Function Compute

Developer Guide

MORE THAN JUST CLOUD | Alibaba Cloud

Developer Guide

Programming guide

Concepts

Handler

When creating a function, you must specify an event handler function that can be called by Function Compute (FC) during code running. The signature of the Node.js handler function is:

```
exports.myHandler = function(event, context, callback) {
...

// Use callback() to return information to the caller.
}
```

myHandler is the name of the function called by FC. You must export this function so that it is visible to FC. For example, if your logic entrance is in helloworld.js, helloworld.myHandler ({filename}.{handlername}) is the function handler. For more information, see Create a function.

- event: event playload in Buffer type.
- context: object that contains useful runtime information of the called function which is in Context type.
- callback: function that informs FC that the function call can be stopped and returns data to callers.

Function example:

```
exports.myHandler = function(event, context, callback) {
```

```
// Assume that the given event is a JSON string.
var eventObj = JSON.parse(event.toString());

console.log("value1 = " + eventObj.key1);
console.log("value2 = " + eventObj.key2);

// The first callback parameter is of the error type and null indicates that the function execution is successful.
callback(null, "some success message");
// or
// callback("some error type");
}
```

The preceding example shows an event handling function. The given event is converted to a JSON object first, and console.log() statement, and then stores the event data to Alibaba Cloud Log Service. When callback is called, the function exits, and “some success message” is returned.

Event format

When a function is called, FC passes the Body of InvokeFunction request to the function. FC does not restrict the event format. You can select a format such as JSON, XML, Image, or Video based on your business requirements. The interpretation of the event is completely controlled by developers.

The following example shows that if you use an OSS trigger, then the event is in JSON format.

```
{
"events": [
{
"eventName": "ObjectCreated:PutObject",
"eventSource": "acs:oss",
"eventTime": "2017-04-21T12:46:37.000Z",
"eventVersion": "1.0",
"oss": {
"bucket": {
"arn": "acs:oss:cn-shanghai:123456789:bucketname",
"name": "testbucket",
"ownerIdentity": "123456789",
"virtualBucket": ""
},
"object": {
"deltaSize": 122539,
"eTag": "688A7BF4F233DC9C88A80BF985AB7329",
"key": "image/a.jpg",
"size": 122539
},
"ossSchemaVersion": "1.0",
"ruleId": "9adac8e253828f4f7c0466d941fa3db81161e853"
},
"region": "cn-shanghai",
"requestParameters": {
"sourceIPAddress": "140.205.128.221"
},
"responseElements": {
"requestId": "58F9FF2D3DF792092E12044C"
```

```
},
"userIdentity": {
"principalId": "123456789"
}
}
]
}
```

Callback usage

The callback parameter is used to return the function execution result to the caller. The callback function signature is defined as follows:

```
callback(Error error, Object result)
```

When callback is called, the FC terminates the call.

error: An optional parameter which indicates that the function fails and returns the error information to the caller.

result: An optional parameter that returns the successful function execution result to the caller. When the **error** parameter is passed into callback, this parameter is ignored.

Example of callback usage.

```
callback();                          // function execution is successful with no information to return.
callback(null); // function execution is successful with no information to return.
callback(null, null); // function execution is successful with no information to return.
callback(null, "success"); // function execution is successful with a "success" string as returned value.
callback(new Error("error")); // function execution fails with an "error" message as returned value.
callback(new Error("error"), "ignored message") // function execution fails with an "error" message and the second parameter is ignored.
callback("error"); // function execution fails with an "error" message
```

Note:

- You must call the callback function to inform FC that function execution is completed. Otherwise, the function execution cannot be finished until a timeout error occurs.
- When an error message is returned using the callback error parameter, FC automatically records the error message in the logstore that is associated to the service. If the size of the error message is larger than 256 KB, only the first 256 KB of data is stored and "Truncated by FunctionCompute" is displayed. For more information about how to configure function log, see relevant documentation.
- If the function is called synchronously, the error message or result data is added to the HTTP body and returned to the client. If the function is called asynchronously, the error

message or result data is not returned.

Context

A context object is provided by the Function Compute for users to obtain useful runtime information. For example, you can obtain the ID of the current request and record it in the running log. Thus, you use it to track the details of request execution.

Context object (Node.js)

```
{
// Request ID of the current request
requestId: "...",

// Credentials contains your identity verification information that is required for you to access other services.
// For example, you need them to access data in the specified OSS directory.
// You must explicitly assign roles to the service on creation to use them.
// For more information, see the permission management documentation.
credentials: {
accessKeyId: "...",
accessKeySecret: "...",
securityToken: "..."
},

// Related function information
function: {
name: "...",
handler: "...",
memory: 128, // Integer, in MB.
timeout: 60 // Integer, in seconds.
}
}
```

Note: Currently, all attributes of the context objects are read-only.

