

ALIBABA CLOUD

阿里云

分布式任务调度 SchedulerX  
API参考

文档版本：20201016

 阿里云

## 法律声明

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

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

# 通用约定

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

# 目录

1.API概览	05
2.调用方式	06
3.获取AccessKey	08
4.CreateAppGroup	09
5.CreateJob	13
6.GetJobInfo	22
7.UpdateJob	31
8.DeleteJob	36
9.EnableJob	38
10.DisableJob	41
11.ExecuteJob	44
12.StopInstance	47
13.GetJobInstanceList	49
14.GetJobInstance	54
15.ExecuteWorkflow	59
16.EnableWorkflow	61
17.DisableWorkflow	63
18.DeleteWorkflow	65

# 1.API概览

分布式任务调度SchedulerX2.0提供以下API接口。

## API列表

API	描述
<b>CreateAppGroup</b>	调用CreateAppGroup创建应用分组。
<b>CreateJob</b>	调用CreateJob创建任务。
<b>GetJobInfo</b>	调用GetJobInfo获取指定Jobid任务详情，通常用来更新任务。
<b>UpdateJob</b>	调用UpdateJob更新任务配置信息。
<b>DeleteJob</b>	调用DeleteJob删除指定任务。
<b>EnableJob</b>	调用EnableJob启用指定任务。
<b>DisableJob</b>	调用DisableJob停用指定任务。
<b>ExecuteJob</b>	调用ExecuteJob触发一次任务。
<b>StopInstance</b>	调用StopInstance终止某次正在运行的实例。
<b>GetJobInstanceList</b>	调用GetJobInstanceList获取指定任务ID的执行实例列表。
<b>GetJobInstance</b>	调用GetJobInstance获取指定任务实例详情。
<b>ExecuteWorkflow</b>	调用ExecuteWorkflow触发一次工作流。
<b>EnableWorkflow</b>	调用EnableWorkflow启用指定工作流。
<b>DisableWorkflow</b>	调用DisableWorkflow禁用指定工作流。
<b>DeleteWorkflow</b>	调用DeleteWorkflow删除指定工作流。

## 2. 调用方式

您可以使用SDK调用分布式任务调度2.0的API，完成日常的管理操作。

### 子账号使用

您可以使用子账号（RAM）调用分布式任务调度2.0的API，但需要先获取、配置子账号的AccessKey ID和AccessKey Secret，并由主账号对子账号授予相应资源的操作权限。

### 支持地域列表

在使用SDK调用API的时候需要用到地域的Region ID、Domain等信息，参见下面的表格。

 说明 下表中Domain（VPC）适用于没有公网的VPC的环境。

名称	RegionId	Domain	Domain（VPC）
华东1（杭州）	cn-hangzhou	schedulerx.cn-hangzhou.aliyuncs.com	schedulerx-vpc.cn-hangzhou.aliyuncs.com
华南1（深圳）	cn-shenzhen	schedulerx.cn-shenzhen.aliyuncs.com	schedulerx-vpc.cn-shenzhen.aliyuncs.com
华东2（上海）	cn-shanghai	schedulerx.cn-shanghai.aliyuncs.com	schedulerx-vpc.cn-shanghai.aliyuncs.com
华北2（北京）	cn-beijing	schedulerx.cn-beijing.aliyuncs.com	schedulerx-vpc.cn-beijing.aliyuncs.com
华北3（张家口）	cn-zhangjiakou	schedulerx.cn-zhangjiakou.aliyuncs.com	schedulerx-vpc.cn-zhangjiakou.aliyuncs.com
美国（弗尼亚）	us-east-1	schedulerx.us-east-1.aliyuncs.com	schedulerx-vpc.us-east-1.aliyuncs.com
政务云	cn-north-2-gov-1	schedulerx.cn-north-2-gov-1.aliyuncs.com	schedulerx-vpc.cn-north-2-gov-1.aliyuncs.com
公网测试	cn-test	schedulerx.aliyuncs.com	无

### 添加SDK依赖

在应用程序的文件中添加OpenAPI的SDK依赖。

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-schedulerx2</artifactId>
  <version>1.0.3</version>
</dependency>

<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>4.3.3</version>
</dependency>
```

## 使用SDK调用API

下面以一个通用的示例说明如何使用SDK调用API。

```
public static void main(String[] args) throws Exception{
    // OpenAPI的接入点，具体参见支持地域列表和购买实例的地域。String regionId = "XXXXX";
    //鉴权使用的AccessKeyId。String accessKeyId = "XXXXXXXXXXXXXXXXXXXX";
    //鉴权使用的AccessKeySecret。String accessKeySecret = "XXXXXXXXXXXXXXXXXXXX";
    //产品名称String productName = "schedulerx2";
    //参见支持地域列表，选择Domain。String domain = "schedulerx.xxxx.aliyuncs.com";
    //构建OpenAPI客户端。DefaultProfile.addEndpoint(regionId, productName, domain);
    DefaultProfile defaultProfile = DefaultProfile.getProfile(regionId, accessKeyId, accessKeySecret);
    DefaultAcsClient defaultAcsClient = new DefaultAcsClient(defaultProfile);
    //构建接口请求（根据接口中参数类型和是否必填进行填充）。EnableJobRequest request = new EnableJobRequest();
    request.setJobId(jobId);
    request.setNamespace("xxxxxx");
    request.setGroupId("yyyyyy");
    //发送请求。EnableJobResponse response = defaultAcsClient.getAcsResponse(request);
    if (!response.getSuccess()) {
        System.out.println(response.getMessage());
        System.out.println("EnableJob: "+response.getRequestId());
    }
}
```


## 3. 获取AccessKey

您可以为阿里云主账号和子账号创建一个访问密钥（AccessKey）。在调用阿里云API时您需要使用AccessKey完成身份验证。

### 背景信息

AccessKey包括AccessKey ID和AccessKey Secret。

- AccessKey ID：用于标识用户。
- AccessKey Secret：用于验证用户的密钥。AccessKey Secret必须保密。

 **警告** 主账号AccessKey泄露会威胁您所有资源的安全。建议使用子账号（RAM用户）AccessKey进行操作，可以有效降低AccessKey泄露的风险。

### 操作步骤

1. 使用主账号登录 [阿里云管理控制台](#)。
2. 将鼠标置于页面右上方的账号图标，单击**AccessKey 管理**。
3. 在安全提示页面，选择获取主账号或者子账号的Accesskey。

4. 获取账号Accesskey。
  - 获取主账号AccessKey
    - a. 在安全提示页面单击**继续使用AccessKey**。
    - b. 在安全信息管理页面，单击**创建AccessKey**。
    - c. 在手机验证页面，获取并填入校验码，单击**确定**。
    - d. 在新建用户AccessKey页面，展开AccessKey详情，查看AccessKey ID和AccessKey Secret。可以单击**保存AK信息**，下载AccessKey信息。
  - 获取子账号AccessKey
    - a. 在安全提示页面单击**开始使用子用户AccessKey**
      - 如果是未创建RAM用户，请在系统跳转的[RAM访问控制台](#)的创建用户页面，创建RAM用户，创建完成后会自动为RAM用户生成访问密钥（AccessKey）。详情请参见[创建RAM用户](#)。
      - 如果是已创建RAM用户，继续执行后续步骤直接获取RAM用户的Accesskey。
    - b. 登录[RAM访问控制台](#)，在左侧导航栏选择**人员管理 > 用户**。
    - c. 在用户页面搜索需要获取AccessKey的用户。
    - d. 单击目标用户登录名称，在用户详情页认证管理页签下的**用户 AccessKey**区域，单击**创建AccessKey**。
    - e. 在创建 AccessKey页面，查看AccessKey ID和AccessKey Secret。  
您还可以单击**下载CSV文件**或者**复制**，完成对AccessKey信息的保存。



## 4.CreateAppGroup

调用CreateAppGroup创建应用分组。

### 请求参数

名称	类型	是否必选	示例值	描述
AppName	String	是	DocTest	应用名称。
GroupId	String	是	TestSchedulerx.defaultGroup	应用ID，在控制台的应用管理页面中获取。
Namespace	String	是	adcf35d-e2fe-4fe9-bbaa-20e90ffc****	命名空间ID，在控制台的命名空间页面中获取。
NamespaceName	String	是	Test	命名空间名称。
RegionId	String	是	cn-hangzhou	地域ID。
NamespaceSource	String	否	schedulerx	特殊第三方才需要填写。
Description	String	否	Test	应用描述。
MaxJobs	Integer	否	1000	最大任务数。
AlarmJson	String	否	AlarmTest	告警JSON。

### 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码
Data	Struct		任务分组信息
AppGroupId	Long	6607	任务分组ID

名称	类型	示例值	描述
Message	String	Your request is denied as lack of ssl protect.	错误信息，仅错误时返回错误信息。
RequestId	String	883AFE93-FB03-4FA9-A958-E750C6DE120C	请求唯一ID
Success	Boolean	true	创建应用是否成功。取值如下： <ul style="list-style-type: none"> <li>• true：创建应用成功。</li> <li>• false：创建应用失败。</li> </ul>

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=CreateAppGroup
&AppName=DocTest
&GroupId=TestSchedulerx.defaultGroup
&Namespace=adcf35d-e2fe-4fe9-bbaa-20e90ffc****
&NamespaceName=Test
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<CreateAppGroupResponse>
  <RequestId>883AFE93-FB03-4FA9-A958-E750C6DE120C</RequestId>
  <Message>Your request is denied as lack of ssl protect.</Message>
  <Data>
    <AppGroupId>6607</AppGroupId>
  </Data>
  <Code>200</Code>
  <Success>true</Success>
</CreateAppGroupResponse>
```

