ALIBABA CLOUD

# Alibaba Cloud

## FunctionFlow

## SDK Reference

Document Version: 20220610

Alibaba Cloud

# Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

# Document conventions

| Style | Description | Example |
| --- | --- | --- |
| Danger | A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results. | **Danger:**<br>Resetting will result in the loss of user configuration data. |
| Warning | A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results. | **Warning:**<br>Restarting will cause business interruption. About 10 minutes are required to restart an instance. |
| Notice | A caution notice indicates warning information, supplementary instructions, and other content that the user must understand. | **Notice:**<br>If the weight is set to 0, the server no longer receives new requests. |
| Note | A note indicates supplemental instructions, best practices, tips, and other content. | **Note:**<br>You can use Ctrl + A to select all files. |
| > | Closing angle brackets are used to indicate a multi-level menu cascade. | Click **Settings**> **Network**> **Set network type**. |
| **Bold** | Bold formatting is used for buttons , menus, page names, and other UI elements. | Click **OK**. |
| `Courier font` | Courier font is used for commands | Run the `cd /d C:/window` command to enter the Windows system folder. |
| *Italic* | Italic formatting is used for parameters and variables. | `bae log list --instanceid`<br>*Instance_ID* |
| [] or [a\|b] | This format is used for an optional value, where only one item can be selected. | `ipconfig [-all\|-t]` |
| {} or {a\|b} | This format is used for a required value, where only one item can be selected. | `switch {active\|stand}` |

# Table of Contents

# 1.SDK reference

This topic describes SDK for multiple programming languages.

## Supported languages

- Java
- Python
- PHP
- .NET
- Go
- Node.js

# 2.SDK for Go

This topic describes how to use SDK for Go.

## Prerequisites

Your system needs to meet the environment requirements. For example, Go 1.10.x or later has been installed in your system.

## Installation

Run the `go get` command to download and install SDK for Go.

```
$ go get -u github.com/aliyun/alibaba-cloud-sdk-go/sdk
```

If you use glide to manage dependencies, you can also use it to install Alibaba Cloud SDK for Go.

```
$ glide get github.com/aliyun/alibaba-cloud-sdk-go
```

## Quick start

You must have an Alibaba Cloud account and an AccessKey pair to use SDK for Go. The following example shows how to use SDK for Go to call . In this example, a flow is created, an execution is initiated, and the execution details are queried.

Request method

```
package main
import (
    "fmt"
    "time"
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk/requests"
    "github.com/aliyun/alibaba-cloud-sdk-go/services/fnf"
)
var (
    flowDefinitionType = "FDL"
    flowName           = "xxx"
    flowDefinition     = `xxx`
    flowDescription    = "some descriptions"
    roleArn            = "acs:ram::${Your_Account_ID}:${Your_Role}"
    executionName      = "xxx"
)
// CreateFlow ...
func CreateFlow(cli *fnf.Client) (*fnf.CreateFlowResponse, error) {
    request := fnf.CreateCreateFlowRequest()
    request.Name = flowName
    request.Definition = flowDefinition
    request.Description = flowDescription
    request.Type = flowDefinitionType
    request.RoleArn = roleArn
    return cli.CreateFlow(request)
}
// StartExecution ...
func StartExecution(cli *fnf.Client) (*fnf.StartExecutionResponse, error) {
    request := fnf.CreateStartExecutionRequest()
    request.FlowName = flowName
    request.ExecutionName = executionName
    return cli.StartExecution(request)
}
// DescribeExecution ...
func DescribeExecution(cli *fnf.Client) (*fnf.DescribeExecutionResponse, error) {
    request := fnf.CreateDescribeExecutionRequest()
    request.FlowName = flowName
    request.ExecutionName = executionName
    return cli.DescribeExecution(request)
}
// GetExecutionHistory ...
func GetExecutionHistory(cli *fnf.Client) (*fnf.GetExecutionHistoryResponse, error) {
    request := fnf.CreateGetExecutionHistoryRequest()
    // request.Limit and request.NextToken can set here. For easy demo, we passed.
    request.FlowName = flowName
    request.ExecutionName = executionName
    return cli.GetExecutionHistory(request)
}
```

Create a client and use the preceding functions to initiate a series of calls.

