

腾讯云无服务器云函数

管理云函数触发器

产品文档



腾讯云

【版权声明】

©2013-2017 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

文档声明.....	2
管理云函数触发器.....	4
什么是函数触发器	4
支持的触发器类型	6
COS 触发器	6
定时触发器	9
CMQ Topic 触发器	12
API网关触发器	14

管理云函数触发器

什么是函数触发器

腾讯云无服务器云函数是典型的事件触发（Event-Triggered）形态的无服务器运行环境，核心组件是 SCF 函数和事件源。其中，事件源是发布事件（Event）的腾讯云服务或用户自定义代码，SCF 函数是事件的处理者，而函数触发器就是管理函数和事件源对应关系的集合。例如以下场景：

- 图像/视频处理：用户上传图片时将图片切割成合适的尺寸。用户使用该应用上传照片，应用将这些用户照片存储到 COS 中并且创建每个用户照片的缩略图，并在用户页面上显示这些缩略图。本场景下，您需要选择 COS 作为事件源，在文件创建时将事件（Event）发布给 SCF 函数，事件数据提供关于存储桶和文件的所有信息。
= 数据处理：半夜12点，分析一天来收集的数据（比如clickstream）并生成报告。本场景下，您需要选择 定时器 作为事件源，在一个特定时间将事件（Event）发布给 SCF 函数。
- 自定义的应用程序：在您的某个应用程序中调用第一个图像处理 SCF 函数，作为应用程序的一个模块。本场景下，您需要该应用程序中自行调用 [InvokeFunction](#) 来发布事件（Event）。

这些事件源可以是以下任意之一：

- 内部事件源：这些是经过预配置可与 SCF 一起使用的腾讯云云服务。目前 SCF 支持 COS 和定时器两种内部事件源，当您配置了这些事件源触发函数时，函数将在出现事件时被自动调用。事件源和函数的关联关系（即事件源映射）将在事件源侧维护。例如，COS 提供 [Bucket通知配置 API](#)。使用此 API，您可以将 Bucket 事件和函数绑定起来
- 自定义应用程序：您可以让自定义应用程序发布事件和调用 SCF 函数。

示例 1：COS发布事件并调用函数

您可以配置 COS 的事件源映射，决定 COS 在发生何种行为时触发 SCF 函数（如PUT、DELETE对象等）。COS 的事件源映射存储在 COS 中，使用存储桶通知功能，引导 COS 在出现特定事件类型时调用函数：

- 创建 COS 触发器
- 用户在存储桶中创建/删除对象

- COS 检测到对象创建/删除事件。
- COS 自动调用函数，将根据存储在 COS 配置中的事件源映射明确应该调用哪个函数。将 Bucket 及 Object 信息作为事件数据传递给函数。

示例 2：定时器发布时间并调用函数

定时器的时间源映射将保存在 SCF 函数配置中，决定何时自动触发函数：

- 创建定时触发器
- 该定时器在配置时间时自动调用函数

示例 3：自定义应用程序调用函数

如果您需要在自定义应用程序中调用某个 SCF

函数，在这种情况下您不需要配置函数触发器，也不需设置事件源映射。此时，事件源使用 Invoke API。

- 自定义应用程序使用 InvokeFunction API 调用函数，自行传入事件数据。
- 函数接收到触发请求并执行
- 如果使用了同步调用方式，函数将向应用程序返回结果

注意：在此示例中，由于自定义应用程序和函数均为同一个用户生产的，可以指定用户凭证（appid、secretid 和secretkey）。

注意事项

1. 目前单个云函数仅支持 2 个 COS 触发器和 2 个 定时触发器
2. 由于不同云服务的限制，事件源映射关系有着特定的限制。例如：对于 COS 触发器而言，同一个 COS Buckt 的相同事件（例如文件上传），不能触发多个不同的函数。