#### JSON 格式

```
{
  "RequestId": "883AFE93-FB03-4FA9-A958-E750C6DE120C",
  "Message": "Your request is denied as lack of ssl protect.",
  "Data": {
    "AppGroupId": 6607
  },
  "Code": 200,
  "Success": true
}
```

## 示例Demo

```
package com.alibaba.schedulerx.pop;

import com.alibaba.schedulerx.common.util.JsonUtil;

import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.schedulerx2.model.v20190430.CreateAppGroupRequest;
import com.aliyuncs.schedulerx2.model.v20190430.CreateAppGroupResponse;

public class TestCreateAppGroup {

    public static void main(String[] args) throws Exception {
        // OpenAPI的接入点，具体查看上表支持地域列表以及购买机器地域填写。
        String regionId = "cn-shanghai";
        //鉴权使用的AccessKeyId，由阿里云官网控制台获取。
        String accessKeyId = "<yourAccessKeyId>";
        //鉴权使用的AccessKeySecret，由阿里云官网控制台获取。
        String accessKeySecret = "<yourAccessKeySecret>";
        //产品名称
        String productName = "schedulerx2";
        //对照支持地域列表选择Domain填写。
        String domain = "schedulerx.cn-shanghai.aliyuncs.com";
        //构建OpenAPI客户端。
        DefaultProfile.addEndpoint(regionId, productName, domain);
        DefaultProfile defaultProfile = DefaultProfile.getProfile(regionId, accessKeyId, accessKeySecret);
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);

        CreateAppGroupRequest request = new CreateAppGroupRequest();
        request.setNamespace("xxxxxx");
```

```
request.setNamespace( "xxxxx" );
request.setNamespaceName("xxxx");
request.setAppName("xxxx");
request.setDescription("xxx");
request.setGroupId("xxx");

//发送请求。
CreateAppGroupResponse response = client.getAcsResponse(request);
if (!response.getSuccess()) {
    System.out.println(response.getMessage());
    System.out.println("createApp: "+response.getRequestId());
} else {
    System.out.println(JsonUtil.toJson(response));
}
}
}
```

## 5.CreateJob

调用CreateJob创建任务。

### 请求参数

名称	类型	是否必选	示例值	描述
ExecuteMode	String	是	standalone	任务执行模式，目前支持以下几种任务执行模式： <ul style="list-style-type: none"> <li>• 单机运行：standalone</li> <li>• 广播运行：broadcastst</li> <li>• 并行计算：parallel</li> <li>• 内存网格：grid</li> <li>• 网格计算：batch</li> <li>• 分片运行：shard</li> </ul>
GroupId	String	是	testSchedulerx.defaultGroup	应用ID，在控制台的应用管理页面中获取。
JobType	String	是	java	任务类型，目前支持以下几种任务类型： <ul style="list-style-type: none"> <li>• java</li> <li>• python</li> <li>• shell</li> <li>• go</li> <li>• http</li> <li>• nodejs</li> </ul>
Name	String	是	helloworld	任务名。
Namespace	String	是	adcfc35d-e2fe-4fe9-bbaa-20e90ffc****	命名空间ID，在控制台的命名空间页面中获取。
RegionId	String	是	cn-hangzhou	地域ID。
TimeType	Integer	是	1	时间类型，目前支持以下几种时间类型： <ul style="list-style-type: none"> <li>• cron: 1</li> <li>• fixed_rate: 3</li> <li>• second_delay: 4</li> <li>• api: 100</li> </ul>

名称	类型	是否必选	示例值	描述
Description	String	否	Test	任务描述。
Content	String	否	echo 'hello'	任务类型选择为python、shell、go时，必填的脚本代码内容。
Parameters	String	否	test	用户自定义参数，运行时可以获取。
MaxConcurrency	Integer	否	1	最大同时运行实例数量，默认值为1，即上次触发没有运行结束，即使到了运行时刻也不会进行下次触发。
MaxAttempt	Integer	否	0	错误最大重试次数，根据业务需求填写，默认值为0。
AttemptInterval	Integer	否	30	错误重试间隔，单位s，默认值为30。
ClassName	String	否	com.alibaba.schedulerx.test.helloworld	任务接口类完整路径。 当您选择Java任务类型时，才有该字段且必须填写完整路径。
JarUrl	String	否	暂不支持，不用填写	上传到OSS的完整路径。 如果选择JAR包运行，可以将相应JAR包上传到OSS的该路径下。
PageSize	Integer	否	100	并行网格任务高级配置，单次拉取子任务数，默认值为100。
ConsumerSize	Integer	否	5	并行网格任务高级配置，单机单次触发执行线程数，默认值为5。
QueueSize	Integer	否	10000	并行网格任务高级配置，子任务队列缓存上限，默认值为10000。
DispatcherSize	Integer	否	5	并行网格任务高级配置，子任务分发线程数，默认值为5。

名称	类型	是否必选	示例值	描述
TimeExpression	String	否	0 0/10 ***?	<p>时间表达式，根据选择的时间类型设置时间表达式。</p> <ul style="list-style-type: none"> <li><b>cron</b>: 填写标准的cron表达式，支持在线验证。</li> <li><b>api</b>: 无时间表达式。</li> <li><b>fixed_rate</b>: 填写具体固定频率值，单位s。如30表示每隔30s触发一次。</li> <li><b>second_delay</b>: 填写固定延迟多少秒执行一次（1s~60s可选）。</li> </ul>
Calendar	String	否	暂不支持，不用填写	cron类型可以选择填写自定义日历。
DataOffset	Integer	否	2400	cron类型可以选择时间偏移，单位s。
TimeoutEnable	Boolean	否	false	<p>超时报警开关。取值如下：</p> <ul style="list-style-type: none"> <li><b>true</b>: 开启超时报警开关。</li> <li><b>false</b>: 关闭超时报警开关。</li> </ul>
Timeout	Long	否	7200	超时阈值，单位s，默认值7200。
TimeoutKillEnable	Boolean	否	false	<p>超时终止开关。取值如下：</p> <ul style="list-style-type: none"> <li><b>true</b>: 开启超时终止开关。</li> <li><b>false</b>: 关闭超时终止开关。</li> </ul>
FailEnable	Boolean	否	false	<p>失败报警开关。取值如下：</p> <ul style="list-style-type: none"> <li><b>true</b>: 开启失败报警开关。</li> <li><b>false</b>: 关闭失败报警开关。</li> </ul>
SendChannel	String	否	sms	报警发送形式，目前只支持短信发送报警，默认值sms。
ContactInfo.N.UserName	String	否	userA	报警联系人姓名。
ContactInfo.N.UserPhone	String	否	1381111****	报警接收手机号。
NamespaceSource	String	否	schedulerx	特殊第三方才需要填写。

名称	类型	是否必选	示例值	描述
TaskMaxAttempt	Integer	否	0	并行网格任务高级配置，子任务失败重试次数，默认值为0。
TaskAttemptInterval	Integer	否	0	并行网格任务高级配置，子任务失败重试间隔，默认值为0。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码
Data	Struct		任务详细信息
JobId	Long	92583	任务ID
Message	String	groupid not exist groupid: testSchedulerx.defaultGroup namespace: adcf35d-e2fe-4fe9-bbaa-20e90ffc****	错误信息，仅错误时返回错误信息。
RequestId	String	39090022-1F3B-4797-8518-6B61095F1AF0	请求唯一ID
Success	Boolean	true	创建任务是否成功。取值如下： <ul style="list-style-type: none"> <li>true：创建任务成功。</li> <li>false：创建任务失败。</li> </ul>

## 示例

请求示例



```
http(s)://[Endpoint]/?Action=CreateJob
&ExecuteMode=standalone
&GroupId=testSchedulerx.defaultGroup
&JobType=java
&Name=helloworld
&Namespace=adcf35d-e2fe-4fe9-bbaa-20e90ffc****
&RegionId=cn-hangzhou
&TimeType=1
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<CreateJobResponse>
  <RequestId>39090022-1F3B-4797-8518-6B61095F1AF0</RequestId>
  <Message>groupid not exist groupId: testSchedulerx.defaultGroup namespace: adcf35d-e2fe-4fe9-
bbaa-20e90ffc****</Message>
  <Data>
    <JobId>92583</JobId>
  </Data>
  <Code>200</Code>
  <Success>>true</Success>
</CreateJobResponse>
```

#### JSON 格式

```
{
  "RequestId": "39090022-1F3B-4797-8518-6B61095F1AF0",
  "Message": "groupid not exist groupId: testSchedulerx.defaultGroup namespace: adcf35d-e2fe-4fe
9-bbaa-20e90ffc****",
  "Data": {
    "JobId": 92583
  },
  "Code": 200,
  "Success": true
}
```

