.oss";
        event_.Specversion = "1.0";
        event_.Type = "oss:ObjectCreated:PostObject";
        event_.Time = "2020-08-24T13:54:05.965Asia/Shanghai";
        event_.Subject = "1.0";
        event_.Type = "acs:oss:cn-hangzhou:1234567:xls-papk/game_apk/123.jpg";
        event_.Extensions = new Dictionary<string, object>
        {
            { "aliyuneventbusname", "demo-bus" },
        };
        try
        {
            AlibabaCloud.SDK.EventBridge.Models.PutEventsResponse resp = client.PutEvents(new List<AlibabaCloud.SDK.EventBridge.Models.CloudEvent>
            {
                event_
            });
            Console.WriteLine("-----Publish event to the aliyun EventBus-----");
            Console.WriteLine(AlibabaCloud.TeaUtil.Common.ToJSONString(resp.ToMap()));
        }
    }
}

```

```
}
catch (TeaException error)
{
    Console.WriteLine(error.Message);
}
catch (Exception _error)
{
    TeaException error = new TeaException(new Dictionary<string, object>
    { { "message", _error.Message }
    });
    Console.WriteLine(error.Message);
}
}
static void Main(string[] args)
{
    AlibabaCloud.SDK.EventBridge.EventBridgeClient client = Client.CreateClient();
    Client.PutEvents(client);
    Console.ReadKey();
}
}
```

## 1.6.8. 事件目标

### 1.6.8.1. 事件目标概述

您可以通过C# SDK调用事件目标接口创建事件目标、删除事件目标和查询事件目标。

C# SDK提供以下事件目标接口：

- [CreateTargets](#)
- [ListTargets](#)
- [DeleteTargets](#)

### 1.6.8.2. CreateTargets

本文介绍如何使用C# SDK在一个事件规则下创建一个或多个投递目标。

示例代码如下：

```
using System;
using System.Collections.Generic;
using Tea;
namespace test
```

```
{
    public class Client
    {
        /**
         * Create client初始化公共请求参数。
         */
        public static AlibabaCloud.SDK.EventBridge.EventBridgeClient CreateClient()
        {
            AlibabaCloud.SDK.EventBridge.Models.Config config = new AlibabaCloud.SDK.EventBridge.Models.Config();
            // 您的AccessKey ID。
            config.AccessKeyId = "<accessKeyId>";
            // 您的AccessKey Secret。
            config.AccessKeySecret = "<accessKeySecret>";
            // 您的接入点。
            config.Endpoint = "<endpoint>";
            return new AlibabaCloud.SDK.EventBridge.EventBridgeClient(config);
        }

        public static void CreateTargetsSample(AlibabaCloud.SDK.EventBridge.EventBridgeClient client)
        {
            try
            {
                AlibabaCloud.SDK.EventBridge.Models.CreateTargetsRequest createTargetsRequest = new AlibabaCloud.SDK.EventBridge.Models.CreateTargetsRequest();
                createTargetsRequest.EventBusName = "demo-bus";
                createTargetsRequest.RuleName = "myRule";
                AlibabaCloud.SDK.EventBridge.Models.TargetEntry targetEntry = new AlibabaCloud.SDK.EventBridge.Models.TargetEntry();
                targetEntry.Id = "1234";
                targetEntry.Endpoint = "http://www.baidu.com";
                List<AlibabaCloud.SDK.EventBridge.Models.TargetEntry> list = new List<AlibabaCloud.SDK.EventBridge.Models.TargetEntry>
                {
                    targetEntry
                };
                createTargetsRequest.Targets = list;
                AlibabaCloud.SDK.EventBridge.Models.CreateTargetsResponse response = client.CreateTargets(createTargetsRequest);
                Console.WriteLine("-----Create targets success-----");
                Console.WriteLine(AlibabaCloud.TeaUtil.Common.ToJSONString(response.ToMap()));
            }
            catch (Exception e)
            {
                Console.WriteLine(e.ToString());
            }
        }
    }
}
```

```
    }
    catch (TeaException error)
    {
        Console.WriteLine(error.Message);
    }
    catch (Exception _error)
    {
        TeaException error = new TeaException(new Dictionary<string, object>
        { { "message", _error.Message }
        });
        Console.WriteLine(error.Message);
    }
}

static void Main(string[] args)
{
    AlibabaCloud.SDK.EventBridge.EventBridgeClient client = CreateClient();
    CreateTargetsSample(client);
    Console.ReadKey();
}
}
```

### 1.6.8.3. ListTargets

本文介绍如何使用C# SDK查询一个事件规则下的所有投递目标。

示例代码如下：

```
using System;
using System.Collections.Generic;
using Tea;
namespace Alibabacloud.Sample
{
    public class Client
    {
        /**
         * Create client初始化公共请求参数。
         */
        public static AlibabaCloud.SDK.EventBridge.EventBridgeClient CreateClient()
        {
            AlibabaCloud.SDK.EventBridge.Models.Config config = new AlibabaCloud.SDK.EventBridge.Models.Con
            fig();
```

```
// 您的AccessKey ID。
config.AccessKeyId = "<accessKeyId>";
// 您的AccessKey Secret。
config.AccessKeySecret = "<accessKeySecret>";
// 您的接入点。
config.Endpoint = "<endpoint>";
return new AlibabaCloud.SDK.EventBridge.EventBridgeClient(config);
}

public static void ListTargetsSample(AlibabaCloud.SDK.EventBridge.EventBridgeClient client)
{
    try
    {
        AlibabaCloud.SDK.EventBridge.Models.ListTargetsRequest listTargetsRequest = new AlibabaCloud.S
DK.EventBridge.Models.ListTargetsRequest();
        listTargetsRequest.EventBusName = "demo-bus";
        listTargetsRequest.RuleName = "myRule";
        AlibabaCloud.SDK.EventBridge.Models.ListTargetsResponse response = client.ListTargets(listTarget
sRequest);
        Console.WriteLine("-----list targets success-----");
        Console.WriteLine(AlibabaCloud.TeaUtil.Common.ToJSONString(response.ToMap()));
    }
    catch (TeaException error)
    {
        Console.WriteLine(error.Message);
    }
    catch (Exception _error)
    {
        TeaException error = new TeaException(new Dictionary<string, object>
        { { "message", _error.Message }
        });
        Console.WriteLine(error.Message);
    }
}

static void Main(string[] args)
{
    AlibabaCloud.SDK.EventBridge.EventBridgeClient client = Client.CreateClient();
    Client.ListTargetsSample(client);
    Console.ReadKey();
}
}
```

## 1.6.8.4. DeleteTargets

本文介绍如何使用C# SDK删除一个事件规则下的一个或多个投递目标。

示例代码如下：