> **Note** If you need to implement debugging without modification, place the following function in the same file as the preceding request method code block to avoid errors upon import.

```
func main() {
    fnfCli, err := fnf.NewClientWithAccessKey("cn-hangzhou", "AccessID", "AccessKey")
    if err != nil {
        panic(err)
    }
    // Create a flow
    _, err = CreateFlow(fnfCli)
    if err != nil {
        panic(err)
    }
    // StartExecution
    _, err = StartExecution(fnfCli)
    if err != nil {
        panic(err)
    }
    time.Sleep(time.Second)
    // DescribeExecution
    desResp, err := DescribeExecution(fnfCli)
    if err != nil {
        panic(err)
    }
    fmt.Println(fmt.Sprintf("%s status: %s", desResp.Name, desResp.Status))
    // GetExecutionHistory
    _, err = GetExecutionHistory(fnfCli)
    if err != nil {
        panic(err)
    }
}
```

# 3.Use Alibaba Cloud SDK for Java

This topic describes how to use Alibaba Cloud SDK for Java.

## Prerequisites

- An Alibaba Cloud account and an AccessKey pair are created. An AccessKey pair contains an `AccessKey ID` and an `AccessKey secret`. You can go to the Security Management page of the Alibaba Cloud Management Console to create or view AccessKey pairs. You can also contact your system administrator to obtain an AccessKey pair.
- Serverless Workflow is activated in the Alibaba Cloud Management Console. This allows you to use Alibaba Cloud SDK for Java to call the API operations of Serverless Workflow.
- Java Development Kit (JDK) 1.6 or later is installed.

## Dependency installation

You must install `aliyun-java-sdk-core` to use the SDK of an Alibaba Cloud service. For example, to use SDK, you must install `aliyun-java-sdk-core` and `aliyun-java-sdk-fnf`.

We recommend that you use Apache Maven to manage project dependencies. If you use Apache Maven to manage a Java project, add the following dependencies to the `pom.xml` file of the project:

```
<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <version>[4.3.2,5.0.0)</version>
</dependency>
<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-fnf</artifactId>
    <version>[1.0.0,5.0.0)</version>
</dependency>
```

If you do not use Apache Maven to download the JAR package from the central repository, add the following dependency to the `pom.xml` file. Otherwise, a `NoClassDefFoundError` error is reported.

```
<dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.8.5</version>
</dependency>
```

## Quick start

The following example shows you how to use Alibaba Cloud SDK for Java to call . In this example, a flow is created, an execution is initiated, and the execution details are queried.

To call Alibaba Cloud SDK for Java, perform the following steps:

1. Create a DefaultAcsClient object and initialize the object.

2. Create an API request and set the required parameters. For more information, see List of operations by function.
3. Initiate the API request and process the response or exceptions.

> **Notice** The following example describes how to use Alibaba Cloud SDK for Java to call . When you call the service, you may encounter a debugging issue or want to use advanced features such as the connection pool, HTTPS, proxy, and log features. For more information, see README.

The following code provides an example on how to create an API request.

```
package com.test;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.fnf.model.v20190315.*;
class FnFOperations {
    static String flowName = "xxx";
    static String execName = "xxx";
    static String flowDesc = "xxx";
    static String flowDef = "xxx";
    static String roleArn = "xxx";
    static String flowType = "xxx";
    // Create a Serverless Workflow flow.
    static CreateFlowResponse createFlow(IAcsClient fnfClient) throws ClientException {
        CreateFlowRequest request = new CreateFlowRequest();
        request.setName(flowName);
        request.setDefinition(flowDef);
        request.setType(flowType);
        request.setDescription(flowDesc);
        request.setRoleArn(roleArn);
        return fnfClient.getAcsResponse(request);
    }
    // Initiate an execution.
    static StartExecutionResponse startExecution(IAcsClient fnfClient) throws ClientExcepti
on {
        StartExecutionRequest request = new StartExecutionRequest();
        request.setFlowName(flowName);
        request.setExecutionName(execName);
        return fnfClient.getAcsResponse(request);
    }
    // Query the execution result.
    static DescribeExecutionResponse describeExecution(IAcsClient fnfClient) throws ClientE
xception {
        DescribeExecutionRequest request = new DescribeExecutionRequest();
        request.setFlowName(flowName);
        request.setExecutionName(execName);
        return fnfClient.getAcsResponse(request);
    }
    // Query historical execution records.
    static GetExecutionHistoryResponse getExecutionHistory(IAcsClient fnfClient) throws Cli
entException {
        GetExecutionHistoryRequest request = new GetExecutionHistoryRequest();
        request.setFlowName(flowName);
        request.setExecutionName(execName);
        return fnfClient.getAcsResponse(request);
    }
}
```

Create a client object and use the preceding functions to initiate a series of requests.