## 创建JAVA任务

```
package com.alibaba.schedulerx.pop;
```

```
import java.util.ArrayList;
import java.util.List;

import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.schedulerx2.model.v20190430.CreateJobRequest;
import com.aliyuncs.schedulerx2.model.v20190430.CreateJobRequest.ContactInfo;
import com.aliyuncs.schedulerx2.model.v20190430.CreateJobResponse;

public class CreateJavaJob {

    public static void main(String[] args) throws Exception {
        // OpenAPI的接入点，具体查看上表支持地域列表以及购买机器地域填写。
        String regionId = "cn-shanghai";
        // 鉴权使用的AccessKeyId，由阿里云官网控制台获取。
        String accessKeyId = "<yourAccessKeyId>";
        // 鉴权使用的AccessKeySecret，由阿里云官网控制台获取。
        String accessKeySecret = "<yourAccessKeySecret>";
        // 产品名称。
        String productName = "schedulerx2";
        // 对照支持地域列表选择Domain填写。
        String domain = "schedulerx.cn-shanghai.aliyuncs.com";
        // 构建OpenAPI客户端。
        DefaultProfile.addEndpoint(regionId, productName, domain);
        DefaultProfile defaultProfile = DefaultProfile.getProfile(regionId, accessKeyId, accessKeySecret);
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);

        CreateJobRequest request = new CreateJobRequest();
        request.setJobType("java");
        request.setExecuteMode("standalone");
        request.setDescription("test");
        request.setName("helloworld");
        request.setClassName("com.alibaba.schedulerx.test.helloworld");
        request.setTimeType(1);
        request.setTimeExpression("0 0/10 * * * ?");
        request.setNamespace("xxxxx");
        request.setGroupid("xxxxxxx");
        // monitor
        request.setTimeoutEnable(true);
        request.setTimeoutKillEnable(true);
    }
}
```

```
request.setFailEnable(true);
request.setTimeout(12300L);
List<ContactInfo> contactInfosList = new ArrayList<>();
ContactInfo contactInfo1 = new ContactInfo();
contactInfo1.setUserName("userA");
contactInfo1.setUserPhone("1381111****");
ContactInfo contactInfo2 = new ContactInfo();
contactInfo2.setUserName("userB");
contactInfo2.setUserPhone("1382222****");
contactInfosList.add(contactInfo1);
contactInfosList.add(contactInfo2);
request.setContactInfos(contactInfosList);
// attrs
//request.setQueueSize(123);
request.setTaskMaxAttempt(1);
request.setTaskAttemptInterval(100);
CreateJobResponse response = client.getAcsResponse(request);
if (response.getSuccess()) {
    System.out.println("jobId=" + response.getData().getJobId());
} else {
    System.out.println(response.getMessage());
}
}
```

## 创建HTTP任务

```
package com.alibaba.schedulerx.pop;

import com.alibaba.schedulerx.common.domain.HttpAttribute;
import com.alibaba.schedulerx.common.util.JsonUtil;

import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.schedulerx2.model.v20190430.CreateJobRequest;
import com.aliyuncs.schedulerx2.model.v20190430.CreateJobResponse;

public class CreateHttpJob {
```

```
public static void main(String[] args) throws Exception {
    // OpenAPI的接入点，具体查看上表支持地域列表以及购买机器地域填写。
    String regionId = "cn-shanghai";
    //鉴权使用的AccessKeyId，由阿里云官网控制台获取。
    String accessKeyId = "<yourAccessKeyId>";
    //鉴权使用的AccessKeySecret，由阿里云官网控制台获取。
    String accessKeySecret = "<yourAccessKeySecret>";
    //产品名称。
    String productName = "schedulerx2";
    //对照支持地域列表选择Domain填写。
    String domain = "schedulerx.cn-shanghai.aliyuncs.com";
    //构建OpenAPI客户端。
    DefaultProfile.addEndpoint(regionId, productName, domain);
    DefaultProfile defaultProfile = DefaultProfile.getProfile(regionId, accessKeyId, accessKeySecret);
    DefaultAcsClient client = new DefaultAcsClient(defaultProfile);

    CreateJobRequest request = new CreateJobRequest();
    request.setNamespace("xxxxxxx");
    request.setGroupId("xxxxxx");
    request.setJobType("http");
    request.setName("testHttpJob");
    request.setDescription("testHttpJob");
    request.setTimeType(1);
    request.setTimeExpression("20 0/5 * * * ?");
    request.setExecuteMode("standalone");
    HttpAttribute httpAttribute = new HttpAttribute();
    httpAttribute.setUrl("http://192.168.0.0:8080/test");
    httpAttribute.setMethod("GET");
    httpAttribute.setTimeout(10); //单位秒
    httpAttribute.setRespKey("code");
    httpAttribute.setRespValue("200");
    request.setContent(JsonUtil.toJson(httpAttribute));
    //POST参数，格式key1=value1&key2=value2。
    request.setParameters("key1=value1&key2=value2");

    //发送请求。
    CreateJobResponse response = client.getAcsResponse(request);
    if (!response.isSuccess()) {
        System.out.println(response.getMessage());
        System.out.println("createApp: "+response.getRequestId());
    }
}
```

```
} else {  
    System.out.println(JsonUtil.toJson(response));  
}  
}  
  
}
```

## 6. GetJobInfo

调用GetJobInfo获取指定Jobid任务详情，通常用来更新任务。

### 请求参数

名称	类型	是否必选	示例值	描述
GroupId	String	是	testSchedulerx.defaultGroup	应用ID，在控制台的应用管理页面中获取。
JobId	Long	是	92583	任务ID，在控制台的任务管理页面中获取。
Namespace	String	是	adcf35d-e2fe-4fe9-bbaa-20e90ffc****	命名空间ID，在控制台的命名空间页面中获取。
RegionId	String	是	cn-hangzhou	地域ID。
NamespaceSource	String	否	schedulerx	特殊第三方才需要填写。

### 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码
Data	Struct		指定任务信息
JobConfigInfo	Struct		任务配置信息
AttemptInterval	Integer	30	错误重试间隔时间，单位s，默认值为30。
ClassName	String	com.alibaba.test.helloworld	任务接口类完整路径，仅是Java类型任务时有该字段。
Content	String	echo "clear" > /home/admin/edas-container/logs/catalina.out	脚本类型任务，具体执行代码。

名称	类型	示例值	描述
Description	String	test	任务描述
ExecuteMode	String	standalone	<p>任务执行模式，目前包含以下几种任务执行模式：</p> <ul style="list-style-type: none"> <li>• 单机运行：standalone</li> <li>• 广播运行：broadcast</li> <li>• 并行计算：parallel</li> <li>• 内存网格：grid</li> <li>• 网格计算：batch</li> <li>• 分片运行：shard</li> </ul>
JarUrl	String	https://test.oss-cn-hangzhou.aliyuncs.com/schedulerX/test.jar	<p>上传到OSS的完整路径。</p> <p>如果选择JAR包运行，可以将相应JAR包上传到OSS的该路径下。</p>
JobMonitorInfo	Struct		任务监控信息
ContactInfo	Array of ContactInfo		联系人信息
UserName	String	userA	用户名称
UserPhone	String	1381111****	用户手机号
MonitorConfig	Struct		报警开关以及阈值配置
FailEnable	Boolean	true	<p>失败报警开关。取值如下：</p> <ul style="list-style-type: none"> <li>• true：开启失败报警开关。</li> <li>• false：关闭失败报警开关。</li> </ul>
SendChannel	String	sms	报警发送形式，目前只支持sms。
Timeout	Long	12300	超时阈值，单位s，默认7200。
TimeoutEnable	Boolean	true	<p>超时报警开关。取值如下：</p> <ul style="list-style-type: none"> <li>• true：开启超时报警开关。</li> <li>• false：关闭超时报警开关。</li> </ul>

名称	类型	示例值	描述
TimeoutKillEnable	Boolean	true	超时终止本次触发开关，默认关闭。
MapTaskXAttrs	Struct		高级配置，仅限于并行计算、内存网格和网格计算使用。
ConsumerSize	Integer	5	单机单次触发执行线程数，默认值为5。
DispatcherSize	Integer	5	子任务分发线程数，默认值为5。
PageSize	Integer	100	并行任务单次拉取子任务数，默认值为100。
QueueSize	Integer	10000	子任务队列缓存上限，默认值为10000。
TaskAttemptInterval	Integer	0	子任务失败重试间隔。
TaskMaxAttempt	Integer	0	子任务失败重试次数。
MaxAttempt	Integer	0	错误最大重试次数，根据业务需求填写，默认值为0。
MaxConcurrency	String	1	最大同时运行实例数量，默认值为1，即上次触发没有运行结束，即使到了运行时刻也不会进行下次触发。
Name	String	helloworld	任务名
Parameters	String	test	用户自定义参数，运行时可以获取。
Status	Integer	1	任务状态。取值如下： <ul style="list-style-type: none"> <li>1: 启用，可以被正常触发。</li> <li>0: 禁用，不会被触发。</li> </ul>
TimeConfig	Struct		时间配置信息
Calendar	String	工作日	cron类型可以选择填写自定义日历。