支持的触发器类型

COS 触发器

用户可以编写 SCF 函数来处理 COS Bucket 中的对象创建和对象删除事件。COS 可将事件发布给 SCF 函数并将事件数据作为参数来调用该函数。用户可以在 COS Bucket 中添加存储桶通知配置，该配置可标识触发函数的事件类型和希望调用的函数名称等信息，更多内容请参考 [PutBucketNotification](#) 接口。

COS 触发器具有以下特点：

- Push 模型：COS 会监控指定的 Bucket 动作（事件类型）并调用相关函数，将事件数据推送给 SCF 函数。在推模型中使用 Bucket 通知来保存 COS 的事件源映射。
- 异步调用：COS 始终使用异步调用类型来调用函数，结果不会返回给调用方。有关调用类型的更多信息，请参阅[调用类型](#)。

COS 触发器属性

- COS Bucket（必选）：配置 COS Bucket，仅支持选择同地域下的 COS 存储桶。
- 事件类型（必选）：目前支持“文件上传”和“文件删除”两种类型，决定了触发器何时触发函数。例如，选择“文件上传”时，会在该 COS Bucket 中有文件上传时触发该函数。

COS 触发器使用限制

为了避免 COS 的事件生产投递出现错误，COS 针对每个 Bucket 的每个事件（如文件上传/文件删除等），限制只能绑定一个可触发的函数。因此，在您创建 COS 触发器时，请不要将多个函数的触发器绑定至同一个 COS Bucket 的同一个事件上。

同时，目前 COS 触发器仅支持同地域 COS Bucket 事件触发，即广州创建的 SCF 函数，在配置 COS 触发器时，仅支持选择广州（华南）的 COS Bucket。如果您想要使用特定地域的 COS Bucket 事件来触发 SCF 函数，可以通过在对应地域下创建函数来实现。

COS 触发器的事件消息结构

在指定的 COS Bucket 发生对象创建或对象删除事件时，会将类似以下的 JSON 格式事件数据发送给绑定的 SCF 函数。

```
{
  "Records": [
    {
      "event": {
        "eventVersion": "1.0",
        "eventSource": "qcs::cos",
        "eventName": "cos:ObjectCreated:*",
        "eventTime": "1970-01-01T00:00:00.000Z",
        "eventQueue": "qcs:0:cos:gz:1251111111:cos",
        "requestParameters": {
          "requestSourceIP": "111.111.111.111",
          "requestHeaders": {
            "Authorization": "Example"
          }
        }
      },
      "cos": {
        "cosSchemaVersion": "1.0",
        "cosNotificationId": "设置的或返回的 ID",
        "cosBucket": {
          "name": "bucketname",
          "appid": "1251111111",
          "region": "gz",
        },
        "cosObject": {
          "key": "/test.jpg",
          "size": "1024",
          "meta": {
            "Content-Type": "text/plain",
            "x-cos-meta-test": "自定义的 meta",
            "x-image-test": "自定义的 meta"
          }
        }
      }
    }
  ]
}
```

```
},  
"url": "访问文件的源站url"  
}  
}  
}  
]  
}
```

定时触发器

用户可以编写 SCF 函数来处理定时任务。定时器会在指定时间自动触发 SCF 函数。定时触发器具有以下特点：

- **Push 模型**：定时器指定时间到达时直接调用相关函数的 Invoke 接口来触发函数。该事件源映射关系保存在 SCF 函数中。
- **异步调用**
：定时器始终使用异步调用类型来调用函数，结果不会返回给调用方。有关调用类型的更多信息，请参阅[调用类型](#)。

定时触发器属性

- **定时器名称（必选）**：最大支持 60 个字符，支持

a-z

,

A-Z

,

0-9

,

-

和

-

。必须以字母开头，且一个函数下不支持同名的多个定时触发器。

- **触发周期（必选）**：指定的函数触发时间。用户可以使用控制台上的默认值，或选择自定义标准的

CRON 表达式来决定何时触发函数。有关 CRON 表达式的更多信息，请参考下面的内容。

Cron 表达式

创建定时触发器时，用户能够使用标准的 cron 表达式的形式自定义何时触发。

Cron 表达式语法

Cron 表达式有六个必需字段，按空格分隔。

第一位	第二位	第三位	第四位	第五位	第六位
分钟	小时	日	月	星期	年

其中，每个字段都有相应的取值范围：

字段	值	通配符
分钟	0-59	, - * /
小时	0-23	, - * /
日	1-31	, - * ? /
月	1-12 或 JAN-DEC	, - * /
星期	1-7 或 SUN-SAT	, - * ? /
年代	1970-2199	, - * /

