

腾讯云批量计算

SDK 文档

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
SDK 文档.....	4
Java SDK	4
计算环境接口文档.....	4
作业接口文档.....	9
Python SDK.....	15
计算环境接口文档	15
作业接口文档.....	20

SDK 文档

Java SDK

计算环境接口文档

开发准备

- 下载和安装 [Java SDK](#)。
- 首次使用批量计算，参考 [开始前的准备](#)。
- 了解更多计算环境配置参数，参考 [创建计算环境 API 文档](#)。

快速入门

```
import java.util.TreeMap;
```

```
import com.qcloud.QcloudApiModuleCenter;
```

```
import com.qcloud.Module.Batch;
```

```
import com.qcloud.Utilities.Json.JSONObject;
```

```
public class BatchDemo {
```

```
    public static void main(String[] args) {
```

```
        TreeMap<String, Object> config = new TreeMap<String, Object>();
```

```
        config.put("SecretId", "您的SecretId");
```

```
        config.put("SecretKey", "您的SecretKey");
```

```
        config.put("RequestMethod", "GET");
```

```
        config.put("DefaultRegion", "gz"); // 地域, gz: guangzhou
```

```
        QcloudApiModuleCenter module = new QcloudApiModuleCenter(new Batch(), config);
```

```
    }
```

```
}
```

创建计算环境

```
TreeMap<String, Object> envParams = new TreeMap<String, Object>();
envParams.put("ComputeEnv.EnvName", "batch-env"); // 计算环境名称
envParams.put("ComputeEnv.EnvType", "MANAGED");
envParams.put("ComputeEnv.DesiredComputeNodeCount", 10); // 期望节点数目
envParams.put("ComputeEnv.EnvData.InstanceType", "S1.SMALL1"); // 实例类型
envParams.put("ComputeEnv.EnvData.ImageId", "img-er9shcln"); // 镜像标识
envParams.put("ComputeEnv.EnvData.SystemDisk.DiskType", "LOCAL_BASIC"); // 系统盘类型
envParams.put("ComputeEnv.EnvData.SystemDisk.DiskSize", 50); // 系统盘大小
envParams.put("ComputeEnv.EnvData.LoginSettings.Password", "B1[habcdB1[habcd)]; // 登录密码
envParams.put("Placement.Zone", "ap-guangzhou-2"); // 可用区
envParams.put("Version", "2017-03-12");
```

```
String createRes = null;
try {
    createRes = module.call("CreateComputeEnv", envParams);
    JSONObject result = new JSONObject(createRes);
    System.out.println(result);

    result = result.getJSONObject("Response");
    System.out.println(result.getString("EnvId"));
} catch (Exception e) {
    System.out.println("error..." + e.getMessage());
}
```

更多参数说明及错误信息，详见 [创建计算环境 API 文档](#)。

修改计算环境

```
TreeMap<String, Object> envParams = new TreeMap<String, Object>();
envParams.put("EnvId", "env-cc44pzme"); // 计算环境标识
envParams.put("DesiredComputeNodeCount", 100); // 期望节点数目
envParams.put("Version", "2017-03-12");
```

```
String modRes = null;
try {
    modRes = module.call("ModifyComputeEnv", envParams);
    JSONObject result = new JSONObject(modRes);
    System.out.println(result);
} catch (Exception e) {
    System.out.println("error..." + e.getMessage());
}
```

更多参数说明及错误信息，详见 [修改计算环境 API 文档](#)。

删除计算集群

```
TreeMap<String, Object> delParams = new TreeMap<String, Object>();
delParams.put("EnvId", "env-cc44pzme"); // 计算环境标识
delParams.put("Version", "2017-03-12");
```

```
String delRes = null;
try {
    delRes = module.call("DeleteComputeEnv", delParams);
    JSONObject result = new JSONObject(delRes);
    System.out.println(result);
} catch (Exception e) {
    System.out.println("error..." + e.getMessage());
}
```

更多参数说明及错误信息，详见 [删除计算环境 API 文档](#)。

查看计算环境的创建信息

```
TreeMap<String, Object> infoParams = new TreeMap<String, Object>();
infoParams.put("EnvId", "env-cc44pzme"); // 计算环境标识
```

```
infoParams.put("Version", "2017-03-12");

String infoRes = null;
try {
    infoRes = module.call("DescribeComputeEnvCreateInfo", infoParams);
    JSONObject result = new JSONObject(infoRes);
    System.out.println(result);
} catch (Exception e) {
    System.out.println("error..." + e.getMessage());
}
```