**Note** If you need to perform debugging without modification, place the following public class in the same file as the preceding code block that is used to create an API request. This helps prevent errors upon import.

```
public class Main {
    public static void main(String[] args) {
        // Create DefaultAcsClient
        DefaultProfile profile = DefaultProfile.getProfile(
                "<your-region-id>",           // The ID of the region.
                "<your-access-key-id>", // The AccessKey ID of the RAM user.
                "<your-access-key-secret>"); //The AccessKey secret of the RAM user.
        IAcsClient client = new DefaultAcsClient(profile);
        try {
            // Create Flow
            CreateFlowResponse createFlowResponse = FnFOperations.createFlow(client);
            System.out.println(creatFlowResponse);
            // Start Execution
            StartExecutionResponse startExeResp = FnFOperations.startExecution(client);
            System.out.println(startExeResp);
            // Describe Execution
            DescribeExecutionResponse descExeResp = FnFOperations.describeExecution(client)
;
            System.out.println(descExeResp);
        } catch (ClientException e) {
            e.printStackTrace();
        }
        try {
            GetExecutionHistoryResponse resp = FnFOperations.getExecutionHistory(client);
            for (GetExecutionHistoryResponse.EventsItem event:resp.getEvents()) {
                System.out.printf("event %s status: %s%n", event.getStepName(), event.getTy
pe());
            }
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

# 4.SDK for Node.js

This topic describes how to use SDK for Node.js.

## Installation

For example, run the following npm command to install SDK for Node.js:

```
npm install @alicloud/fnf-2019-03-15
```

Alibaba Cloud SDK for Node.js is also released at GitHub.

## Quick start

You must have an Alibaba Cloud account and an AccessKey pair to use SDK for Node.js. Example:

```
const FnFClient = require('@alicloud/fnf-2019-03-15');
async function demo() {
  const client = new FnFClient({
    // Manually enter the service endpoint. For more information, see API references > Requ
est syntax > Endpoints.
    endpoint: '{endpoint}',
    accessKeyId: 'xxx',
    accessKeySecret: 'xxx'
  });
  // Create a flow.
  const createResp = await client.createFlow ({
    Name : 'test',
    Definition :
`version:
type: flow
steps:
  - type: pass
    name: pass1`,
      Description : 'test',
    Type : 'FDL'
  });
  console.log("create: %s", createResp)
  // Start an execution.
  const startResp = await client.startExecution ({
    FlowName: 'testabc'
  });
  console.log("start: %s", startResp)
  // Query the execution result.
  const getResultResp = await client.getExecutionHistory ({
    FlowName: 'test',
    ExecutionName: 'xxx'
  });
  console.log("start: %s", getResultResp)
  // Update the flow.
  const res = await client.updateFlow ({
    Name : 'test',
    Definition :
`version:
type: flow
steps:
  - type: pass
    name: pass2`
  });
  console.log("%s", res)
}
demo();
```

# 5.SDK for PHP

This topic describes how to use SDK for PHP.

## Installation

Run the following composer command to install SDK for PHP:

```
composer require alibabacloud/fnf
```

If the download speed is slow, you can switch to the composer image in China.

```
composer config repo.packagist composer https://mirrors.aliyun.com/composer/
```

Alibaba Cloud SDK for PHP is also released at Github. For more information, see README-zh-CN.

## Quick start

You must have an Alibaba Cloud account and an AccessKey pair to use SDK for PHP. Example:

```
<? php
// If you use composer to download and debug dependencies locally, add the following refere
nce:
require 'vendor/autoload.php';
use AlibabaCloud\Client\AlibabaCloud;
use AlibabaCloud\Client\Exception\ClientException;
use AlibabaCloud\Client\Exception\ServerException;
use AlibabaCloud\Fnf\Fnf;
// Set the global client.
AlibabaCloud::accessKeyClient('foo', 'bar')
            ->regionId('cn-hangzhou')
            ->asDefaultClient();
try {
    // Access API operations of the service.
    $request1 = Fnf::v20190315()->createFlow();
    // Set options and parameters and execute requests.
    // Create a flow.
    $result1 = $request1->withDefinition('xxx') // API parameters
                                    ->withName('xxx')
                                    ->withRoleArn('xxx')
                                    ->withDescription('xxx')
                                    ->withType('FDL')
                                    //->scheme("https")
                                    //->client('client1') // Specifies the sending client. O
therwise, the global client is used.
                                    ->debug(true) // Outputs details. If this field is set t
o false, details are not output.
                                    ->connectTimeout(10) // Throws an exception when the con
nection times out.
                                    ->timeout(10) // Throws an exception upon timeout.
                                    ->request(); // Executes requests.
    // Start an execution.
    // The following example shows how to use options to pass in parameters related to SDK
```

