

# **Batch Compute**

## **SDK Documentation**

### **Product Documentation**



## Copyright Notice

©2013–2019 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

## Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

## Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

---

# Contents

## SDK Documentation

- Instructions

- Java SDK

  - Compute Environment Documentation

  - Job API Documentation

- Python SDK

  - Compute Environment Documentation

  - Job API Documentation

# SDK Documentation

## Instructions

Last updated : 2019-10-16 19:50:43

### Preparations

- Download and install the SDK for your language in Tencent Cloud SDK Center.
- If this is your first time using BatchCompute, please see [Preparation](#).
- For more information about API parameters, see [API Documentation](#).

### Access Steps

1. Select **BatchCompute** and find the corresponding API.
2. Enter **Personal key** and **Input parameters** according to the help prompts.
3. (Optional) Select **Only view required parameters**.
4. In the **Code generation** column on the right, copy the different language code to local execution.
5. Before the producer is used, a real request is initiated in the **Online calling** column to verify whether the results meet expectations.

### Sample Codes

#### Submitting a Job (Python Version)

```
from tencentcloud.common import credential
from tencentcloud.common.profile.client_profile import ClientProfile
from tencentcloud.common.profile.http_profile import HttpProfile
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudSDKException
from tencentcloud.batch.v20170312 import batch_client, models

try:
    cred = credential.Credential("your secret id", "your secret key")
    httpProfile = HttpProfile()
    httpProfile.endpoint = "batch.tencentcloudapi.com"

    clientProfile = ClientProfile()
    clientProfile.httpProfile = httpProfile
    client = batch_client.BatchClient(cred, "ap-guangzhou", clientProfile)
```

```
req = models.SubmitJobRequest()
params = '{"Placement":{"Zone":"ap-guangzhou-3"},"Job":{"JobName":"demo","JobDescription":"test job","Priority":1,"Tasks":[{"TaskName":"task","TaskInstanceNum":1,"Application":{"Command":"echo hello"},"ComputeEnv":{"EnvData":{"InstanceType":"S2.SMALL1","ImageId":"img-enf3kukl"},"SystemDisk":{"DiskType":"CLOUD_PREMIUM","DiskSize":50}}},"MaxRetryCount":1,"Timeout":3600}]}'
req.from_json_string(params)

resp = client.SubmitJob(req)
print(resp.to_json_string())

except TencentCloudSDKException as err:
    print(err)
```

## Querying a Job (Python Version)

```
from tencentcloud.common import credential
from tencentcloud.common.profile.client_profile import ClientProfile
from tencentcloud.common.profile.http_profile import HttpProfile
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudSDKException
from tencentcloud.batch.v20170312 import batch_client, models
try:
    cred = credential.Credential("your secret id", "your secret key")
    httpProfile = HttpProfile()
    httpProfile.endpoint = "batch.tencentcloudapi.com"

    clientProfile = ClientProfile()
    clientProfile.httpProfile = httpProfile
    client = batch_client.BatchClient(cred, "ap-guangzhou", clientProfile)

    req = models.DescribeJobRequest()
    params = '{"JobId":"job-mhgy1dot"}'
    req.from_json_string(params)

    resp = client.DescribeJob(req)
    print(resp.to_json_string())

except TencentCloudSDKException as err:
    print(err)
```

## Submitting a Job (Java Version)

```
import com.tencentcloudapi.common.Credential;
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;
import com.tencentcloudapi.common.exception.TencentCloudSDKException;
```

```

import com.tencentcloudapi.batch.v20170312.BatchClient;

import com.tencentcloudapi.batch.v20170312.models.SubmitJobRequest;
import com.tencentcloudapi.batch.v20170312.models.SubmitJobResponse;

public class SubmitJob
{
    public static void main(String [] args) {
    try{

        Credential cred = new Credential("your secret id", "your secret key");

        HttpProfile httpProfile = new HttpProfile();
        httpProfile.setEndpoint("batch.tencentcloudapi.com");

        ClientProfile clientProfile = new ClientProfile();
        clientProfile.setHttpProfile(httpProfile);

        BatchClient client = new BatchClient(cred, "ap-guangzhou", clientProfile);

        String params = "{\r\n\"Placement\": {\r\n\"Zone\": \"ap-guangzhou-3\"}, \r\n\"Job\": {\r\n\"JobName\": \"demo\", \r\n\"Job\r\n\r\nDescription\": \"test job\", \r\n\"Priority\": 1, \r\n\"Tasks\": [\r\n{\r\n\"TaskName\": \"task\", \r\n\"TaskInstanceNum\":\r\n\r\n1, \r\n\"Application\": {\r\n\"Command\": \"echo hello\"}, \r\n\"ComputeEnv\": {\r\n\"EnvData\": {\r\n\"InstanceType\": \"S\r\n\r\n2.SMALL1\", \r\n\"ImageId\": \"img-enf3kukl\", \r\n\"SystemDisk\": {\r\n\"DiskType\": \"CLOUD_PREMIUM\", \r\n\"DiskSize\r\n\r\n\": 50}}}, \r\n\"MaxRetryCount\": 1, \r\n\"Timeout\": 3600} ]}}";
        SubmitJobRequest req = SubmitJobRequest.fromJsonString(params, SubmitJobRequest.class);

        SubmitJobResponse resp = client.SubmitJob(req);

        System.out.println(SubmitJobRequest.toJsonString(resp));
    } catch (TencentCloudSDKException e) {
        System.out.println(e.toString());
    }

}

}