更多参数说明及错误信息，详见 [查看计算环境的创建信息 API 文档](#)。

查看计算环境信息

```
TreeMap<String, Object> desParams = new TreeMap<String, Object>();
desParams.put("EnvId", "env-cc44pzme"); // 计算环境标识
desParams.put("Version", "2017-03-12");

String desRes = null;
try {
    desRes = module.call("DescribeComputeEnv", desParams);
    JSONObject result = new JSONObject(desRes);
    System.out.println(result);
} catch (Exception e) {
    System.out.println("error..." + e.getMessage());
}
```

更多参数说明及错误信息，详见 [查看计算环境信息 API 文档](#)。

查看计算环境列表

```
TreeMap<String, Object> listParams = new TreeMap<String, Object>();  
listParams.put("Version", "2017-03-12");
```

```
String listRes = null;  
try {  
    listRes = module.call("DescribeComputeEnvs", listParams);  
    JSONObject result = new JSONObject(listRes);  
    System.out.println(result);  
} catch (Exception e) {  
    System.out.println("error..." + e.getMessage());  
}
```

更多参数说明及错误信息，详见 [查看计算环境列表 API 文档](#)。

作业接口文档

开发准备

- 下载和安装 [Java SDK](#) 。
- 首次使用批量计算，参考 [开始前的准备](#)。
- 了解更多作业配置参数，参考 [提交作业 API 文档](#)。

快速入门

```
import java.util.TreeMap;

import com.qcloud.QcloudApiModuleCenter;
import com.qcloud.Module.Batch;
import com.qcloud.Utilities.Json.JSONObject;

public class BatchDemo {
    public static void main(String[] args) {
        TreeMap<String, Object> config = new TreeMap<String, Object>();
        config.put("SecretId", "您的SecretId");
        config.put("SecretKey", "您的SecretKey");
        config.put("RequestMethod", "GET");
        config.put("DefaultRegion", "gz"); // 地域, gz: guangzhou

        QcloudApiModuleCenter module = new QcloudApiModuleCenter(new Batch(), config);
    }
}
```

提交作业

```
TreeMap<String, Object> jobParams = new TreeMap<String, Object>();
jobParams.put("Job.JobName", "batch-job"); // 作业名称
jobParams.put("Job.JobDescription", "demo job"); // 作业描述
```

```
jobParams.put("Job.Priority", 1); // 作业优先级
jobParams.put("Job.Tasks.0.TaskName", "taskA"); // 任务名称
jobParams.put("Job.Tasks.0.TaskInstanceNum", 10); // 任务实例并发数
jobParams.put("Job.Tasks.0.Application.DeliveryForm", "LOCAL"); // 程序包来源
jobParams.put("Job.Tasks.0.Application.Command", "echo 'hello'"); // 命令行
jobParams.put("Job.Tasks.0.EnvId", "env-gbyctcy9"); // 计算环境标识
jobParams.put("Job.Tasks.0.RedirectInfo.StdoutRedirectPath",
"cos://bucketgz-1251783334.cos.ap-guangzhou.myqcloud.com/stdout/"); // 标准输出地址
jobParams.put("Job.Tasks.0.RedirectInfo.StderrRedirectPath",
"cos://bucketgz-1251783334.cos.ap-guangzhou.myqcloud.com/stderr/"); // 标准错误地址
jobParams.put("Placement.Zone", "ap-guangzhou-2"); // 可用区
jobParams.put("Version", "2017-03-12");
```

```
String submitRes = null;
String jobId = null;
try {
    submitRes = module.call("SubmitJob", jobParams);
    JSONObject result = new JSONObject(submitRes);
    System.out.println(result);

    result = result.getJSONObject("Response");
    jobId = result.getString("JobId");
    System.out.println(jobId);
} catch (Exception e) {
    System.out.println("error..." + e.getMessage());
}
```

更多参数说明及错误信息，详见 [提交作业 API 文档](#)

终止作业

```
TreeMap<String, Object> jobParams = new TreeMap<String, Object>();
jobParams.put("JobId", "job-8kwnzki7"); // 作业标识
```

```
jobParams.put("Version", "2017-03-12");
```

```
String termRes = null;
try {
    termRes = module.call("TerminateJob", jobParams);
    JSONObject result = new JSONObject(termRes);
    System.out.println(result);
} catch (Exception e) {
    System.out.println("error..." + e.getMessage());
}
```

更多参数说明及错误信息，详见 [终止作业 API 文档](#)