```
using System;
using System.Collections.Generic;
using Tea;
namespace AlibabaCloud.Sample
{
    public class Client
    {
        /**
         * Create client初始化公共请求参数。
         */
        public static AlibabaCloud.SDK.EventBridge.EventBridgeClient CreateClient()
        {
            AlibabaCloud.SDK.EventBridge.Models.Config config = new AlibabaCloud.SDK.EventBridge.Models.Config();
            // 您的AccessKey ID。
            config.AccessKeyId = "<accessKeyId>";
            // 您的AccessKey Secret。
            config.AccessKeySecret = "<accessKeySecret>";
            // 您的接入点。
            config.Endpoint = "<endpoint>";
            return new AlibabaCloud.SDK.EventBridge.EventBridgeClient(config);
        }
        public static void DeleteTargetsSample(AlibabaCloud.SDK.EventBridge.EventBridgeClient client)
        {
            try
            {
                AlibabaCloud.SDK.EventBridge.Models.DeleteTargetsRequest deleteTargetsRequest = new AlibabaCloud.SDK.EventBridge.Models.DeleteTargetsRequest();
                deleteTargetsRequest.EventBusName = "demo-bus";
                deleteTargetsRequest.RuleName = "myRule";
                List<string> list = new List<string>
                {
                    "1234"
                };
            }
        }
    }
}
```

```
deleteTargetsResponse.TargetIds = list;
AlibabaCloud.SDK.EventBridge.Models.DeleteTargetsResponse response = client.DeleteTargets(deleteTargetsResponse);
Console.WriteLine("-----delete targets success-----");
Console.WriteLine(AlibabaCloud.TeaUtil.Common.ToJSONString(response.ToMap()));
}
catch (TeaException error)
{
    Console.WriteLine(error.Message);
}
catch (Exception _error)
{
    TeaException error = new TeaException(new Dictionary<string, object>
    { { "message", _error.Message }
    });
    Console.WriteLine(error.Message);
}
}
static void Main(string[] args)
{
    AlibabaCloud.SDK.EventBridge.EventBridgeClient client = Client.CreateClient();
    Client.DeleteTargetsSample(client);
    Console.ReadKey();
}
}
```

# 1.7. TypeScript SDK

## 1.7.1. 概述

本文介绍事件总线Event Bridge TypeScript SDK中包含的内容。

### 内容介绍

事件总线Event Bridge TypeScript SDK提供下表中丰富的文档内容，您在实际操作过程中可以参见这些内容。

| 文档                               | 说明                       |
|----------------------------------|--------------------------|
| <a href="#">版本说明</a>             | 介绍TypeScript SDK的版本更新记录。 |
| <a href="#">安装TypeScript SDK</a> | 介绍如何安装TypeScript SDK。    |

| 文档                   | 说明                            |
|----------------------|-------------------------------|
| <a href="#">快速入门</a> | 介绍如何快速使用TypeScript SDK完成常见操作。 |
| <a href="#">事件总线</a> | 介绍如何使用TypeScript SDK管理事件总线。   |
| <a href="#">事件规则</a> | 介绍如何快速使用TypeScript SDK管理事件规则。 |
| <a href="#">事件</a>   | 介绍如何快速使用TypeScript SDK发布事件。   |
| <a href="#">事件目标</a> | 介绍如何快速使用TypeScript SDK管理事件目标。 |

## 1.7.2. 版本说明

本文介绍TypeScript SDK的版本信息，包含下载链接、发布时间、更新说明等，以便您按需获取适用的TypeScript SDK。

### Version 1.2.x

| 发布时间       | 下载                                                | 更新说明                                                                                                                                           |
|------------|---------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| 2020-12-18 | <a href="#">SDK下载</a><br><a href="#">sample下载</a> | <ul style="list-style-type: none"> <li>Event格式兼容CloudEvents生态标准。</li> <li>签名算法采用Aliyun Cloud Service通用签名算法。</li> <li>客户端代码采用自动生成方式。</li> </ul> |

## 1.7.3. 安装TypeScript SDK

本文介绍如何安装事件总线EventBridge TypeScript SDK。

### 环境准备

1. 安装Node.js 8.0或以上的版本。更多信息，请参见[安装Node.js](#)。

执行命令 `node -v` 查看Node.js版本。

2. 全局安装Typescript。

```
npm install -g typescript
```

### 安装SDK

1. 执行以下命令安装TypeScript SDK。

```
npm install @alicloud/eventbridge -S
```

2. 执行以下命令安装打印端口。

```
npm install @alicloud/tea-console -S
```

## 1.7.4. 快速入门

本文以发布事件到事件总线EventBridge为例介绍如何快速使用TypeScript SDK调用API。

### 前提条件

- [开通事件总线EventBridge服务](#)
- [创建AccessKey](#)
- [安装TypeScript SDK](#)

### 发布事件

示例代码如下：