通配符分别代表了以下意义：

通配符	含义
, (逗号)	代表取用逗号隔开的字符的并集。例如：在“小时”字段中 1, 2, 3表示1点、2点和3点
- (破折号)	包含指定范围的所有值。例如：在“日”字段中，1-15包含指定月份的 1 号到 15 号
* (星号)	表示所有值。在“小时”字段中，* 表示每个小时
/ (正斜杠)	指定增量。在“分钟”字段中，输入 1/10以指定从第一分钟开始的每隔十分钟重复。例如，第 11 分钟、第 21 分钟和第 31 分钟，依此类推

通配符	含义
? (问号)	表示忽略相应字段。例如，在“日”字段中输入 7，并希望不管 7 号是星期几都执行，则可以在“星期”字段中输入？

注意事项

- 不支持快于 5 分钟的 Cron 表达式
- 不支持在 Cron 表达式中的“日”和“星期”字段同时指定值。如果在其中一个字段中指定了值，则必须在另一个字段中使用？。

示例

下面展示了一些 Cron 表达式和相关含义的示例：

- 15 23 * * ? *

每天晚上 23:15 运行

- 0 18 ? * MON-FRI *

每个工作日下午 6:00 运行

- 0 8 1 * ? *

每个月 1 号上午 8:00 运行

- 0/15 * * * ? *

每 15 分钟运行一次

CMQ Topic 触发器

用户可以编写 SCF 函数来处理 CMQ Topic 主题队列中收到的消息。CMQ Topic 可将消息传递给 SCF 函数并将消息内容和相关信息作为参数来调用该函数。

CMQ Topic 主题队列触发器具有以下特点：

- **Push 模型**：CMQ Topic 主题队列在接受到消息后，会将消息推送给所有订阅该主题的订阅者，配置了触发 SCF 函数的情况下，函数也会被作为订阅者接收到队列的推送。在推模型中，CMQ Topic 主题队列会保存对 SCF 云函数的事件源映射。
- **异步调用**：CMQ Topic 主题队列始终使用异步调用类型来调用函数，结果不会返回给调用方。有关调用类型的更多信息，请参阅[调用类型](#)。

CMQ Topic 主题队列触发器属性

- **CMQ Topic (必选)**：配置的 CMQ Topic 主题队列，仅支持选择同地域下的 CMQ 队列。

CMQ Topic 触发器绑定限制

由于 CMQ Topic 主题队列，在单个 Topic 下最多支持 100 个订阅者。因此在 SCF 云函数触发器绑定时，如果达到此限制，可能绑定失败。在未达到此限制前，一个 Topic 下可以绑定多个 SCF 云函数。

同时，目前 CMQ Topic 触发器仅支持同地域 CMQ Topic 消息发，即广州创建的 SCF 函数，在配置 CMQ Topic 触发器时，仅支持选择广州（华南）的 CMQ Topic。如果您想要使用特定地域的 CMQ Topic 消息来触发 SCF 函数，可以通过在对应地域下创建函数来实现。

CMQ Topic 触发器的事件消息结构

在指定的 CMQ Topic 主题队列接受到消息时，会将类似以下的 JSON 格式事件数据发送给绑定的 SCF 函数。

```
{  
  "Records": [  

```

```
{  
  "CMQ": {  
    "type": "topic",  
    "topicOwner": "120xxxxx",  
    "topicName": "testtopic",  
    "subscriptionName": "xxxxxx",  
    "publishTime": "1970-01-01T00:00:00.000Z",  
    "msgId": "123345346",  
    "requestId": "123345346",  
    "msgBody": "Hello from CMQ!",  
    "msgTag": ["tag1", "tag2"]  
  }  
}  
]  
}
```