```

## Querying a Job (Java Version)

```

import com.tencentcloudapi.common.Credential;
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;
import com.tencentcloudapi.common.exception.TencentCloudSDKException;

import com.tencentcloudapi.batch.v20170312.BatchClient;

```

```
import com.tencentcloudapi.batch.v20170312.models.DescribeJobRequest;
import com.tencentcloudapi.batch.v20170312.models.DescribeJobResponse;

public class DescribeJob
{
public static void main(String [] args) {
try{

Credential cred = new Credential("your secret id", "your secret key");

HttpProfile httpProfile = new HttpProfile();
httpProfile.setEndpoint("batch.tencentcloudapi.com");

ClientProfile clientProfile = new ClientProfile();
clientProfile.setHttpProfile(httpProfile);

BatchClient client = new BatchClient(cred, "ap-guangzhou", clientProfile);

String params = "{\":JobId\":\":job-mhgy1dot\"}";
DescribeJobRequest req = DescribeJobRequest.fromJsonString(params, DescribeJobRequest.class);

DescribeJobResponse resp = client.DescribeJob(req);

System.out.println(DescribeJobRequest.toJsonString(resp));
} catch (TencentCloudSDKException e) {
System.out.println(e.toString());
}

}

}
```

# Java SDK

## Compute Environment Documentation

Last updated : 2019-09-23 18:07:10

### Preparations for Development

- Download and install the [Java SDK](#).
- Before using Batch for the first time, see [Preparation](#).

### Getting Started

```
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", "Your SecretId");
        config.put("SecretKey", "Your SecretKey");
        config.put("RequestMethod", "GET");
        config.put("DefaultRegion", "gz"); // Region; gz: guangzhou

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

### Creating a Compute Environment

```
TreeMap<String, Object> envParams = new TreeMap<String, Object>();
envParams.put("ComputeEnv.EnvName", "batch-env"); // Compute environment name
envParams.put("ComputeEnv.EnvType", "MANAGED");
envParams.put("ComputeEnv.DesiredComputeNodeCount", 10); // Number of desired nodes
envParams.put("ComputeEnv.EnvData.InstanceType", "S1.SMALL1"); // Instance type
envParams.put("ComputeEnv.EnvData.ImageId", "img-er9shc1n"); // Image ID
```



```
envParams.put("ComputeEnv.EnvData.SystemDisk.DiskType", "LOCAL_BASIC"); // System disk type
envParams.put("ComputeEnv.EnvData.SystemDisk.DiskSize", 50); // System disk size
envParams.put("ComputeEnv.EnvData.LoginSettings.Password", "B1[habcdB1[habcd]"); // Login password
envParams.put("Placement.Zone", "ap-guangzhou-2"); // Availability zone
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());
}
```

## Modifying a Compute Environment

```
TreeMap<String, Object> envParams = new TreeMap<String, Object>();
envParams.put("EnvId", "env-cc44pzme"); // Compute environment ID
envParams.put("DesiredComputeNodeCount", 100); // Number of desired nodes
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());
}
```

## Deleting a Compute Cluster

```
TreeMap<String, Object> delParams = new TreeMap<String, Object>();
delParams.put("EnvId", "env-cc44pzme"); // Compute environment ID
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());
}
```

## Viewing Compute Environment Creation Information

```
TreeMap<String, Object> infoParams = new TreeMap<String, Object>();
infoParams.put("EnvId", "env-cc44pzme"); // Compute environment ID
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());
}
```

## Viewing Compute Environment Information