删除作业

```
TreeMap<String, Object> jobParams = new TreeMap<String, Object>();
jobParams.put("JobId", "job-8kwnzki7"); // 作业标识
jobParams.put("Version", "2017-03-12");
```

```
String delRes = null;
try {
    delRes = module.call("DeleteJob", jobParams);
    JSONObject result = new JSONObject(delRes);
    System.out.println(result);
} catch (Exception e) {
    System.out.println("error..." + e.getMessage());
}
```

更多参数说明及错误信息，详见 [删除作业 API 文档](#)

查看作业的提交信息

```
TreeMap<String, Object> jobParams = new TreeMap<String, Object>();  
jobParams.put("JobId", "job-8kwnzki7"); // 作业标识  
jobParams.put("Version", "2017-03-12");
```

```
String desRes = null;  
try {  
    desRes = module.call("DescribeJobSubmitInfo", jobParams);  
    JSONObject result = new JSONObject(desRes);  
    System.out.println(result);  
} catch (Exception e) {  
    System.out.println("error..." + e.getMessage());  
}
```

更多参数说明及错误信息，详见 [查看作业的提交信息 API 文档](#)

查看作业信息

```
TreeMap<String, Object> jobParams = new TreeMap<String, Object>();  
jobParams.put("JobId", "job-8kwnzki7"); // 作业标识  
jobParams.put("Version", "2017-03-12");
```

```
String desRes = null;  
try {  
    desRes = module.call("DescribeJob", jobParams);  
    JSONObject result = new JSONObject(desRes);  
    System.out.println(result);  
} catch (Exception e) {  
    System.out.println("error..." + e.getMessage());  
}
```

更多参数说明及错误信息，详见 [查看作业信息 API 文档](#)

查看作业列表

```
TreeMap<String, Object> jobParams = new TreeMap<String, Object>();
jobParams.put("Version", "2017-03-12");
```

```
String desRes = null;
try {
    desRes = module.call("DescribeJobs", jobParams);
    JSONObject result = new JSONObject(desRes);
    System.out.println(result);
} catch (Exception e) {
    System.out.println("error..." + e.getMessage());
}
```

更多参数说明及错误信息，详见 [查看作业列表 API 文档](#)

查看任务信息

```
TreeMap<String, Object> jobParams = new TreeMap<String, Object>();
jobParams.put("JobId", "job-8kwnzki7"); // 作业标识
jobParams.put("TaskName", "task A"); // 任务名称
jobParams.put("Version", "2017-03-12");
```

```
String desRes = null;
try {
    desRes = module.call("DescribeTask", jobParams);
    JSONObject result = new JSONObject(desRes);
    System.out.println(result);
} catch (Exception e) {
    System.out.println("error..." + e.getMessage());
}
```

更多参数说明及错误信息，详见 [查看任务信息 API 文档](#)

终止任务实例

```
TreeMap<String, Object> jobParams = new TreeMap<String, Object>();
jobParams.put("JobId", "job-8kwnzki7"); // 作业标识
jobParams.put("TaskName", "taskA"); // 任务名称
jobParams.put("TaskInstanceIndex", 0); // 任务名称
jobParams.put("Version", "2017-03-12");
```

```
String termRes = null;
try {
    termRes = module.call("TerminateTaskInstance", jobParams);
    JSONObject result = new JSONObject(termRes);
    System.out.println(result);
} catch (Exception e) {
    System.out.println("error..." + e.getMessage());
}
```

更多参数说明及错误信息，详见 [终止任务实例 API 文档](#)

Python SDK

计算环境接口文档

开发准备

- 下载和安装 [Python SDK](#)。
- 首次使用批量计算，参考 [开始前的准备](#)。
- 了解更多计算环境配置参数，参考 [创建计算环境 API 文档](#)。

快速入门

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

# 引入云API入口模块
from QcloudApi.qcloudapi import QcloudApi

# 公共配置
module = 'batch'

config = {
    'Region': 'ap-beijing',
    'secretId': '您的secretId',
    'secretKey': '您的secretKey',
}