```
import EventBridge, * as $EventBridge from '@alicloud/eventbridge';
import Util from '@alicloud/tea-util';
import Console from '@alicloud/tea-console';
import * as $tea from '@alicloud/tea-typescript';
export default class Client {
  /**
   * Create client初始化公共请求参数。
   */
  static async createClient(): Promise<EventBridge> {
    let config = new $EventBridge.Config({});
    // 您的AccessKey ID。
    config.accessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.accessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.endpoint = "<endpoint>";
    return new EventBridge(config);
  }
  /**
   * PutEvents
   */
  static async PutEvents(client: EventBridge): Promise<void> {
    let event = new $EventBridge.CloudEvent({});
    event.datacontenttype = "application/json";
    event.data = Util.toBytes("test");
    event.id = "a5074581-7e74-4e4c-868f-47e7afdf8445";
    event.source = "acs.oss";
    event.specversion = "1.0";
    event.type = "oss:ObjectCreated:PostObject";
    event.time = "2020-08-24T13:54:05.965Asia/Shanghai";
```

```
event.subject = "1.0";
event.type = "acs:oss:cn-hangzhou:1234567:xls-papk/game_apk/123.jpg";
event.extensions = {
    aliyuneventbusname: "demo-bus",
};
try {
    let resp = await client.putEvents([
        event
    ]);
    Console.log("-----Publish event to the aliyun EventBus-----");
    Console.log(Util.toJSONString($tea.toMap(resp)));
} catch (error) {
    Console.log(error.message);
}
}
static async main(args: string[]): Promise<void> {
    let client = await Client.createClient();
    await Client.PutEvents(client);
}
}
Client.main(process.argv.slice(2));
```

## 1.7.5. 事件总线

### 1.7.5.1. 事件总线概述

您可以通过TypeScript SDK创建总线、删除总线、获取总线详情和查询事件总线。

TypeScript SDK提供以下事件总线接口：

- [CreateEventBus](#)
- [GetEventBus](#)
- [DeleteEventBus](#)
- [ListEventBuses](#)

### 1.7.5.2. CreateEventBus

本文介绍如何使用TypeScript SDK创建一个事件总线。

示例代码如下：

```
import EventBridge, * as $EventBridge from '@alicloud/eventbridge';
import Util from '@alicloud/tea-util';
import Console from '@alicloud/tea-console';
import * as $tea from '@alicloud/tea-typescript';
export default class Client {
  /**
   * Create client初始化公共请求参数。
   */
  static async createClient(): Promise<EventBridge> {
    let config = new $EventBridge.Config({});
    // 您的AccessKey ID。
    config.accessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.accessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.endpoint = "<endpoint>";
    return new EventBridge(config);
  }
  static async createEventBusSample(client: EventBridge): Promise<void> {
    try {
      let createEventBusRequest = new $EventBridge.CreateEventBusRequest({});
      createEventBusRequest.eventBusName = "demo-bus";
      let resp = await client.createEventBus(createEventBusRequest);
      Console.log("-----Create bus success -----");
      Console.log(Util.toJSONString($tea.toMap(resp)));
    } catch (error) {
      Console.log(error.message);
    }
  }
  static async main(args: string[]): Promise<void> {
    let client = await Client.createClient();
    await Client.createEventBusSample(client);
  }
}
Client.main(process.argv.slice(2));
```

### 1.7.5.3. GetEventBus

本文介绍如何使用TypeScript SDK获取一个事件总线的详细信息。

示例代码如下：

```
import EventBridge, * as $EventBridge from '@alicloud/eventbridge';
import Util from '@alicloud/tea-util';
import Console from '@alicloud/tea-console';
import * as $tea from '@alicloud/tea-typescript';
export default class Client {
  /**
   * Create client初始化公共请求参数。
   */
  static async createClient(): Promise<EventBridge> {
    let config = new $EventBridge.Config({});
    // 您的AccessKey ID。
    config.accessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.accessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.endpoint = "<endpoint>";
    return new EventBridge(config);
  }
  static async getEventBusSample(client: EventBridge): Promise<void> {
    try {
      let getEventBusRequest = new $EventBridge.GetEventBusRequest({});
      getEventBusRequest.eventBusName = "demo-bus";
      let resp = await client.getEventBus(getEventBusRequest);
      Console.log("-----get event bus success -----");
      Console.log(Util.toJSONString($tea.toMap(resp)));
    } catch (error) {
      Console.log(error.message);
    }
  }
  static async main(args: string[]): Promise<void> {
    let client = await Client.createClient();
    await Client.getEventBusSample(client);
  }
}
Client.main(process.argv.slice(2));
```

## 1.7.5.4. DeleteEventBus

本文介绍如何使用TypeScript SDK删除一个事件总线。

示例代码如下：

```
import EventBridge, * as $EventBridge from '@alicloud/eventbridge';
import Util from '@alicloud/tea-util';
import Console from '@alicloud/tea-console';
import * as $tea from '@alicloud/tea-typescript';
export default class Client {
  /**
   * Create client初始化公共请求参数。
   */
  static async createClient(): Promise<EventBridge> {
    let config = new $EventBridge.Config({});
    // 您的AccessKey ID。
    config.accessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.accessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.endpoint = "<endpoint>";
    return new EventBridge(config);
  }
  static async deleteEventBusSample(client: EventBridge): Promise<void> {
    try {
      let deleteEventBusRequest = new $EventBridge.DeleteEventBusRequest({});
      deleteEventBusRequest.eventBusName = "demo-bus";
      let resp = await client.deleteEventBus(deleteEventBusRequest);
      Console.log("-----Delete bus success -----");
      Console.log(Util.toJSONString($tea.toMap(resp)));
    } catch (error) {
      Console.log(error.message);
    }
  }
  static async main(args: string[]): Promise<void> {
    let client = await Client.createClient();
    await Client.deleteEventBusSample(client);
  }
}
Client.main(process.argv.slice(2));
```

## 1.7.5.5. ListEventBuses

本文介绍如何使用TypeScript SDK查询所有事件总线。

示例代码如下：