API网关触发器

用户可以编写 SCF 云函数来实现 Web 后端服务，并通过 API 网关对外提供服务。API 网关会将 API 请求内容以参数形式传递给函数，并将函数返回作为 API 响应返回给请求方。

API 网关触发器具有以下特点：

- **Push 模型**：API 网关在接受到 API 请求后，如果 API 在网关上的后端配置了对接云函数，函数将会被触发运行，同时 API 网关会将 API 请求的相关信息，例如具体接受到请求的服务和 API 规则，请求的实际路径，方法，请求的 path，header，query 等参数，全部封装为请求入参，以 event 入参的形式发送给触发的函数。
- **同步调用**：API 网关以同步调用的方式来调用函数，会在 API 网关中配置的超时时间未到前等待函数返回。有关调用类型的更多信息，请参阅[调用类型](#)。

API 网关触发器配置

API 网关触发器配置并不在云函数中进行配置，而是在 API 网关中进行。在 API 网关中配置 API 规则时，后端配置可选 Cloud Function，在选择 Cloud Function 后，即可选择与 API 服务相同地域的云函数。

在 API 网关配置对接云函数时，同样会配置超时时间。API 网关中的请求超时时间，和云函数的运行超时时间，两者分别生效。超时规则如下：

- API 网关超时时间 > 云函数超时时间：云函数超时先生效，API 请求响应为 200 http code，但返回内容为云函数超时报错内容。
- API 网关超时时间 < 云函数超时时间：API 网关超时先生效，API 请求返回 5xx http code，标识请求超时。

API 网关触发器绑定限制

API 网关中，一条 API 规则仅能绑定一个云函数，但一个云函数可以被多个 API 规则绑定为后端。同时，目前 API 网关触发器仅支持同地域云函数绑定，即广州创建的云函数，仅支持被广州创建的 API 服务中规则所绑定和触发。如果您想要使用特定地域的 API

网关配置来触发云函数，可以通过在对应地域下创建函数来实现。

API 网关触发器的事件消息结构

在API网关接收到请求时，会将类似以下的 JSON 格式事件数据发送给绑定的云函数。

```
{
  "requestContext": {
    "serviceName": "testsvc",
    "path": "/test/{path}",
    "httpMethod": "POST",
    "requestId": "c6af9ac6-7b61-11e6-9a41-93e8deadbeef",
    "identity": {
      "secretId": "abdcxxxxxxxxsdfs"
    },
    "sourceIp": "10.0.2.14",
    "stage": "prod"
  },
  "headers": {
    "Accept-Language": "en-US,en,cn",
    "Accept": "text/html,application/xml,application/json",
    "Host": "service-3ei3tii4-251000691.ap-guangzhou.apigateway.myqcloud.com",
    "User-Agent": "User Agent String"
  },
  "body": "{\"test\": \"body\"}",
  "pathParameters": {
    "path": "value"
  },
  "queryStringParameters": {
    "foo": "bar"
  },
  "headerParameters": {
    "Refer": "10.0.2.14"
  },
}
```

```
"stageVariables": {  
  "stage": "test"  
},  
"path": "/test/value",  
"query": "foo=bar&bob=alice",  
"httpMethod": "POST"  
}
```

其中，

requestContext

结构标识了 API 请求来源服务和 API 规则、所生效的环境、请求id、认证信息等内容。

pathParameters

,

queryStringParameters

,

headerParameters

等内容包含了 API 规则中所配置入参的参数名称和实际值，

httpMethod

,

path

,

body

、

headers

等内容包含了实际请求内容。

API 网关触发器对云函数返回内容的处理

由于使用的是同步调用，API 网关会触发云函数后等待函数执行完成，并将函数返回，作为 API 请求的响应内容返回给 API 请求的发起方，因此，此特性可以用来使用云函数实现 API 后端服务，将 API 请求进行处理后，返回处理结果并被传递至 API 请求者。