service = QcloudApi(module, config)
```

创建计算环境

```
try:
    action = 'CreateComputeEnv'
```

```
action_params = {
  'Version': '2017-03-12',
  "ComputeEnv": {
    "EnvName": "Cluster-A", # 计算环境名称
    "EnvDescription": "first cluster", # 计算环境描述
    "EnvType": "MANAGED",
    "MountDataDisks": [
      {
        "FileSystemType": "NTFS",
        "LocalPath": "J://", # 数据盘挂载点
      },
    ],
    "EnvData": {
      "InstanceType": "S2.SMALL1", # 实例类型
      "ImageId": "img-er9shcln", # 镜像标识
      "LoginSettings": {
        "Password": "B1[habcdB1[habcd" # 登录密码
      },
      "InternetAccessible": {
        "PublicIpAssigned": "TRUE", # 公网IP
        "InternetMaxBandwidthOut": 50 # 公网带宽
      },
      "SystemDisk": {
        "DiskType": "LOCAL_BASIC", # 系统盘类型
        "DiskSize": 50 # 系统盘大小
      },
      "DataDisks": [
        {
          "DiskType": "LOCAL_BASIC", # 数据盘类型
          "DiskSize": 50 # 数据盘大小
        }
      ],
    },
    "DesiredComputeNodeCount": 1 # 计算节点数量
  }
}
```



```
},
"Placement": {
  "Zone": "ap-beijing-2" # 可用区
},
}

print(service.generateUrl(action, action_params))
print(service.call(action, action_params))

except Exception as e:
  import traceback
  print('traceback.format_exc():\n%s' % traceback.format_exc())
```

更多参数说明及错误信息，详见 [创建计算环境 API 文档](#)。

修改计算环境

```
try:
  action = 'ModifyComputeEnv'
  action_params = {
    'Version': '2017-03-12',
    'EnvId': "env-cc44pzme", # 计算环境标识
    'DesiredComputeNodeCount': 100, # 期望节点数目
  }
  print(service.generateUrl(action, action_params))
  print(service.call(action, action_params))

except Exception as e:
  import traceback
  print('traceback.format_exc():\n%s' % traceback.format_exc())
```

更多参数说明及错误信息，详见 [修改计算环境 API 文档](#)。

删除计算集群

```
try:
    action = 'DeleteComputeEnv'
    action_params = {
        'Version': '2017-03-12',
        'EnvId': "env-cc44pzme", # 计算环境标识
    }
    print(service.generateUrl(action, action_params))
    print(service.call(action, action_params))
except Exception as e:
    import traceback
    print('traceback.format_exc():\n%s' % traceback.format_exc())
```

更多参数说明及错误信息，详见 [删除计算环境 API 文档](#)。

查看计算环境的创建信息

```
try:
    action = 'DescribeComputeEnvCreateInfo'
    action_params = {
        'Version': '2017-03-12',
        'EnvId': "", # 计算环境标识
    }
    print(service.generateUrl(action, action_params))
    print(service.call(action, action_params))
except Exception as e:
    import traceback
    print('traceback.format_exc():\n%s' % traceback.format_exc())
```

更多参数说明及错误信息，详见 [查看计算环境的创建信息 API 文档](#)。

查看计算环境信息

```
try:
    action = 'DescribeComputeEnv'
    action_params = {
        'Version': '2017-03-12',
        'EnvId': "", # 计算环境标识
    }
    print(service.generateUrl(action, action_params))
    print(service.call(action, action_params))
except Exception as e:
    import traceback
    print('traceback.format_exc():\n%s' % traceback.format_exc())
```

更多参数说明及错误信息，详见 [查看计算环境信息 API 文档](#)。

查看计算环境列表

```
try:
    action = 'DescribeComputeEnvs'
    action_params = {
        'Version': '2017-03-12',
    }
    print(service.generateUrl(action, action_params))
    print(service.call(action, action_params))
except Exception as e:
    import traceback
    print('traceback.format_exc():\n%s' % traceback.format_exc())
```

更多参数说明及错误信息，详见 [查看计算环境列表 API 文档](#)。

作业接口文档

开发准备

- 下载和安装 [Python SDK](#)。
- 首次使用批量计算，参考 [开始前的准备](#)。
- 了解更多作业配置参数，参考 [提交作业 API 文档](#)。

快速入门

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

# 引入云API入口模块
from QcloudApi.qcloudapi import QcloudApi

# 公共配置
module = 'batch'

config = {
    'Region': 'ap-guangzhou', # 目标地域
    'secretId': '您的secretId',
    'secretKey': '您的secretKey',
}

service = QcloudApi(module, config)
```