```
import EventBridge, * as $EventBridge from '@alicloud/eventbridge';
import Util from '@alicloud/tea-util';
import Console from '@alicloud/tea-console';
import * as $tea from '@alicloud/tea-typescript';
export default class Client {
  /**
   * Create client初始化公共请求参数。
   */
  static async createClient(): Promise<EventBridge> {
    let config = new $EventBridge.Config({});
    // 您的AccessKey ID。
    config.accessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.accessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.endpoint = "<endpoint>";
    return new EventBridge(config);
  }
  static async listEventBusesSample(client: EventBridge): Promise<void> {
    try {
      let listEventBusesRequest = new $EventBridge.ListEventBusesRequest({});
      listEventBusesRequest.limit = 100;
      let resp = await client.listEventBuses(listEventBusesRequest);
      Console.log("-----list bus success -----");
      Console.log(Util.toJSONString($tea.toMap(resp)));
    } catch (error) {
      Console.log(error.message);
    }
  }
  static async main(args: string[]): Promise<void> {
    let client = await Client.createClient();
    await Client.listEventBusesSample(client);
  }
}
Client.main(process.argv.slice(2));
```

## 1.7.6. 事件规则

### 1.7.6.1. 事件规则概述

您可以通过TypeScript SDK创建规则、禁用规则、启用规则、删除规则等。

TypeScript SDK提供以下事件规则接口：

- [CreateRule](#)
- [GetRule](#)
- [EnableRule](#)
- [DisableRule](#)
- [UpdateRule](#)
- [ListRules](#)
- [DeleteRule](#)

### 1.7.6.2. CreateRule

本文介绍如何使用TypeScript SDK创建一个事件规则。

示例代码如下：

```
import EventBridge, * as $EventBridge from '@alicloud/eventbridge';
import Util from '@alicloud/tea-util';
import Console from '@alicloud/tea-console';
import * as $tea from '@alicloud/tea-typescript';
export default class Client {
  /**
   * Create client初始化公共请求参数。
   */
  static async createClient(): Promise<EventBridge> {
    let config = new $EventBridge.Config({});
    // 您的AccessKey ID。
    config.accessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.accessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.endpoint = "<endpoint>";
    return new EventBridge(config);
  }
  static async createEventRuleSample(client: EventBridge): Promise<void> {
    try {
      let createEventRuleRequest = new $EventBridge.CreateRuleRequest({});
      let targetEntrv = new $EventBridge.TargetEntrv({});
```

```

        targetEntry.id = "dingtalk.target";
        targetEntry.endpoint = "https://oapi.dingtalk.com/robot/send?access_token=1019d4a19e2ef6b2f*****
*****396fc5e94814ed8460";
        targetEntry.type = "acs.dingtalk";
        let param1 = new $EventBridge.EBTargetParam({ });
        param1.resourceKey = "URL";
        param1.form = "CONSTANT";
        param1.value = "https://oapi.dingtalk.com/robot/send?access_token=1019d4a19e2ef6b2f*****3
96fc5e94814ed8460";
        let param2 = new $EventBridge.EBTargetParam({ });
        param2.resourceKey = "SecretKey";
        param2.form = "CONSTANT";
        param2.value = "SEC121a71ff304a65b4f7c*****1f4d9f6c1ca514300d15234";
        let param3 = new $EventBridge.EBTargetParam({ });
        param3.resourceKey = "Body";
        param3.form = "TEMPLATE";
        param3.value = `{n  \"key\": \"${$.source}\",n  \"value\": \"${$.data}\"n}`;
        param3.template = `{"msgtype": "text", "text": {"content": "Hello: ${key}"}}`;
        let paramList = [
            param1,
            param2,
            param3
        ];
        targetEntry.paramList = paramList;
        let targetEntryList = [
            targetEntry
        ];
        createEventRuleRequest.targets = targetEntryList;
        createEventRuleRequest.ruleName = "myRule";
        createEventRuleRequest.eventBusName = "demo-bus";
        createEventRuleRequest.filterPattern = `{"source":["acs.oss"],"type":["oss:ObjectCreated:Uploa
dPart"]}`;
        createEventRuleRequest.status = "enable";
        let resp = await client.createRule(createEventRuleRequest);
        Console.log("-----create rule success-----");
        Console.log(Util.toJSONString($tea.toMap(resp)));
    } catch (error) {
        Console.log(error.message);
    }
}

```

```
static async main(args: string[]): Promise<void> {  
    let client = await Client.createClient();  
    await Client.createEventRuleSample(client);  
}  
}  
Client.main(process.argv.slice(2));
```

### 1.7.6.3. GetRule

本文介绍如何使用TypeScript SDK获取一个事件规则的详细信息。

示例代码如下：

```
import EventBridge, * as $EventBridge from '@alicloud/eventbridge';
import Util from '@alicloud/tea-util';
import Console from '@alicloud/tea-console';
import * as $tea from '@alicloud/tea-typescript';
export default class Client {
  /**
   * Create client初始化公共请求参数。
   */
  static async createClient(): Promise<EventBridge> {
    let config = new $EventBridge.Config({});
    // 您的AccessKey ID。
    config.accessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.accessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.endpoint = "<endpoint>";
    return new EventBridge(config);
  }
  static async getEventRuleSample(client: EventBridge): Promise<void> {
    try {
      let getEventRuleRequest = new $EventBridge.GetRuleRequest({});
      getEventRuleRequest.ruleName = "myRule";
      getEventRuleRequest.eventBusName = "demo-bus";
      let resp = await client.getRule(getEventRuleRequest);
      Console.log("-----get rule success-----");
      Console.log(Util.toJSONString($tea.toMap(resp)));
    } catch (error) {
      Console.log(error.message);
    }
  }
  static async main(args: string[]): Promise<void> {
    let client = await Client.createClient();
    await Client.getEventRuleSample(client);
  }
}
Client.main(process.argv.slice(2));
```

## 1.7.6.4. EnableRule

本文介绍如何使用TypeScript SDK启用一个事件规则。

示例代码如下：