```
calls.
    // You can also use withxxx to call SDK for PHP by referring to $request1.
    $options = [
                    'debug'=>true,
                    'connect_timeout'=>10,
                    'timeout'=>10,
                    'form_params'=>[
                        'FlowName'=>'xxx',
                        'Input'=>'{"execID": "exe1"}',
                    ],
                ];
    $result2 = Fnf::v20190315()
                                    ->startExecution($options)
                                    ->options([
                                        'form_params'=>[
                                            'Input'=>'Overwrites the value of the Input para
meter in options, in json format'.
                                        ],
                                    ])
                                    ->debug(false) // The last call will overwrite the forme
r ones.
                                    ->request();
    // Query the execution result.
    $request3 = Fnf::v20190315()->getExecutionHistory();
    // Set options and parameters and execute requests.
    // Create a flow.
    $result3 = $request3->withFlowName('xxx') // API parameters
                                    ->withExecutionName('xxx')
                                    ->debug(true) // Outputs details. If this field is set t
o false, details are not output.
                                    ->connectTimeout(10) // Throws an exception when the con
nection times out.
                                    ->timeout(10) // Throws an exception upon timeout.
                                    ->request(); // Executes requests.
} catch (ClientException $exception) {
    echo $exception->getMessage().PHP_EOL;
} catch (ServerException $exception) {
    echo $exception->getMessage().PHP_EOL;
    echo $exception->getErrorCode().PHP_EOL;
    echo $exception->getRequestId().PHP_EOL;
    echo $exception->getErrorMessage().PHP_EOL;
}
```

# 6.SDK for Python

This topic describes how to use SDK for Python.

## Prerequisites

1. You must have an Alibaba Cloud account and an AccessKey pair ( `AccessKey ID` and `AccessKey secret` ) to use Alibaba Cloud SDK for Python. You can log on to the Security Management page to create or view the AccessKey pair. You can also contact your system administrator to obtain the AccessKey pair.
2. To use Alibaba Cloud SDK to access API operations of a service, you must activate this service in the Alibaba Cloud Management Console.

## Installation

We recommend that you run the following pip commands to install SDK for Python:

```
pip install aliyun-python-sdk-core # Install the Alibaba Cloud SDK core library.
pip install aliyun-python-sdk-fnf # Install Serverless Workflow SDK.
```

## Quick start

The following example shows how to use SDK for Python to call . In this example, a flow is created, an execution is initiated, and the execution details are queried.

Request method

```
# encoding: utf-8
import time
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkfnf.request.v20190315 import CreateFlowRequest
from aliyunsdkfnf.request.v20190315 import StartExecutionRequest
from aliyunsdkfnf.request.v20190315 import DescribeExecutionRequest
from aliyunsdkfnf.request.v20190315 import GetExecutionHistoryRequest
flow_definition_type = "FDL"
flow_name = "xxx"
flow_definition = "xxx"
flow_description = "some descriptions"
role_arn = "acs:ram::${Your_Account_ID}:${Your_Role}"
execution_name = "xxx"
def create_flow(fnf_cli):
    request = CreateFlowRequest.CreateFlowRequest()
    request.set_Type(flow_definition_type)
    request.set_Description(flow_description)
    request.set_Definition(flow_definition)
    request.set_RoleArn(role_arn)
    request.set_Name(flow_name)
    return fnf_cli.do_action_with_exception(request)
def start_execution(fnf_cli):
    request = StartExecutionRequest.StartExecutionRequest()
    request.set_FlowName(flow_name)
    request.set_ExecutionName(execution_name)
    return fnf_cli.do_action_with_exception(request)
def describe_execution(fnf_cli):
    request = DescribeExecutionRequest.DescribeExecutionRequest()
    request.set_FlowName(flow_name)
    request.set_ExecutionName(execution_name)
    return fnf_cli.do_action_with_exception(request)
def get_execution_history(fnf_cli):
    request = GetExecutionHistoryRequest.GetExecutionHistoryRequest()
    request.set_FlowName(flow_name)
    request.set_ExecutionName(execution_name)
    return fnf_cli.do_action_with_exception(request)
```

Create a client and use the preceding functions to initiate a series of calls.

> **Note** If you need to implement debugging without modification, place the following function in the same file as the preceding request method code block to avoid errors upon import.

```
def main():
    # Create an AcsClient instance.
    client = AcsClient(
        "<your-access-key-id>",
        "<your-access-key-secret>",
        "<your-region-id>"
    )
    try:
        create_resp = create_flow(client)
        print(create_resp)
        start_resp = start_execution(client)
        print(start_resp)
        time.sleep(1)
        desc_resp = describe_execution(client)
        print(desc_resp)
        get_resp = get_execution_history(client)
        print(get_resp)
    except ServerException as e:
        print(e.get_request_id())
if __name__ == '__main__':
    main()
```