名称	类型	示例值	描述
DataOffset	Integer	0	cron类型可以选择时间偏移，单位s。
TimeExpression	String	0 0/10 * * * ?	时间表达式，目前支持以下几种时间表达类型： <ul style="list-style-type: none"> <li>• api：无时间表达式。</li> <li>• fix_rate：具体固定频率值，如30表示每隔30s触发一次。</li> <li>• cron：标准的cron表达式。</li> <li>• second_delay：固定延迟多少秒执行一次（1s~60s可选）。</li> </ul>
TimeType	Integer	1	时间配置类型，目前支持以下几种时间类型： <ul style="list-style-type: none"> <li>• cron：1</li> <li>• fix_rate：3</li> <li>• second_delay：4</li> <li>• api：100</li> </ul>
Message	String	jobid: 92583 not match groupid: testSchedulerx.defaultGroup	错误信息，仅错误时返回错误信息。
RequestId	String	4F68ABED-AC31-4412-9297-D9A8F0401108	请求唯一ID
Success	Boolean	true	获取任务详情是否成功。取值如下： <ul style="list-style-type: none"> <li>• true：获取任务详情成功。</li> <li>• false：获取任务详情失败。</li> </ul>

## 示例

### 请求示例

```

http(s)://[Endpoint]/?Action=GetJobInfo
&GroupId=testSchedulerx.defaultGroup
&JobId=92583
&Namespace=adcf35d-e2fe-4fe9-bbaa-20e90ffc****
&<公共请求参数>
    
```

### 正常返回示例

XML 格式

```
<GetJobInfoResponse>
  <RequestId>4F68ABED-AC31-4412-9297-D9A8F0401108</RequestId>
  <Message>jobid: 92583 not match groupid: testSchedulerx.defaultGroup</Message>
  <Data>
    <JobConfigInfo>
      <Status>1</Status>
      <MaxAttempt>0</MaxAttempt>
      <Description>test</Description>
      <Parameters>test</Parameters>
      <JarUrl>https://test.oss-cn-hangzhou.aliyuncs.com/schedulerX/test.jar</JarUrl>
      <MaxConcurrency>1</MaxConcurrency>
      <TimeConfig>
        <TaskAttemptInterval>0</TaskAttemptInterval>
        <QueueSize>10000</QueueSize>
        <PageSize>100</PageSize>
        <DispatcherSize>5</DispatcherSize>
        <TaskMaxAttempt>0</TaskMaxAttempt>
        <Calendar>工作日</Calendar>
        <TimeExpression>0 0/10 * * * ?</TimeExpression>
        <ContactInfo>
          <UserName>userA</UserName>
          <UserPhone>1381111****</UserPhone>
        </ContactInfo>
        <ConsumerSize>5</ConsumerSize>
        <DataOffset>0</DataOffset>
        <TimeType>1</TimeType>
        <MonitorConfig>
          <TimeoutEnable>true</TimeoutEnable>
          <Timeout>12300</Timeout>
          <FailEnable>true</FailEnable>
          <SendChannel>sms</SendChannel>
          <TimeoutKillEnable>true</TimeoutKillEnable>
        </MonitorConfig>
      </TimeConfig>
      <Name>helloworld</Name>
      <MapTaskXAttrs>
        <TaskAttemptInterval>0</TaskAttemptInterval>
        <QueueSize>10000</QueueSize>
        <PageSize>100</PageSize>
        <DispatcherSize>5</DispatcherSize>
        <TaskMaxAttempt>0</TaskMaxAttempt>
```

```
<Calendar>工作日</Calendar>
<TimeExpression>0 0/10 * * * ?</TimeExpression>
<ContactInfo>
  <UserName>userA</UserName>
  <UserPhone>1381111****</UserPhone>
</ContactInfo>
<ConsumerSize>5</ConsumerSize>
<DataOffset>0</DataOffset>
<TimeType>1</TimeType>
<MonitorConfig>
  <TimeoutEnable>true</TimeoutEnable>
  <Timeout>12300</Timeout>
  <FailEnable>true</FailEnable>
  <SendChannel>sms</SendChannel>
  <TimeoutKillEnable>true</TimeoutKillEnable>
</MonitorConfig>
</MapTaskXAttrs>
<JobMonitorInfo>
  <TaskAttemptInterval>0</TaskAttemptInterval>
  <QueueSize>10000</QueueSize>
  <PageSize>100</PageSize>
  <DispatcherSize>5</DispatcherSize>
  <TaskMaxAttempt>0</TaskMaxAttempt>
  <Calendar>工作日</Calendar>
  <TimeExpression>0 0/10 * * * ?</TimeExpression>
  <ContactInfo>
    <UserName>userA</UserName>
    <UserPhone>1381111****</UserPhone>
  </ContactInfo>
  <ConsumerSize>5</ConsumerSize>
  <DataOffset>0</DataOffset>
  <TimeType>1</TimeType>
  <MonitorConfig>
    <TimeoutEnable>true</TimeoutEnable>
    <Timeout>12300</Timeout>
    <FailEnable>true</FailEnable>
    <SendChannel>sms</SendChannel>
    <TimeoutKillEnable>true</TimeoutKillEnable>
  </MonitorConfig>
</JobMonitorInfo>
<Content>echo "clear" &amp;pn:nt: /home/admin/edas-container/logs/catalina.out</Content>
```

```

<Content><end_clear_dump>gt; /home/admin/eas-container/logs/catalina.out</Content>
<ClassName>com.alibaba.test.helloworld</ClassName>
<AttemptInterval>30</AttemptInterval>
<ExecuteMode>standalone</ExecuteMode>
</JobConfigInfo>
</Data>
<Code>200</Code>
<Success>true</Success>
</GetJobInfoResponse>

```

### JSON 格式

```

{
  "RequestId": "4F68ABED-AC31-4412-9297-D9A8F0401108",
  "Message": "jobid: 92583 not match groupId: testSchedulerx.defaultGroup",
  "Data": {
    "JobConfigInfo": {
      "Status": 1,
      "MaxAttempt": 0,
      "Description": "test",
      "Parameters": "test",
      "JarUrl": "https://test.oss-cn-hangzhou.aliyuncs.com/schedulerX/test.jar",
      "MaxConcurrency": 1,
      "TimeConfig": {
        "TaskAttemptInterval": 0,
        "QueueSize": 10000,
        "PageSize": 100,
        "DispatcherSize": 5,
        "TaskMaxAttempt": 0,
        "Calendar": "工作日",
        "TimeExpression": "0 0/10 * * * ?",
        "ContactInfo": {
          "UserName": "userA",
          "UserPhone": "1381111****"
        },
        "ConsumerSize": 5,
        "DataOffset": 0,
        "TimeType": 1,
        "MonitorConfig": {
          "TimeoutEnable": true,
          "Timeout": 12300,
          "TimeUnit": "second"
        }
      }
    }
  }
}

```

```
        "FailEnable": true,
        "SendChannel": "sms",
        "TimeoutKillEnable": true
    }
},
"Name": "helloworld",
"MapTaskXAttrs": {
    "TaskAttemptInterval": 0,
    "QueueSize": 10000,
    "PageSize": 100,
    "DispatcherSize": 5,
    "TaskMaxAttempt": 0,
    "Calendar": "工作日",
    "TimeExpression": "0 0/10 * * * ?",
    "ContactInfo": {
        "UserName": "userA",
        "UserPhone": "1381111****"
    },
    "ConsumerSize": 5,
    "DataOffset": 0,
    "TimeType": 1,
    "MonitorConfig": {
        "TimeoutEnable": true,
        "Timeout": 12300,
        "FailEnable": true,
        "SendChannel": "sms",
        "TimeoutKillEnable": true
    }
},
"JobMonitorInfo": {
    "TaskAttemptInterval": 0,
    "QueueSize": 10000,
    "PageSize": 100,
    "DispatcherSize": 5,
    "TaskMaxAttempt": 0,
    "Calendar": "工作日",
    "TimeExpression": "0 0/10 * * * ?",
    "ContactInfo": {
        "UserName": "userA",
        "UserPhone": "1381111****"
    },
},
```

```
    "ConsumerSize": 5,
    "DataOffset": 0,
    "TimeType": 1,
    "MonitorConfig": {
      "TimeoutEnable": true,
      "Timeout": 12300,
      "FailEnable": true,
      "SendChannel": "sms",
      "TimeoutKillEnable": true
    }
  },
  "Content": "echo \"clear\" &gt; /home/admin/edas-container/logs/catalina.out",
  "ClassName": "com.alibaba.test.helloworld",
  "AttemptInterval": 30,
  "ExecuteMode": "standalone"
}
},
"Code": 200,
"Success": true
}
```