```
import EventBridge, * as $EventBridge from '@alicloud/eventbridge';
import Util from '@alicloud/tea-util';
import Console from '@alicloud/tea-console';
import * as $tea from '@alicloud/tea-typescript';
export default class Client {
  /**
   * Create client初始化公共请求参数。
   */
  static async createClient(): Promise<EventBridge> {
    let config = new $EventBridge.Config({});
    // 您的AccessKey ID。
    config.accessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.accessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.endpoint = "<endpoint>";
    return new EventBridge(config);
  }
  static async enableEventRuleSample(client: EventBridge): Promise<void> {
    try {
      let enableEventRuleRequest = new $EventBridge.EnableRuleRequest({});
      enableEventRuleRequest.ruleName = "myRule";
      enableEventRuleRequest.eventBusName = "demo-bus";
      let resp = await client.enableRule(enableEventRuleRequest);
      Console.log("-----enable rule success-----");
      Console.log(Util.toJSONString($tea.toMap(resp)));
    } catch (error) {
      Console.log(error.message);
    }
  }
  static async main(args: string[]): Promise<void> {
    let client = await Client.createClient();
    await Client.enableEventRuleSample(client);
  }
}
Client.main(process.argv.slice(2));
```

## 1.7.6.5. DisableRule

本文介绍如何使用TypeScript SDK禁用一个事件规则。

示例代码如下：

```
import EventBridge, * as $EventBridge from '@alicloud/eventbridge';
import Util from '@alicloud/tea-util';
import Console from '@alicloud/tea-console';
import * as $tea from '@alicloud/tea-typescript';
export default class Client {
  /**
   * Create client初始化公共请求参数。
   */
  static async createClient(): Promise<EventBridge> {
    let config = new $EventBridge.Config({});
    // 您的AccessKey ID。
    config.accessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.accessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.endpoint = "<endpoint>";
    return new EventBridge(config);
  }
  static async disableEventRuleSample(client: EventBridge): Promise<void> {
    try {
      let disableEventRuleRequest = new $EventBridge.DisableRuleRequest({});
      disableEventRuleRequest.ruleName = "myRule";
      disableEventRuleRequest.eventBusName = "demo-bus";
      let resp = await client.disableRule(disableEventRuleRequest);
      Console.log("-----disable rule success-----");
      Console.log(Util.toJSONString($tea.toMap(resp)));
    } catch (error) {
      Console.log(error.message);
    }
  }
  static async main(args: string[]): Promise<void> {
    let client = await Client.createClient();
    await Client.disableEventRuleSample(client);
  }
}
Client.main(process.argv.slice(2));
```

## 1.7.6.6. UpdateRule

本文介绍如何使用TypeScript SDK更新一个事件规则的配置。

示例代码如下：

```

import EventBridge, * as $EventBridge from '@alicloud/eventbridge';
import Util from '@alicloud/tea-util';
import Console from '@alicloud/tea-console';
import * as $tea from '@alicloud/tea-typescript';
export default class Client {
  /**
   * Create client初始化公共请求参数。
   */
  static async createClient(): Promise<EventBridge> {
    let config = new $EventBridge.Config({});
    // 您的AccessKey ID。
    config.accessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.accessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.endpoint = "<endpoint>";
    return new EventBridge(config);
  }
  static async updateEventRuleSample(client: EventBridge): Promise<void> {
    try {
      let updateEventRuleRequest = new $EventBridge.UpdateRuleRequest({});
      updateEventRuleRequest.eventBusName = "demo-bus";
      updateEventRuleRequest.ruleName = "myRule";
      updateEventRuleRequest.filterPattern = `{"source":["acs.oss"],"type":["oss:BucketQueried:GetB
ucketStat"]}`;
      let resp = await client.updateRule(updateEventRuleRequest);
      Console.log("-----update rule success-----");
      Console.log(Util.toJSONString($tea.toMap(resp)));
    } catch (error) {
      Console.log(error.message);
    }
  }
  static async main(args: string[]): Promise<void> {
    let client = await Client.createClient();
    await Client.updateEventRuleSample(client);
  }
}
Client.main(process.argv.slice(2));

```

## 1.7.6.7. ListRules

本文介绍如何使用TypeScript SDK查询一个事件总线下的所有事件规则。

示例代码如下：

```
import EventBridge, * as $EventBridge from '@alicloud/eventbridge';
import Util from '@alicloud/tea-util';
import Console from '@alicloud/tea-console';
import * as $tea from '@alicloud/tea-typescript';
export default class Client {
  /**
   * Create client初始化公共请求参数。
   */
  static async createClient(): Promise<EventBridge> {
    let config = new $EventBridge.Config({});
    // 您的AccessKey ID。
    config.accessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.accessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.endpoint = "<endpoint>";
    return new EventBridge(config);
  }
  static async listEventRulesSample(client: EventBridge): Promise<void> {
    try {
      let listEventRulesRequest = new $EventBridge.ListRulesRequest({});
      listEventRulesRequest.eventBusName = "demo-bus";
      let resp = await client.listRules(listEventRulesRequest);
      Console.log("-----list rule success-----");
      Console.log(Util.toJSONString($tea.toMap(resp)));
    } catch (error) {
      Console.log(error.message);
    }
  }
  static async main(args: string[]): Promise<void> {
    let client = await Client.createClient();
    await Client.listEventRulesSample(client);
  }
}
Client.main(process.argv.slice(2));
```

## 1.7.6.8. DeleteRule

本文介绍如何使用TypeScript SDK删除一个事件规则。

示例代码如下：