提交作业

```
try:
    action = 'SubmitJob'
    action_params = {
        'Version': '2017-03-12',
```

```
'Job': {
  'JobName': 'batch-job', # 作业名称
  'JobDescription': 'batch job', # 作业描述
  'Priority': '1', # 作业优先级
  'Tasks': [
    {
      'TaskName': 'task1', # 任务名称
      'TaskInstanceNum': 1, # 任务实例数目
      'FailedAction': 'FAST_INTERRUPT', # 失败作业的处理方式
      'Application': {
        'DeliveryForm': 'LOCAL', # 程序包来源
        'Command': 'echo hello', # 命令行
      },
      'EnvId': 'env-gbyctcy9', # 计算环境标识
      'RedirectInfo': {
        'StdoutRedirectPath': 'cos://bucketgz-1251783334.cos.ap-guangzhou.myqcloud.com/stdout/', #
标准输出存储路径
        'StderrRedirectPath': 'cos://bucketgz-1251783334.cos.ap-guangzhou.myqcloud.com/stderr/', #
标准错误存储路径
      }
    }
  ],
  'Placement': {
    'Zone': 'ap-guangzhou-2' # 可用区
  }
}

print(service.generateUrl(action, action_params))
print(service.call(action, action_params))
except Exception as e:
  import traceback
  print('traceback.format_exc():\n%s' % traceback.format_exc())
```

更多参数说明及错误信息，详见 [提交作业 API 文档](#)

终止作业

```
try:
    action = 'TerminateJob'
    action_params = {
        'Version': '2017-03-12',
        'JobId': 'job-8kwnzki7', # 作业标识
    }
    print(service.generateUrl(action, action_params))
    print(service.call(action, action_params))
except Exception as e:
    import traceback
    print('traceback.format_exc():\n%s' % traceback.format_exc())
```

更多参数说明及错误信息，详见 [终止作业 API 文档](#)

删除作业

```
try:
    action = 'DeleteJob'
    action_params = {
        'Version': '2017-03-12',
        'JobId': 'job-8kwnzki7', # 作业标识
    }
    print(service.generateUrl(action, action_params))
    print(service.call(action, action_params))
except Exception as e:
    import traceback
    print('traceback.format_exc():\n%s' % traceback.format_exc())
```

更多参数说明及错误信息，详见 [删除作业 API 文档](#)

查看作业的提交信息

```
try:
    action = 'DescribeJobSubmitInfo'
    action_params = {
        'Version': '2017-03-12',
        'JobId': 'job-8kwnzki7', # 作业标识
    }
    print(service.generateUrl(action, action_params))
    print(service.call(action, action_params))
except Exception as e:
    import traceback
    print('traceback.format_exc():\n%s' % traceback.format_exc())
```

更多参数说明及错误信息，详见 [查看作业的提交信息 API 文档](#)

查看作业信息

```
try:
    action = 'DescribeJob'
    action_params = {
        'Version': '2017-03-12',
        'JobId': 'job-8kwnzki7', # 作业标识
    }
    print(service.generateUrl(action, action_params))
    print(service.call(action, action_params))
except Exception as e:
    import traceback
    print('traceback.format_exc():\n%s' % traceback.format_exc())
```

更多参数说明及错误信息，详见 [查看作业信息 API 文档](#)

查看作业列表

```
try:
    action = 'DescribeJobs'
    action_params = {
        'Version': '2017-03-12'
    }
    print(service.generateUrl(action, action_params))
    print(service.call(action, action_params))
except Exception as e:
    import traceback
    print('traceback.format_exc():\n%s' % traceback.format_exc())
```

更多参数说明及错误信息，详见 [查看作业列表 API 文档](#)

查看任务信息

```
try:
    action = 'DescribeTask'
    action_params = {
        'Version': '2017-03-12',
        'JobId': 'job-8kwnzki7', # 作业标识
        'TaskName': 'task A' # 任务名称
    }
    print(service.generateUrl(action, action_params))
    print(service.call(action, action_params))
except Exception as e:
    import traceback
    print('traceback.format_exc():\n%s' % traceback.format_exc())
```


更多参数说明及错误信息，详见 [查看任务信息 API 文档](#)

终止任务实例

```
try:
    action = 'TerminateTaskInstance'
    action_params = {
        'Version': '2017-03-12',
        'JobId': 'job-8kwnzki7', # 作业标识
        'TaskName': 'task A', # 任务名称
        'TaskInstanceIndex': 1 # 第1个实例
    }
    print(service.generateUrl(action, action_params))
    print(service.call(action, action_params))
except Exception as e:
    import traceback
    print('traceback.format_exc():\n%s' % traceback.format_exc())
```

更多参数说明及错误信息，详见 [终止任务实例 API 文档](#)