# 7.UpdateJob

调用UpdateJob更新任务配置信息。默认先调用获取任务找到对应字段进行修改。

## 请求参数

名称	类型	是否必选	示例值	描述
ExecuteMode	String	是	standalone	任务执行模式，目前支持以下几种任务执行模式： <ul style="list-style-type: none"> <li>• 单机运行：standalone</li> <li>• 广播运行：broadcast</li> <li>• 并行计算：parallel</li> <li>• 内存网格：grid</li> <li>• 网格计算：batch</li> <li>• 分片运行：shard</li> </ul>
GroupId	String	是	testSchedulerx.defaultGroup	应用ID，在控制台的应用管理页面中获取。
JobId	Long	是	92583	任务ID，在控制台的任务管理页面中获取。
Name	String	是	helloworld	任务名。
Namespace	String	是	adcfc35d-e2fe-4fe9-bbaa-20e90ffc****	命名空间ID，在控制台的命名空间页面中获取。
RegionId	String	是	cn-hangzhou	地域ID。
TimeType	Integer	是	1	时间配置类型，目前支持以下几种配置类型： <ul style="list-style-type: none"> <li>• cron: 1</li> <li>• fix_rate: 3</li> <li>• second_delay: 4</li> <li>• api: 100</li> </ul>
Description	String	否	test	任务描述。

名称	类型	是否必选	示例值	描述
Content	String	否	echo 'hello'	任务类型为python、shell、go的必填脚本内容。
Parameters	String	否	test	用户自定义参数，运行时可以获取。
MaxConcurrency	Integer	否	1	最大同时运行实例数量，默认值为1，即上次触发没有运行结束，不会进行下次触发即使到了运行时刻。
MaxAttempt	Integer	否	0	错误最大重试次数，根据业务需求填写。
AttemptInterval	Integer	否	30	错误重试间隔时间，单位s，默认值为30。
ClassName	String	否	com.alibaba.test.helloworld	任务接口类完整路径。 是Java任务类型时，才有该字段且必须填写完整路径。
JarUrl	String	否	暂不支持，不用填写	上传到OSS的完整路径。 如果选择JAR包运行，可以将相应JAR包上传到OSS的该路径下。
PageSize	Integer	否	100	并行网格任务高级配置，单次拉取子任务数，默认值为100。
ConsumerSize	Integer	否	5	并行网格任务高级配置，单机单次触发执行线程数，默认值为5。
QueueSize	Integer	否	10000	并行网格任务高级配置，子任务队列缓存上限，默认值为10000。
DispatcherSize	Integer	否	5	并行网格任务高级配置，子任务分发线程数，默认值为5。



名称	类型	是否必选	示例值	描述
TimeExpression	String	否	30	时间表达式，根据选择的时间类型设置时间表达式。 <ul style="list-style-type: none"> <li>• <b>cron</b>: 填写标准的cron表达式，支持在线验证。</li> <li>• <b>api</b>: 无时间表达式。</li> <li>• <b>fixed_rate</b>: 填写具体固定频率值，单位s。如30表示每隔30s触发一次。</li> <li>• <b>second_delay</b>: 填写固定延迟多少秒执行一次（1s~60s可选）。</li> </ul>
Calendar	String	否	工作日	cron类型可以选择填写自定义日历。
DataOffset	Integer	否	2400	cron类型可以选择时间偏移，单位s。
TimeoutEnable	Boolean	否	true	超时报警开关。取值如下： <ul style="list-style-type: none"> <li>• <b>true</b>: 开启超时报警开关。</li> <li>• <b>false</b>: 关闭超时报警开关。</li> </ul>
Timeout	Long	否	7200	超时阈值，单位s。
TimeoutKillEnable	Boolean	否	true	超时终止本次触发开关。取值如下： <ul style="list-style-type: none"> <li>• <b>true</b>: 开启超时终止开关。</li> <li>• <b>false</b>: 关闭超时终止开关。</li> </ul>
FailEnable	Boolean	否	true	失败报警开关。取值如下： <ul style="list-style-type: none"> <li>• <b>true</b>: 开启失败报警开关。</li> <li>• <b>false</b>: 关闭失败报警开关。</li> </ul>
SendChannel	String	否	sms	报警发送形式，目前只支持sms。
ContactInfo.N.UserName	String	否	userA	用户名。
ContactInfo.N.UserPhone	String	否	1381111****	用户手机号。
NamespaceSource	String	否	schedulerx	特殊第三方才需要填写。
TaskMaxAttempt	Integer	否	0	并行网格任务高级配置，子任务失败重试次数。

名称	类型	是否必选	示例值	描述
TaskAttemptInterval	Integer	否	0	并行网格任务高级配置，子任务失败重试间隔。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码
Message	String	job type is java className can not be blank	错误信息，仅出错时返回错误信息。
RequestId	String	4F68ABED-AC31-4412-9297-D9A8F0401108	请求唯一ID
Success	Boolean	true	是否成功

## 示例

### 请求示例

```

http(s)://[Endpoint]/?Action=UpdateJob
&ExecuteMode=standalone
&GroupId=testSchedulerx.defaultGroup
&JobId=92583
&Name=helloworld
&Namespace=adcfc35d-e2fe-4fe9-bbaa-20e90ffc****
&RegionId=cn-hangzhou
&TimeType=1
&<公共请求参数>

```

### 正常返回示例

#### XML 格式

```

<UpdateJobResponse>
  <RequestId>4F68ABED-AC31-4412-9297-D9A8F0401108</RequestId>
  <Message>job type is java className can not be blank</Message>
  <Code>200</Code>
  <Success>true</Success>
</UpdateJobResponse>

```

## JSON 格式

```
{
  "RequestId": "4F68ABED-AC31-4412-9297-D9A8F0401108",
  "Message": "job type is java className can not be blank",
  "Code": 200,
  "Success": true
}
```

## 8.DeleteJob

调用DeleteJob删除指定任务。

### 请求参数

名称	类型	是否必选	示例值	描述
GroupId	String	是	testSchedulerx.defaultGroup	应用ID，在控制台的的应用管理页面中获取。
JobId	Long	是	92583	任务ID，在控制台的任务管理页面中获取。
Namespace	String	是	adcf35d-e2fe-4fe9-bbaa-20e90ffc****	命名空间ID，在控制台的命名空间页面中获取。
RegionId	String	是	cn-hangzhou	地域ID。
NamespaceSource	String	否	schedulerx	特殊第三方才需要填写。

### 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码
Message	String	jobid: 92583 not match groupId: testSchedulerx.defaultGroup	错误消息，仅出错时返回错误信息。
RequestId	String	4F68ABED-AC31-4412-9297-D9A8F0401108	请求唯一ID
Success	Boolean	true	是否成功

### 示例

请求示例

```
http(s)://[Endpoint]/?Action=DeleteJob
&GroupId=testSchedulerx.defaultGroup
&JobId=92583
&Namespace=adcf35d-e2fe-4fe9-bbaa-20e90ffc****
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<DeleteJobResponse>
  <RequestId>4F68ABED-AC31-4412-9297-D9A8F0401108</RequestId>
  <Message>jobid: 92583 not match groupId: testSchedulerx.defaultGroup</Message>
  <Code>200</Code>
  <Success>>true</Success>
</DeleteJobResponse>
```

#### JSON 格式

```
{
  "RequestId": "4F68ABED-AC31-4412-9297-D9A8F0401108",
  "Message": "jobid: 92583 not match groupId: testSchedulerx.defaultGroup",
  "Code": 200,
  "Success": true
}
```

## 9.EnableJob

调用EnableJob启用指定任务。任务创建完成以后默认启用，所以该功能是在停用任务后使用。

### 请求参数

名称	类型	是否必选	示例值	描述
JobId	Long	是	92583	任务ID，在控制台的任务管理页面中获取。
Namespace	String	是	adcfc35d-e2fe-4fe9-bbaa-20e90ffc****	命名空间ID，在控制台的命名空间页面中获取。
RegionId	String	是	cn-hangzhou	地域ID。
NamespaceSource	String	否	schedulerx	特殊第三方才需要填写。
GroupId	String	否	testSchedulerx.defaultGroup	应用ID，在控制台的的应用管理页面中获取。

### 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码
Message	String	jobid: 92583 not match groupId: testSchedulerx.defaultGroup	错误信息，仅出错时返回错误信息。
RequestId	String	71BCC0E3-64B2-4B63-A870-AFB64EBCB5A7	请求唯一ID
Success	Boolean	true	调用任务是否成功。取值如下： <ul style="list-style-type: none"> <li>• true：调用任务成功。</li> <li>• false：调用任务失败。</li> </ul>

### 示例

请求示例

```
http(s)://[Endpoint]/?Action=EnableJob
&JobId=92583
&Namespace=adfc35d-e2fe-4fe9-bbaa-20e90ffc****
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<EnableJobResponse>
  <RequestId>71BCC0E3-64B2-4B63-A870-AFB64EBCB5A7</RequestId>
  <Message>jobid: 92583 not match groupId: testSchedulerx.defaultGroup</Message>
  <Code>200</Code>
  <Success>>true</Success>
</EnableJobResponse>
```

#### JSON 格式

```
{
  "RequestId": "71BCC0E3-64B2-4B63-A870-AFB64EBCB5A7",
  "Message": "jobid: 92583 not match groupId: testSchedulerx.defaultGroup",
  "Code": 200,
  "Success": true
}
```