```
TreeMap<String, Object> desParams = new TreeMap<String, Object>();
desParams.put("EnvId", "env-cc44pzme"); // Compute environment ID
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());
}
```

## Viewing Compute Environment List

```
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());
}
```

# Job API Documentation

Last updated : 2019-09-23 18:09:14

## Preparations for Development

- Download and install the [Java SDK](#).
- Before using Batch for the first time, see [Preparation](#).

## Getting Started

```
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", "Your SecretId");
config.put("SecretKey", "Your SecretKey");
config.put("RequestMethod", "GET");
config.put("DefaultRegion", "gz"); // Region; gz: guangzhou

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

## Submitting a Job

```
TreeMap<String, Object> jobParams = new TreeMap<String, Object>();
jobParams.put("Job.JobName", "batch-job"); // Job name
jobParams.put("Job.JobDescription", "demo job"); // Job description
jobParams.put("Job.Priority", 1); // Job priority
jobParams.put("Job.Tasks.0.TaskName", "taskA"); // Task name
jobParams.put("Job.Tasks.0.TaskInstanceNum", 10); // Number of concurrent task instances
jobParams.put("Job.Tasks.0.Application.DeliveryForm", "LOCAL"); // Package source
jobParams.put("Job.Tasks.0.Application.Command", "echo 'hello'"); // Command line
```

```
jobParams.put("Job.Tasks.0.EnvId", "env-gbyctcy9"); // Compute environment ID
jobParams.put("Job.Tasks.0.RedirectInfo.StdoutRedirectPath", "cos://bucketgz-1251783334.cos.ap-guangzhou.myqcloud.com/stdout/"); // Standard output address
jobParams.put("Job.Tasks.0.RedirectInfo.StderrRedirectPath", "cos://bucketgz-1251783334.cos.ap-guangzhou.myqcloud.com/stderr/"); // Standard error address
jobParams.put("Placement.Zone", "ap-guangzhou-2"); // Availability zone
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());
}
```

## Terminating a Job

```
TreeMap<String, Object> jobParams = new TreeMap<String, Object>();
jobParams.put("JobId", "job-8kwnzki7"); // Job ID
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());
}
```

## Deleting a Job

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

```
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());
}
```

## Viewing Job Submission Information

```
TreeMap<String, Object> jobParams = new TreeMap<String, Object>();
jobParams.put("JobId", "job-8kwnzki7"); // Job ID
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());
}
```

## Viewing Job Information

```
TreeMap<String, Object> jobParams = new TreeMap<String, Object>();
jobParams.put("JobId", "job-8kwnzki7"); // Job ID
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());
}
```

## Viewing Job List

```
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());
}
```

## Viewing Task Information

```
TreeMap<String, Object> jobParams = new TreeMap<String, Object>();
jobParams.put("JobId", "job-8kwnzki7"); // Job ID
jobParams.put("TaskName", "task A"); // Task name
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());
}
```

## Terminating a Task Instance

```
TreeMap<String, Object> jobParams = new TreeMap<String, Object>();
jobParams.put("JobId", "job-8kwnzki7"); // Job ID
jobParams.put("TaskName", "taskA"); // Task name
jobParams.put("TaskInstanceIndex", 0); // Task name
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());
}
```



# Python SDK

## Compute Environment Documentation

Last updated : 2019-09-23 18:01:58

### Preparations for Development

- Download and install [Python SDK](#).
- Before using Batch for the first time, see [Preparation](#).

#### ##Getting Started

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

# Introducing the Cloud API Entry Module
from QcloudApi.qcloudapi import QcloudApi

# Common configuration
module = 'batch'

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

service = QcloudApi(module, config)
```

#### ##Creating a Compute Environment

```
try:
    action = 'CreateComputeEnv'
    action_params = {
        'Version': '2017-03-12',
        "ComputeEnv": {
            "EnvName": "Cluster-A", # Compute environment name
            "EnvDescription": "first cluster", # Compute environment description
            "EnvType": "MANAGED",
            "MountDataDisks": [
                {
                    "FileSystemType": "NTFS",
```