```
import EventBridge, * as $EventBridge from '@alicloud/eventbridge';
import Util from '@alicloud/tea-util';
import Console from '@alicloud/tea-console';
import * as $tea from '@alicloud/tea-typescript';
export default class Client {
  /**
   * Create client初始化公共请求参数。
   */
  static async createClient(): Promise<EventBridge> {
    let config = new $EventBridge.Config({});
    // 您的AccessKey ID。
    config.accessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.accessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.endpoint = "<endpoint>";
    return new EventBridge(config);
  }
  static async deleteEventRuleSample(client: EventBridge): Promise<void> {
    try {
      let deleteRuleRequest = new $EventBridge.DeleteRuleRequest({});
      deleteRuleRequest.ruleName = "myRule";
      deleteRuleRequest.eventBusName = "demo-bus";
      let resp = await client.deleteRule(deleteRuleRequest);
      Console.log("-----delete rule success-----");
      Console.log(Util.toJSONString($tea.toMap(resp)));
    } catch (error) {
      Console.log(error.message);
    }
  }
  static async main(args: string[]): Promise<void> {
    let client = await Client.createClient();
    await Client.deleteEventRuleSample(client);
  }
}
Client.main(process.argv.slice(2));
```

## 1.7.7. 事件

### 1.7.7.1. 事件概述

您可以通过TypeScript SDK调用事件接口发布事件。

TypeScript SDK提供以下事件接口：

- [PutEvents](#)

### 1.7.7.2. PutEvents

本文介绍如何使用TypeScript SDK发布一个或多个事件。

示例代码如下：

```
import EventBridge, * as $EventBridge from '@alicloud/eventbridge';
import Util from '@alicloud/tea-util';
import Console from '@alicloud/tea-console';
import * as $tea from '@alicloud/tea-typescript';
export default class Client {
  /**
   * Create client初始化公共请求参数。
   */
  static async createClient(): Promise<EventBridge> {
    let config = new $EventBridge.Config({});
    // 您的AccessKey ID。
    config.accessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.accessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.endpoint = "<endpoint>";
    return new EventBridge(config);
  }
  /**
   * PutEvents
   */
  static async PutEvents(client: EventBridge): Promise<void> {
    let event = new $EventBridge.CloudEvent({});
    event.datacontenttype = "application/json";
    event.data = Util.toBytes("test");
    event.id = "a5074581-7e74-4e4c-868f-47e7afdf8445";
    event.source = "acs.oss";
    event.specversion = "1.0";
    event.type = "oss:ObjectCreated:PostObject";
    event.time = "2020-08-24T13:54:05.965Asia/Shanghai";
```

```
event.subject = "1.0";
event.type = "acs:oss:cn-hangzhou:1234567:xls-papk/game_apk/123.jpg";
event.extensions = {
  aliyuneventbusname: "demo-bus",
};
try {
  let resp = await client.putEvents([
    event
  ]);
  Console.log("-----Publish event to the aliyun EventBus-----");
  Console.log(Util.toJSONString($tea.toMap(resp)));
} catch (error) {
  Console.log(error.message);
}
}
static async main(args: string[]): Promise<void> {
  let client = await Client.createClient();
  await Client.PutEvents(client);
}
}
Client.main(process.argv.slice(2));
```

## 1.7.8. 事件目标

### 1.7.8.1. 事件目标概述

您可以通过TypeScript SDK创建事件目标、删除事件目标和查询事件目标。

TypeScript SDK提供以下事件目标接口：

- [CreateTargets](#)
- [ListTargets](#)
- [DeleteTargets](#)

### 1.7.8.2. CreateTargets

本文介绍如何使用TypeScript SDK在一个事件规则下创建一个或多个投递目标。

示例代码如下：

```
import EventBridge, * as $EventBridge from '@alicloud/eventbridge';
import Util from '@alicloud/tea-util';
import Console from '@alicloud/tea-console';
import * as $tea from '@alicloud/tea-typescript';
```

```

export default class Client {
  /**
   * Create client初始化公共请求参数。
   */
  static async createClient(): Promise<EventBridge> {
    let config = new $EventBridge.Config({});
    // 您的AccessKey ID。
    config.accessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.accessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.endpoint = "<endpoint>";
    return new EventBridge(config);
  }

  static async createTargetsSample(client: EventBridge): Promise<void> {
    try {
      let createTargetsRequest = new $EventBridge.CreateTargetsRequest({});
      createTargetsRequest.eventBusName = "demo-bus";
      createTargetsRequest.ruleName = "myRule";
      let targetEntry = new $EventBridge.TargetEntry({});
      targetEntry.id = "dingtalk.target";
      targetEntry.endpoint = "https://oapi.dingtalk.com/robot/send?access_token=1019d4a19e2ef6b2f*****
*****396fc5e94814ed8460";
      targetEntry.type = "acs.dingtalk";
      let param1 = new $EventBridge.EBTargetParam({});
      param1.resourceKey = "URL";
      param1.form = "CONSTANT";
      param1.value = "https://oapi.dingtalk.com/robot/send?access_token=1019d4a19e2ef6b2f*****3
96fc5e94814ed8460";
      let param2 = new $EventBridge.EBTargetParam({});
      param2.resourceKey = "SecretKey";
      param2.form = "CONSTANT";
      param2.value = "SEC121a71ff304a65b4f7c*****1f4d9f6c1ca514300d15234";
      let param3 = new $EventBridge.EBTargetParam({});
      param3.resourceKey = "Body";
      param3.form = "TEMPLATE";
      param3.value = `{
  "key": "${$.source}",
  "value": "${$.data}"
}`;
      param3.template = `{"msgtype": "text", "text": {"content": "Hello: ${key}"}}`;
      let paramList = [
        param1,

```

```
        param2,
        param3
    ];
    targetEntry.paramList = paramList;
    let targetEntryList = [
        targetEntry
    ];
    createTargetsRequest.targets = targetEntryList;
    let response = await client.createTargets(createTargetsRequest);
    Console.log("-----Create targets success-----");
    Console.log(Util.toJSONString($tea.toMap(response)));
} catch (error) {
    Console.log(error.message);
}
}

static async main(args: string[]): Promise<void> {
    let client = await Client.createClient();
    await Client.createTargetsSample(client);
}
}

Client.main(process.argv.slice(2));
```

### 1.7.8.3. ListTargets

本文介绍如何使用TypeScript SDK查询一个事件规则下的所有投递目标。

示例代码如下：