## Demo

```
package com.alibaba.schedulerx.pop;

import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.schedulerx2.model.v20190430.EnableJobRequest;
import com.aliyuncs.schedulerx2.model.v20190430.EnableJobResponse;

public class EnableJob {

    public static void main(String[] args) throws Exception {
        // OpenAPI的接入点，具体查看上表支持地域列表以及购买机器地域填写。
        String regionId = "cn-shanghai";
        //鉴权使用的 AccessKeyId，由阿里云官网控制台获取。
        String accessKeyId = "<yourAccessKeyId>";
        //鉴权使用的 AccessKeySecret，由阿里云官网控制台获取。
        String accessKeySecret = "<yourAccessKeySecret>";
        //产品名称。
        String productName = "schedulerx2";
        //对照支持地域列表选择Domain填写。
        String domain = "schedulerx.cn-shanghai.aliyuncs.com";
        //构建OpenAPI客户端。
        DefaultProfile.addEndpoint(regionId, productName, domain);
        DefaultProfile defaultProfile = DefaultProfile.getProfile(regionId, accessKeyId, accessKeySecret);
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);

        EnableJobRequest request = new EnableJobRequest();
        request.setNamespace("xxxxxx");
        request.setGroupId("xxxxxx");
        request.setJobId(123L);
        EnableJobResponse response = client.getAcsResponse(request);
        if (!response.getSuccess()) {
            System.out.println(response.getMessage());
            System.out.println("EnableJob: "+response.getRequestId());
        } else {
            System.out.println(response.getMessage());
        }
    }
}
```



# 10.DisableJob

调用DisableJob停用指定任务。

## 请求参数

名称	类型	是否必选	示例值	描述
JobId	Long	是	92583	任务ID，在控制台的任务管理页面中获取。
Namespace	String	是	adcfc35d-e2fe-4fe9-bbaa-20e90ffc****	命名空间ID，在控制台的命名空间页面中获取。
RegionId	String	是	cn-hangzhou	地域ID。
NamespaceSource	String	否	schedulerx	特殊第三方才需要填写。
GroupId	String	否	testSchedulerx.defaultGroup	应用ID，在控制台的应用管理页面中获取。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码
Message	String	jobid: 92583 not match groupid: testSchedulerx.defaultGroup	错误消息，仅出错时返回错误信息。
RequestId	String	C8E5FB4A-6D8D-424D-9AAA-4FE06BB74FF9	请求唯一ID
Success	Boolean	true	禁用任务是否成功。取值如下： <ul style="list-style-type: none"> <li>• true：禁用任务成功。</li> <li>• false：禁用任务失败。</li> </ul>

## 示例

请求示例

```
http(s)://[Endpoint]/?Action=DisableJob
&JobId=92583
&Namespace=adcfc35d-e2fe-4fe9-bbaa-20e90ffc****
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<DisableJobResponse>
  <RequestId>C8E5FB4A-6D8D-424D-9AAA-4FE06BB74FF9</RequestId>
  <Message>jobid: 92583 not match groupId: testSchedulerx.defaultGroup</Message>
  <Code>200</Code>
  <Success>>true</Success>
</DisableJobResponse>
```

#### JSON 格式

```
{
  "RequestId": "C8E5FB4A-6D8D-424D-9AAA-4FE06BB74FF9",
  "Message": "jobid: 92583 not match groupId: testSchedulerx.defaultGroup",
  "Code": 200,
  "Success": true
}
```

## Demo

```
package com.alibaba.schedulerx.pop;

import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.schedulerx2.model.v20190430.DisableJobRequest;
import com.aliyuncs.schedulerx2.model.v20190430.DisableJobResponse;


public class DisableJob {

    public static void main(String[] args) throws Exception {
        // OpenAPI的接入点，具体查看上表支持地域列表以及购买机器地域填写。
        String regionId = "cn-shanghai";
        //鉴权使用的AccessKeyId，由阿里云官网控制台获取。
        String accessKeyId = "<yourAccessKeyId>";
        //鉴权使用的AccessKeySecret，由阿里云官网控制台获取。
        String accessKeySecret = "<yourAccessKeySecret>";
        //产品名称。
        String productName = "schedulerx2";
        //对照支持地域列表选择Domain填写。
        String domain = "schedulerx.cn-shanghai.aliyuncs.com";
        //构建OpenAPI客户端。
        DefaultProfile.addEndpoint(regionId, productName, domain);
        DefaultProfile defaultProfile = DefaultProfile.getProfile(regionId, accessKeyId, accessKeySecret);
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);

        DisableJobRequest request = new DisableJobRequest();
        request.setNamespace("xxxxxx");
        request.setGroupId("xxxxxx");
        request.setJobId(123L);
        DisableJobResponse response = client.getAcsResponse(request);
        if (!response.isSuccess()) {
            System.out.println(response.getMessage());
            System.out.println("DisableJob: "+response.getRequestId());
        } else {
            System.out.println(response.getMessage());
        }
    }
}
```

# 11.ExecuteJob

调用ExecuteJob触发一次任务。API类型任务触发需要调用方法，非API类型也可以调用。

 **说明** 因为jobid+scheduleTime是唯一索引，所以同一个任务连续调用ExecuteJob接口的时候，每次要sleep一秒，不然有可能会失败。

## 请求参数

名称	类型	是否必选	示例值	描述
GroupId	String	是	testSchedulerx.defaultGroup	应用ID，在控制台的应用管理页面中获取。
InstanceParameters	String	是	test	本次触发携带参数，可以是任意字符串，processor代码通过context.getInstanceParameters()获取，区别于创建任务自定义参数。
JobId	Long	是	92583	任务ID，在控制台的任务管理页面中获取。
Namespace	String	是	adcfc35d-e2fe-4fe9-bbaa-20e90ffc****	命名空间ID，在控制台的命名空间页面中获取。
RegionId	String	是	cn-hangzhou	地域ID。
NamespaceSource	String	否	schedulerx	特殊第三方才需要填写。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码
Data	Struct		如果成功，会返回任务实例ID。
JobInstanceId	Long	11111111	任务实例ID

名称	类型	示例值	描述
Message	String	groupid not exist groupid: testSchedulerx.def aultGroup namespace: adcf35d-e2fe- 4fe9-bbaa- 20e90ffc****	错误消息，仅出错时返回错误信息。
RequestId	String	4F68ABED-AC31- 4412-9297- D9A8F0401108	请求唯一ID
Success	Boolean	true	触发任务是否成功。取值如下： <ul style="list-style-type: none"> <li>• true：触发任务成功。</li> <li>• false：触发任务失败。</li> </ul>

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=ExecuteJob
&GroupId=testSchedulerx.defaultGroup
&InstanceParameters=test
&JobId=92583
&Namespace=adcf35d-e2fe-4fe9-bbaa-20e90ffc****
&RegionId=cn-hangzhou
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<ExecuteJobResponse>
  <RequestId>4F68ABED-AC31-4412-9297-D9A8F0401108</RequestId>
  <Message>groupid not exist groupId: testSchedulerx.defaultGroup namespace: adcf35d-e2fe-4fe9-  
bbaa-20e90ffc****</Message>
  <Data>
    <JobInstanceId>11111111</JobInstanceId>
  </Data>
  <Code>200</Code>
  <Success>>true</Success>
</ExecuteJobResponse>
```

## JSON 格式

```
{
  "RequestId": "4F68ABED-AC31-4412-9297-D9A8F0401108",
  "Message": "groupid not exist groupId: testSchedulerx.defaultGroup namespace: adafc35d-e2fe-4fe9-bbaa-20e90ffc****",
  "Data": {
    "JobInstanceId": 11111111
  },
  "Code": 200,
  "Success": true
}
```

# 12.StopInstance

调用StopInstance终止某次正在运行的实例。

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	StopInstance	系统规定参数。取值：StopInstance。
GroupId	String	是	testSchedulerx.defaultGroup	应用ID，在控制台的应用管理页面中获取。
InstanceId	Long	是	11111111	运行实例ID。
JobId	Long	是	92583	任务ID，在控制台的任务管理页面中获取。
Namespace	String	是	adcfc35d-e2fe-4fe9-bbaa-20e90ffc****	命名空间ID，在控制台的命名空间页面中获取。
RegionId	String	是	cn-hangzhou	地域ID。
NamespaceSource	String	否	schedulerx	特殊第三方才需要填写。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码
Message	String	Your request is denied as lack of ssl protect.	错误信息，仅错误时返回错误信息。
RequestId	String	4F68ABED-AC31-4412-9297-D9A8F0401108	请求唯一ID
Success	Boolean	true	终止任务运行是否成功。取值如下： <ul style="list-style-type: none"><li>• true：终止任务运行成功。</li><li>• false：终止任务运行失败。</li></ul>

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=StopInstance
&GroupId=testSchedulerx.defaultGroup
&InstanceId=11111111
&JobId=92583
&Namespace=adcfc35d-e2fe-4fe9-bbaa-20e90ffc****
&RegionId=cn-hangzhou
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<StopInstanceResponse>
  <RequestId>4F68ABED-AC31-4412-9297-D9A8F0401108</RequestId>
  <Message>Your request is denied as lack of ssl protect.</Message>
  <Code>200</Code>
  <Success>>true</Success>
</StopInstanceResponse>
```