```
"LocalPath": "J://", # Data disk mounting point
},
],
"EnvData": {
"InstanceType": "S2.SMALL1", # Instance type
"ImageId": "img-er9shcln", # Image ID
"LoginSettings": {
"Password": "B1[habcdB1[habcd" # Login password
},
"InternetAccessible": {
"PublicIpAssigned": "TRUE", # Public IP
"InternetMaxBandwidthOut": 50 # Internet bandwidth
},
"SystemDisk": {
"DiskType": "LOCAL_BASIC", # System disk type
"DiskSize": 50 # System disk size
},
"DataDisks": [
{
"DiskType": "LOCAL_BASIC", # Data disk type
"DiskSize": 50 # Data disk size
}
],
"DesiredComputeNodeCount": 1 # Number of compute nodes
},
"Placement": {
"Zone": "ap-beijing-2" # Availability zone
},
}
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())
```

## Modifying a Compute Environment

```
try:
action = 'ModifyComputeEnv'
action_params = {
'Version': '2017-03-12',
'EnvId': "env-cc44pzme", # Compute environment ID
'DesiredComputeNodeCount': 100, # Number of desired nodes
```

```
}
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())
```

## Deleting a Compute Cluster

```
try:
action = 'DeleteComputeEnv'
action_params = {
'Version': '2017-03-12',
'EnvId': "env-cc44pzme", # Compute environment ID
}
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())
```

## Viewing Compute Environment Creation Information

```
try:
action = 'DescribeComputeEnvCreateInfo'
action_params = {
'Version': '2017-03-12',
'EnvId': "", # Compute environment ID
}
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())
```

## Viewing Compute Environment Information

```
try:
action = 'DescribeComputeEnv'
action_params = {
'Version': '2017-03-12',
'EnvId': "", # Compute environment ID
}
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())
```

## Viewing Compute Environment List

```
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())
```

# Job API Documentation

Last updated : 2019-09-23 18:02:33

## Preparations for Development

- Download and install [Python SDK](#).
- Before using Batch for the first time, see [Preparation](#).

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

# Introducing the Cloud API Entry Module
from QcloudApi.qcloudapi import QcloudApi

# Common configuration
module = 'batch'

config = {
    'Region': 'ap-guangzhou', # Destination region
    'secretId': 'Your secretId',
    'secretKey': 'Your secretKey',
}

service = QcloudApi(module, config)
```

## Submitting a Job

```
try:
    action = 'SubmitJob'
    action_params = {
        'Version': '2017-03-12',
        'Job': {
            'JobName': 'batch-job', # Job name
            'JobDescription': 'batch job', # Job description
            'Priority': '1', # Job priority
            'Tasks': [
                {
                    'TaskName': 'task1', # Task name
                    'TaskInstanceNum': 1, # Number of task instances
```

```
'FailedAction': 'FAST_INTERRUPT', # Processing method for failing job
'Application': {
'DeliveryForm': 'LOCAL', # Package source
'Command': 'echo hello', # Command line
},
'EnvId': 'env-gbyctcy9', # Compute environment ID
'RedirectInfo': {
'StdoutRedirectPath': 'cos://bucketgz-1251783334.cos.ap-guangzhou.myqcloud.com/stdout', # Standard output storage path
'StderrRedirectPath': 'cos://bucketgz-1251783334.cos.ap-guangzhou.myqcloud.com/stderr', # Standard error storage path
}
}
],
},
'Placement': {
'Zone': 'ap-guangzhou-2' # Availability zone
},
}
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())
```

## Terminating a Job

```
try:
action = 'TerminateJob'
action_params = {
'Version': '2017-03-12',
'JobId': 'job-8kwnzki7', # Job ID
}
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())

try:
action = 'DeleteJob'
action_params = {
'Version': '2017-03-12',
'JobId': 'job-8kwnzki7', # Job ID
```

```
}  
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())
```

## Viewing Job Submission Information

```
try:  
action = 'DescribeJobSubmitInfo'  
action_params = {  
    'Version': '2017-03-12',  
    'JobId': 'job-8kwnzki7', # Job ID  
}  
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())
```

## Viewing Job Information

```
try:  
action = 'DescribeJob'  
action_params = {  
    'Version': '2017-03-12',  
    'JobId': 'job-8kwnzki7', # Job ID  
}  
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())
```

## Viewing Job List

```
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())
```

## Viewing Task Information

```
try:
    action = 'DescribeTask'
    action_params = {
        'Version': '2017-03-12',
        'JobId': 'job-8kwnzki7', # Job ID
        'TaskName': 'task A' # Task name
    }
    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())
```

```
try:
    action = 'TerminateTaskInstance'
    action_params = {
        'Version': '2017-03-12',
        'JobId': 'job-8kwnzki7', # Job ID
        'TaskName': 'task A', # Task name
        'TaskInstanceIndex': 1 # The first instance
    }
    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())
```