```
import EventBridge, * as $EventBridge from '@alicloud/eventbridge';
import Util from '@alicloud/tea-util';
import Console from '@alicloud/tea-console';
import * as $tea from '@alicloud/tea-typescript';
export default class Client {
  /**
   * Create client初始化公共请求参数。
   */
  static async createClient(): Promise<EventBridge> {
    let config = new $EventBridge.Config({});
    // 您的AccessKey ID。
    config.accessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.accessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.endpoint = "<endpoint>";
    return new EventBridge(config);
  }
  static async listTargetsSample(client: EventBridge): Promise<void> {
    try {
      let listTargetsRequest = new $EventBridge.ListTargetsRequest({});
      listTargetsRequest.eventBusName = "demo-bus";
      listTargetsRequest.ruleName = "myRule";
      let response = await client.listTargets(listTargetsRequest);
      Console.log("-----list targets success-----");
      Console.log(Util.toJSONString($tea.toMap(response)));
    } catch (error) {
      Console.log(error.message);
    }
  }
  static async main(args: string[]): Promise<void> {
    let client = await Client.createClient();
    await Client.listTargetsSample(client);
  }
}
Client.main(process.argv.slice(2));
```

## 1.7.8.4. DeleteTargets

本文介绍如何使用TypeScript SDK删除一个事件规则下的一个或多个投递目标。

示例代码如下：

```
import EventBridge, * as $EventBridge from '@alicloud/eventbridge';
import Util from '@alicloud/tea-util';
import Console from '@alicloud/tea-console';
import * as $tea from '@alicloud/tea-typescript';
export default class Client {
  /**
   * Create client初始化公共请求参数。
   */
  static async createClient(): Promise<EventBridge> {
    let config = new $EventBridge.Config({});
    // 您的AccessKey ID。
    config.accessKeyId = "<accessKeyId>";
    // 您的AccessKey Secret。
    config.accessKeySecret = "<accessKeySecret>";
    // 您的接入点。
    config.endpoint = "<endpoint>";
    return new EventBridge(config);
  }
  static async removeTargetsSample(client: EventBridge): Promise<void> {
    try {
      let removeTargetsRequest = new $EventBridge.DeleteTargetsRequest({});
      removeTargetsRequest.eventBusName = "demo-bus";
      removeTargetsRequest.ruleName = "myRule";
      let list = [
        "dingtalk.target"
      ];
      removeTargetsRequest.targetIds = list;
      let response = await client.deleteTargets(removeTargetsRequest);
      Console.log("-----remove targets success-----");
      Console.log(Util.toJSONString($tea.toMap(response)));
    } catch (error) {
      Console.log(error.message);
    }
  }
  static async main(args: string[]): Promise<void> {
    let client = await Client.createClient();
    await Client.removeTargetsSample(client);
  }
}
Client.main(process.argv.slice(2));
```

## 2.CloudEvents Client SDK

您可使用开源CloudEvents Java SDK发布事件。

### 前提条件

1. 您已下载IntelliJ IDEA。

针对Java SDK，您可以使用IntelliJ IDEA或者Eclipse的IDE，本文以IntelliJ IDEA为例。IntelliJ IDEA的下载和安装详情请参见[官方网站](#)。

2. 在pom.xml中加入依赖。

```
<dependency>
  <groupId>io.cloudevents</groupId>
  <artifactId>cloudevents-core</artifactId>
  <version>${cloudevents.version}</version>
</dependency>
<dependency>
  <groupId>io.cloudevents</groupId>
  <artifactId>cloudevents-http-vertx</artifactId>
  <version>${cloudevents.version}</version>
</dependency>
<dependency>
  <groupId>io.cloudevents</groupId>
  <artifactId>cloudevents-api</artifactId>
  <version>${cloudevents.version}</version>
</dependency>
<dependency>
  <groupId>io.cloudevents</groupId>
  <artifactId>cloudevents-json-jackson</artifactId>
  <version>${cloudevents.version}</version>
</dependency>
```

 说明 *cloudevents.version*使用最新里程碑版本2.0.0-milestone1。

### 背景信息

- 如需使用控制台执行相关操作，请参见[控制台操作指南](#)。
- 本文提供的SDK中涉及的接口详情，请参见[API参考](#)。

### 发布事件

二进制示例代码如下：

```
import java.net.URI;
import java.util.UUID;
```

```
import io.cloudevents.CloudEvent;
import io.cloudevents.core.builder.CloudEventBuilder;
import io.cloudevents.http.vertx.VertxMessageFactory;
import io.vertx.core.Vertx;
import io.vertx.core.http.HttpClient;
import io.vertx.core.http.HttpClientRequest;
public class SampleBinaryHTTPClient{
    private static String accessKeyId = "your accessKeyId";
    private static String accessKeySecret = "your accessKeySecret";
    private static String endpoint = "http://endpoint/openapi/putEvents";
    public static void main(String[] args) throws Exception {
        final Vertx vertx = Vertx.vertx();
        final HttpClient httpClient = vertx.createHttpClient();
        // 创建一个事件模板来设置基本的CloudEvent属性。
        CloudEventBuilder eventTemplate = CloudEventBuilder.v1()
            .withSource(URI.create("https://github.com/cloudevents/sdk-java/tree/master/examples/vertx"))
            .withType("vertx.example");
        // 创建HTTP请求。
        final HttpClientRequest request = httpClient.postAbs(endpoint)
            .handler(response -> {
                System.out.println(response.statusMessage());
            })
            .exceptionHandler(System.err::println);
        String id = UUID.randomUUID()
            .toString();
        String data = "Event data";
        // 从模板中创建事件。
        final CloudEvent event = eventTemplate.newBuilder()
            .withId(id)
            .withData("application/json", data.getBytes())
            .withExtension("aliyuneventbusname", "jingluo-bus")
            .withSource(URI.create("https://github.com/cloudevents/sdk-java/tree/master/examples/vertx"))
            .withType("vertx.example")
            .withSubject("acs:oss:cn-hangzhou:1234567:xls-papk/game_apk/123.jpg")
            .build();
        request.putHeader("content-type", "application/json");
        request.putHeader("authorization",
            "acs" + ":" + accessKeyId + ":" + SignatureHelper.getSignature(SignatureHelper.getStringToSign(request),
                accessKeySecret) + "");
    }
}
```