#### JSON 格式

```
{
  "RequestId": "4F68ABED-AC31-4412-9297-D9A8F0401108",
  "Message": "Your request is denied as lack of ssl protect.",
  "Code": 200,
  "Success": true
}
```



# 13. GetJobInstanceList

调用GetJobInstanceList获取指定任务ID的执行实例列表。该接口只返回最近10条运行实例列表。

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetJobInstanceList	系统规定参数。取值： GetJobInstanceList。
GroupId	String	是	testSchedulerx.defaultGroup	应用ID，在控制台的应用管理页面中获取。
JobId	Long	是	92583	任务ID，在控制台的任务管理页面中获取。
Namespace	String	是	adcfc35d-e2fe-4fe9-bbaa-20e90ffc****	命名空间ID，在控制台的命名空间页面中获取。
RegionId	String	是	cn-hangzhou	地域ID。
NamespaceSource	String	否	schedulerx	特殊第三方才需要填写。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码
Data	Struct		instance列表
JobInstanceDetails	Array of JobInstanceDetails		任务实例详细信息
DateTime	String	2020-07-27 11:52:10	数据时间
EndTime	String	2020-07-27 11:52:10	任务执行结束时间
Executor	String	A	执行人

名称	类型	示例值	描述
InstanceId	Long	11111111	实例ID
JobId	Long	92583	任务ID
Progress	String	complete	实例进度详情
Result	String	success	实例执行结果
ScheduleTime	String	2020-07-27 11:52:10	任务调度时间
StartTime	String	2020-07-27 11:52:10	任务执行开始时间
Status	Integer	4	实例状态。包含以下几种状态： <ul style="list-style-type: none"> <li>• 1: 等待</li> <li>• 3: 运行中</li> <li>• 4: 成功</li> <li>• 5: 失败</li> <li>• 9: 拒绝</li> </ul> 对应枚举类： com.alibaba.schedulerx.common.domain.InstanceStatus
TimeType	Integer	1	实例调度时间类型，包含以下几种类型： <ul style="list-style-type: none"> <li>• cron: 1</li> <li>• fix_rate: 3</li> <li>• second_delay: 4</li> <li>• api: 100</li> </ul> 对应枚举类： com.alibaba.schedulerx.common.domain.TimeType

名称	类型	示例值	描述
TriggerType	Integer	3	触发类型，包含以下几种触发类型： <ul style="list-style-type: none"> <li>1: 定时调度正常触发</li> <li>2: 数据重刷</li> <li>3: API触发</li> <li>4: 用户手动点击重跑</li> <li>5: 系统重试（系统异常，如DB异常）</li> </ul> 对应枚举类： com.alibaba.schedulerx.common.domain.TriggerType
WorkAddr	String	192.168.0.0:16	被触发客户端IP:Port
Message	String	jobid: 92583 not match groupid: testSchedulerx.defaultGroup	错误信息，仅出错时返回错误信息。
RequestId	String	4F68ABED-AC31-4412-9297-D9A8F0401108	请求唯一ID
Success	Boolean	true	获取任务的执行实例列表是否成功。取值如下： <ul style="list-style-type: none"> <li>true: 获取任务的执行实例列表成功。</li> <li>false: 获取任务的执行实例列表失败。</li> </ul>

## 示例

### 请求示例

```

http(s)://[Endpoint]/?Action=GetJobInstanceList
&GroupId=testSchedulerx.defaultGroup
&JobId=92583
&Namespace=adcfc35d-e2fe-4fe9-bbaa-20e90ffc****
&RegionId=cn-hangzhou
&<公共请求参数>
    
```

### 正常返回示例

XML 格式

```
<GetJobInstanceListResponse>
  <RequestId>4F68ABED-AC31-4412-9297-D9A8F0401108</RequestId>
  <Message>jobid: 92583 not match groupid: testSchedulerx.defaultGroup</Message>
  <Data>
    <JobInstanceDetails>
      <Status>4</Status>
      <TriggerType>3</TriggerType>
      <Progress>complete</Progress>
      <EndTime>2020-07-27 11:52:10</EndTime>
      <InstanceId>11111111</InstanceId>
      <WorkAddr>192.168.0.0:16</WorkAddr>
      <StartTime>2020-07-27 11:52:10</StartTime>
      <DataTime>2020-07-27 11:52:10</DataTime>
      <Result>success</Result>
      <TimeType>1</TimeType>
      <Executor>A</Executor>
      <ScheduleTime>2020-07-27 11:52:10</ScheduleTime>
      <JobId>92583</JobId>
    </JobInstanceDetails>
  </Data>
  <Code>200</Code>
  <Success>true</Success>
</GetJobInstanceListResponse>
```

JSON 格式

```
{
  "RequestId": "4F68ABED-AC31-4412-9297-D9A8F0401108",
  "Message": "jobid: 92583 not match groupId: testSchedulerx.defaultGroup",
  "Data": {
    "JobInstanceDetails": {
      "Status": 4,
      "TriggerType": 3,
      "Progress": "complete",
      "EndTime": "2020-07-27 11:52:10",
      "InstanceId": 11111111,
      "WorkAddr": "192.168.0.0:16",
      "StartTime": "2020-07-27 11:52:10",
      "DataTime": "2020-07-27 11:52:10",
      "Result": "success",
      "TimeType": 1,
      "Executor": "A",
      "ScheduleTime": "2020-07-27 11:52:10",
      "JobId": 92583
    }
  },
  "Code": 200,
  "Success": true
}
```

# 14. GetJobInstance

调用GetJobInstance获取指定任务实例详情。

## 请求参数

名称	类型	是否必选	示例值	描述
GroupId	String	是	testSchedulerx.defaultGroup	应用ID，在控制台的应用管理页面中获取。
JobId	Long	是	92583	任务ID。
JobInstanceId	Long	是	11111111	实例ID。
Namespace	String	是	adcfc35d-e2fe-4fe9-bbaa-20e90ffc****	命名空间ID，在控制台的命名空间页面中获取。
NamespaceSource	String	否	schedulerx	特殊第三方才需要填写。
RegionId	String	否	cn-hangzhou	地域ID。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码
Data	Struct		instance信息
JobInstanceDetail	Struct		任务实例详细信息
DateTime	String	2020-07-27 11:52:10	数据时间
EndTime	String	2020-07-27 11:52:10	任务执行结束时间
Executor	String	A	执行人
InstanceId	Long	11111111	实例ID

名称	类型	示例值	描述
JobId	Long	92583	任务ID
Progress	String	complete	实例进度详情
Result	String	success	实例执行结果
ScheduleTime	String	2020-07-27 11:52:10	任务调度时间
StartTime	String	2020-07-27 11:52:10	任务执行开始时间
Status	Integer	4	实例状态。包含以下几种状态： <ul style="list-style-type: none"> <li>• 1: 等待</li> <li>• 3: 运行中</li> <li>• 4: 成功</li> <li>• 5: 失败</li> <li>• 9: 拒绝</li> </ul> 对应枚举类： com.alibaba.schedulerx.common.domain.InstanceStatus
TimeType	Integer	1	实例调度时间类型，包含以下几种时间类型： <ul style="list-style-type: none"> <li>• cron: 1</li> <li>• fix_rate: 3</li> <li>• second_delay: 4</li> <li>• api: 100</li> </ul> 对应枚举类： com.alibaba.schedulerx.common.domain.TimeType
TriggerType	Integer	3	触发类型，包含以下几种触发类型： <ul style="list-style-type: none"> <li>• 1: 定时调度正常触发</li> <li>• 2: 数据重刷</li> <li>• 3: API触发</li> <li>• 4: 用户手动点击重跑</li> <li>• 5: 系统重试（系统异常，如DB异常）</li> </ul> 对应枚举类： com.alibaba.schedulerx.common.domain.TriggerType

名称	类型	示例值	描述
WorkAddr	String	192.168.0.0:16	被触发客户端IP:Port
Message	String	jobid: 92583 not match groupid: testSchedulerx.defaultGroup	错误消息，仅出错时返回错信息。
RequestId	String	4F68ABED-AC31-4412-9297-D9A8F0401108	请求唯一ID
Success	Boolean	true	获取任务实例详情是否成功。取值如下： <ul style="list-style-type: none"><li>• true：获取任务实例详情成功。</li><li>• false：获取任务实例详情失败。</li></ul>

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=GetJobInstance
&GroupId=testSchedulerx.defaultGroup
&JobId=92583
&JobInstanceId=11111111
&Namespace=adcfc35d-e2fe-4fe9-bbaa-20e90ffc****
&<公共请求参数>
```

### 正常返回示例

XML 格式