```
    VertxMessageFactory.createWriter(request)
        .writeBinary(event);
    }
}
```

结构化示例代码如下：

```
import java.net.URI;
import java.util.UUID;
import io.cloudevents.CloudEvent;
import io.cloudevents.core.builder.CloudEventBuilder;
import io.cloudevents.http.vertx.VertxMessageFactory;
import io.cloudevents.jackson.JsonFormat;
import io.vertx.core.Vertx;
import io.vertx.core.http.HttpClient;
import io.vertx.core.http.HttpClientRequest;
public class SampleStructuredHTTPClient {
    private static String accessKeyId = "your accessKeyId";
    private static String accessKeySecret = "your accessKeySecret";
    private static String endpoint = "http://endpoint/openapi/putEvents";
    public static void main(String[] args) throws Exception {
        final Vertx vertx = Vertx.vertx();
        final HttpClient httpClient = vertx.createHttpClient();
        // 创建一个事件模板来设置基本的CloudEvent属性。
        CloudEventBuilder eventTemplate = CloudEventBuilder.v1()
            .withSource(URI.create("https://github.com/cloudevents/sdk-java/tree/master/examples/vertx"))
            .withType("vertx.example");
        // 创建HTTP请求。
        final HttpClientRequest request = httpClient.postAbs(endpoint)
            .handler(response -> {
                System.out.println(response.statusMessage());
            })
            .exceptionHandler(System.err::println);
        String id = UUID.randomUUID()
            .toString();
        String data = "Event data";
        // 从模板中创建事件。
        final CloudEvent event = eventTemplate.newBuilder()
            .withId(id)
            .withData("application/json", data.getBytes())
            .withExtension("aliyuneventbusname", "jingluo-bus")
            .build();
        request.send(event);
    }
}
```

```
.withSource(URI.create("https://github.com/cloudevents/sdk-java/tree/master/examples/vertx"))
.withType("vertx.example")
.withSubject("acs:oss:cn-hangzhou:1234567:xls-papk/game_apk/123.jpg")
.build();
request.putHeader("content-type", "application/cloudevents+json");
request.putHeader("authorization",
    "acs" + ":" + accessKeyId + ":" + SignatureHelper.getSignature(SignatureHelper.getStringToSign(request),
        accessKeySecret) + "");
VertxMessageFactory.createWriter(request)
    .writeStructured(event, new JsonFormat());
}
}
```

## 签名生成工具

示例代码如下：

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Base64;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.Set;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import io.netty.util.internal.StringUtil;
import io.vertx.core.MultiMap;
import io.vertx.core.http.HttpClientRequest;
public class SignatureHelper {
    public static String getStringToSign(HttpClientRequest request) {
        String method = request.method()
            .name();
        String pathname = request.path();
        MultiMap headers = request.headers();
        Map<String, String> query = buildQueryMap(request.query());
        String contentMD5 = headers.get("content-md5") == null ? "" : (String)headers.get("content-md5");
        String contentType = headers.get("content-type") == null ? "" : (String)headers.get("content-type");
        String date = headers.get("date") == null ? "null" : (String)headers.get("date");
        String header = method + "\n" + contentMD5 + "\n" + contentType + "\n" + date + "\n";
```

```
String canonicalizedHeaders = getCanonicalizedHeaders(headers);
String canonicalizedResource = getCanonicalizedResource(pathname, query);
String stringToSign = header + canonicalizedHeaders + canonicalizedResource;
return stringToSign;
}

private static Map<String, String> buildQueryMap(String query) {
    Map<String, String> map = new HashMap<>();
    if (!StringUtil.isNullOrEmpty(query)) {
        String[] params = query.split("&");
        Arrays.stream(params)
            .forEach(param -> {
                String[] kv = param.split("=");
                map.put(kv[0], kv[1]);
            });
    }
    return map;
}

protected static String getCanonicalizedHeaders(MultiMap headers) {
    String prefix = "x-eventbridge";
    Set<String> keys = headers.names();
    List<String> canonicalizedKeys = new ArrayList();
    Iterator var4 = keys.iterator();
    while (var4.hasNext()) {
        String key = (String)var4.next();
        if (key.startsWith(prefix)) {
            canonicalizedKeys.add(key);
        }
    }
    String[] canonicalizedKeysArray = (String[])canonicalizedKeys.toArray(new String[canonicalizedKeys.size()]);
    Arrays.sort(canonicalizedKeysArray);
    StringBuilder result = new StringBuilder();
    for (int i = 0; i < canonicalizedKeysArray.length; ++i) {
        String key = canonicalizedKeysArray[i];
        result.append(key);
        result.append(":");
        result.append(((String)headers.get(key)).trim());
        result.append("\n");
    }
    return result.toString();
}
```

```
,
protected static String getCanonicalizedResource(String pathname, Map<String, String> query) {
    String[] keys = (String[])query.keySet()
        .toArray(new String[query.size()]);
    if (keys.length <= 0) {
        return pathname;
    } else {
        Arrays.sort(keys);
        StringBuilder result = new StringBuilder(pathname);
        result.append("?");
        for (int i = 0; i < keys.length; ++i) {
            String key = keys[i];
            result.append(key);
            String value = (String)query.get(key);
            if (!StringUtil.isNullOrEmpty(value) && !"".equals(value.trim())) {
                result.append("=");
                result.append(value);
            }
            result.append("&");
        }
        return result.deleteCharAt(result.length() - 1)
            .toString();
    }
}

public static String getSignature(String stringToSign, String secret) throws Exception {
    Mac mac = Mac.getInstance("HmacSHA1");
    mac.init(new SecretKeySpec(secret.getBytes("UTF-8"), "HmacSHA1"));
    byte[] signData = mac.doFinal(stringToSign.getBytes("UTF-8"));
    return Base64.getEncoder()
        .encodeToString(signData);
}
}
```