```
<GetJobInstanceResponse>
  <RequestId>4F68ABED-AC31-4412-9297-D9A8F0401108</RequestId>
  <Message>jobid: 92583 not match groupid: testSchedulerx.defaultGroup</Message>
  <Data>
    <JobInstanceDetail>
      <Status>4</Status>
      <TriggerType>3</TriggerType>
      <Progress>complete</Progress>
      <EndTime>2020-07-27 11:52:10</EndTime>
      <InstanceId>11111111</InstanceId>
      <WorkAddr>192.168.0.0:16</WorkAddr>
      <StartTime>2020-07-27 11:52:10</StartTime>
      <DataTime>2020-07-27 11:52:10</DataTime>
      <Result>success</Result>
      <TimeType>1</TimeType>
      <Executor>A</Executor>
      <ScheduleTime>2020-07-27 11:52:10</ScheduleTime>
      <JobId>92583</JobId>
    </JobInstanceDetail>
  </Data>
  <Code>200</Code>
  <Success>true</Success>
</GetJobInstanceResponse>
```

JSON 格式

```
{
  "RequestId": "4F68ABED-AC31-4412-9297-D9A8F0401108",
  "Message": "jobid: 92583 not match groupId: testSchedulerx.defaultGroup",
  "Data": {
    "JobInstanceDetail": {
      "Status": 4,
      "TriggerType": 3,
      "Progress": "complete",
      "EndTime": "2020-07-27 11:52:10",
      "InstanceId": 11111111,
      "WorkAddr": "192.168.0.0:16",
      "StartTime": "2020-07-27 11:52:10",
      "DataTime": "2020-07-27 11:52:10",
      "Result": "success",
      "TimeType": 1,
      "Executor": "A",
      "ScheduleTime": "2020-07-27 11:52:10",
      "JobId": 92583
    }
  },
  "Code": 200,
  "Success": true
}
```

# 15.ExecuteWorkflow

调用ExecuteWorkflow触发一次工作流。API类型工作流触发需要调用方法，非API类型也可以调用。

## 请求参数

名称	类型	是否必选	示例值	描述
GroupId	String	是	testSchedulerx.defaultGroup	应用ID，在控制台的应用管理页面中获取。
InstanceParameters	String	是	test	工作流实例动态参数，不超过1000字节。
Namespace	String	是	adcf35d-e2fe-4fe9-bbaa-20e90ffc****	命名空间ID，在控制台的命名空间页面中获取。
WorkflowId	Long	是	111	工作流ID。
NamespaceSource	String	否	schedulerx	特殊第三方才需要填写。
RegionId	String	否	cn-hangzhou	地域ID。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码
Data	Struct		如果调用成功，会返回流程实例ID。
WfInstanceId	Long	111111	工作流实例ID
Message	String	Cannot find product according to your domain.	错误消息，仅错误时返回错误信息。
RequestId	String	4F68ABED-AC31-4412-9297-D9A8F0401108	请求唯一ID
Success	Boolean	true	是否成功

## 示例

## 请求示例

```
http(s)://[Endpoint]/?Action=ExecuteWorkflow
&GroupId=testSchedulerx.defaultGroup
&InstanceParameters=test
&Namespace=adfc35d-e2fe-4fe9-bbaa-20e90ffc****
&WorkflowId=111
&<公共请求参数>
```

## 正常返回示例

## XML 格式

```
<ExecuteWorkflowResponse>
  <RequestId>4F68ABED-AC31-4412-9297-D9A8F0401108</RequestId>
  <Message>Cannot find product according to your domain.</Message>
  <Data>
    <WfInstanceId>111111</WfInstanceId>
  </Data>
  <Code>200</Code>
  <Success>>true</Success>
</ExecuteWorkflowResponse>
```

## JSON 格式

```
{
  "RequestId": "4F68ABED-AC31-4412-9297-D9A8F0401108",
  "Message": "Cannot find product according to your domain.",
  "Data": {
    "WfInstanceId": 111111
  },
  "Code": 200,
  "Success": true
}
```

# 16.EnableWorkflow

调用EnableWorkflow启用指定 workflow。

## 请求参数

名称	类型	是否必选	示例值	描述
Namespace	String	是	adcf35d-e2fe-4fe9-bbaa-20e90ffc****	命名空间ID，在控制台的命名空间页面中获取。
RegionId	String	是	cn-hangzhou	地域ID。
WorkflowId	Long	是	111	workflow ID。
NamespaceSource	String	否	schedulervx	特殊第三方才需要填写。
GroupId	String	否	testSchedulerx.defaultGroup	应用ID，在控制台的应用管理页面中获取。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码
Message	String	Your request is denied as lack of ssl protect.	错误信息，仅错误时返回错误信息。
RequestId	String	4F68ABED-AC31-4412-9297-D9A8F0401108	请求唯一ID
Success	Boolean	true	启用 workflow 是否成功。取值如下： <ul style="list-style-type: none"> <li>• true: 启用 workflow 成功。</li> <li>• false: 启用 workflow 失败。</li> </ul>

## 示例

请求示例

```
http(s)://[Endpoint]/?Action=EnableWorkflow
&Namespace=adfc35d-e2fe-4fe9-bbaa-20e90ffc****
&WorkflowId=111
&<公共请求参数>
```

#### 正常返回示例

##### XML 格式

```
<EnableWorkflowResponse>
  <RequestId>4F68ABED-AC31-4412-9297-D9A8F0401108</RequestId>
  <Message>Your request is denied as lack of ssl protect.</Message>
  <Code>200</Code>
  <Success>>true</Success>
</EnableWorkflowResponse>
```

##### JSON 格式

```
{
  "RequestId": "4F68ABED-AC31-4412-9297-D9A8F0401108",
  "Message": "Your request is denied as lack of ssl protect.",
  "Code": 200,
  "Success": true
}
```

# 17.DisableWorkflow

调用DisableWorkflow禁用指定 workflow。

## 请求参数

名称	类型	是否必选	示例值	描述
Namespace	String	是	adcf35d-e2fe-4fe9-bbaa-20e90ffc****	命名空间ID，在控制台的命名空间页面中获取。
RegionId	String	是	cn-hangzhou	地域ID。
WorkflowId	Long	是	111	workflow ID。
NamespaceSource	String	否	schedulerx	特殊第三方才需要填写。
GroupId	String	否	testSchedulerx.defaultGroup	应用ID，在控制台的应用管理页面中获取。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码
Message	String	Your request is denied as lack of ssl protect.	错误消息，仅错误时返回错误信息。
RequestId	String	4F68ABED-AC31-4412-9297-D9A8F0401108	请求唯一ID
Success	Boolean	true	禁用 workflow 是否成功。取值如下： <ul style="list-style-type: none"><li>• true: 禁用 workflow 成功。</li><li>• false: 禁用 workflow 失败。</li></ul>

## 示例

请求示例

```
http(s)://[Endpoint]/?Action=DisableWorkflow
&Namespace=adfc35d-e2fe-4fe9-bbaa-20e90ffc****
&WorkflowId=111
&<公共请求参数>
```

#### 正常返回示例

##### XML 格式

```
<DisableWorkflowResponse>
  <RequestId>4F68ABED-AC31-4412-9297-D9A8F0401108</RequestId>
  <Message>Your request is denied as lack of ssl protect.</Message>
  <Code>200</Code>
  <Success>>true</Success>
</DisableWorkflowResponse>
```

##### JSON 格式

```
{
  "RequestId": "4F68ABED-AC31-4412-9297-D9A8F0401108",
  "Message": "Your request is denied as lack of ssl protect.",
  "Code": 200,
  "Success": true
}
```



# 18.DeleteWorkflow

调用DeleteWorkflow删除指定工作流。

## 请求参数

名称	类型	是否必选	示例值	描述
Namespace	String	是	adcf35d-e2fe-4fe9-bbaa-20e90ffc****	命名空间ID，在控制台的命名空间页面中获取。
RegionId	String	是	cn-hangzhou	地域ID。
WorkflowId	Long	是	111	工作流ID。
NamespaceSource	String	否	schedulervx	特殊第三方才需要填写。
GroupId	String	否	testSchedulerx.defaultGroup	应用ID，在控制台的应用管理页面中获取。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码
Message	String	Your request is denied as lack of ssl protect.	错误消息，仅错误时返回错误信息。
RequestId	String	4F68ABED-AC31-4412-9297-D9A8F0401108	请求唯一ID
Success	Boolean	true	删除工作流是否成功。取值如下： <ul style="list-style-type: none"> <li>• true：删除工作流成功。</li> <li>• false：删除工作流失败。</li> </ul>

## 示例

请求示例

```
http(s)://[Endpoint]/?Action=DeleteWorkflow
&Namespace=adfc35d-e2fe-4fe9-bbaa-20e90ffc****
&WorkflowId=111
&<公共请求参数>
```

#### 正常返回示例

##### XML 格式

```
<DeleteWorkflowResponse>
  <RequestId>4F68ABED-AC31-4412-9297-D9A8F0401108</RequestId>
  <Message>Your request is denied as lack of ssl protect.</Message>
  <Code>200</Code>
  <Success>>true</Success>
</DeleteWorkflowResponse>
```

##### JSON 格式

```
{
  "RequestId": "4F68ABED-AC31-4412-9297-D9A8F0401108",
  "Message": "Your request is denied as lack of ssl protect.",
  "Code": 200,
  "Success": true
}
```