Error message

Two types of errors may occur during a function operation:

- Handled error: an error proactively captured by the user function and returned through callback.
- Unhandled error: an error that is not captured by the user function, such as a syntax error, request timeout, out of memory and so on.

Function Compute encodes errors into JSON strings and returns them to the caller.

Handled error

When user function captures an error and returns it through callback(err), if the err parameter is an object whose type is Error, the stack trace is also returned.

```
// User function.
exports.handler = function(event, context, callback) {
var error = new Error("something is wrong");
callback(error);
};

// Function response.
{
"errorMessage": "something is wrong",
"errorType": "Error",
"stackTrace": [
"export.handler (/var/task/index.js.3:16)"
]
}
```

If the callback parameter is a non-error object, the stack trace is not returned.

```
// User function.
exports.handler = function(event, context, callback) {
callback("something is wrong");
};

// Function response.
{
"errorMessage": "something is wrong"
}
```

Unhandled error

When the user function does not capture any errors, Function Compute tries to capture errors as far as possible and returns the detailed information. If an error fails to be captured by the system, for example, a user function crashes unexpectedly and exits during execution, the system returns a general error.

Typical error scenarios

Error type	Scenario	Error information

Handled	An error returned through callback(err), in which the type of err is error.	{ “errorMessage” : “oops” , “errorType” : “Error” , “stackTrace” : [“exports.handler (/var/task/index.js:5:14)”]}
Handled	An error returned through callback(obj), in which obj is a non-error object.	{ “errorMessage” : “oops” }
Unhandled	The function reference does not exist.	{ “errorMessage” : “Cannot find module ‘/code/X.js” }
Unhandled	The specified handler function does not exist.	{ “errorMessage” : “Handler ‘X’ missing on module ‘Y’ ” }
Unhandled	The function execution times out.	{ “errorMessage” : “Task timed out after X seconds” }
Unhandled	An error related to and handled by another user program.	{ “errorMessage” : “Process exited unexpectedly before completing request” }

Logging

You can use the following codes in function to record logs at different levels:

- console.log()
- console.error()
- console.warn()
- console.info()

Log format

<Timestamp> <Request ID> <Log Level> <Message>

For example, when you run the following code:

```
console.log('hello world');
```

The log output is:

```
2016-08-18T05:38:41.401Z 1234567890 verbose hello world
```

If you have set a Log Service logstore for the service of the function, the “hello world” log from the

preceding example is collected to the configured logstore.

View logs

By using command line tool fcli, you can conveniently view the logs at service or function level. Try the following commands in fcli shell mode:

- To view logs of a service dimension, run logs service_name.
- To view logs of a function dimension, run logs service_name/function_name.
- You can also view the logs generated during a certain period. Run logs --help for more information.

For more information about how to enable service/function log by using fcli, see relevant documentation.

API reference

Request parameter

Request headers

Parameter name	Description	Required
Host	Format: ${accountID}.${region}.fc.aliyuncs.com, accountID is your Alibaba Cloud account ID, and region is the region where your Function Compute instance locates. For a detailed example, see service entry.	Yes
Date	Request time, in UTC format, for example: Mon, 08 May 2017 02:20:49 UTC.	Yes
Authorization	Signature authorization string. The format is FC ${accessKeyID}:${signature}. For example: FC LTAIJZbrgNpFmhrr:HeAESktL6PiPGDmZVlpSz5Xe7Zr40JjM	Yes

	ulNL5qTyr5M=. For more information about how to generate the string, see signature authentication.	
Content-Type	MIME type of the request content. For an InvokeFunction request, the type can be application/json or application/octet-stream. For other requests, the type can only be application/json.	Yes
Content-Length	Length of the request content.	Yes
Content-MD5	MD5 value of the request content.	No

Response headers

Parameter name	Description
X-Fc-Request-Id	Request ID of this request. It is critical for troubleshooting.
Date	Request response time, in UTC format.
Content-Type	MIME type of the response content. For an InvokeFunction request, the type is application/octet-stream. For other requests, the type is application/json.
Content-Length	Length of the response content.

Authorization

For every request, Function Compute (FC) checks the Authorization field in the request headers for the request authorization. Only requests from clients that use the same signature algorithm as the FC server can pass verification. For a request that neither contains a signature header nor has a valid signature, Function Compute returns a HTTP 403 error.

Signature algorithm

```
signature = base64(hmac-sha256(HTTP_METHOD + "\n"
```

```
+ CONTENT-MD5 + "\n"
+ CONTENT-TYPE + "\n"
+ DATE + "\n"
+ CanonicalizedFCHeaders
+ CanonicalizedResource))

Authorization = "FC " + accessKeyID + ":" + signature
```

- HTTP_METHOD indicates the HTTP Method in uppercase (such as PUT, GET, POST, and DELETE).
- CONTENT-MD5 indicates the MD5 value of the request content. If the request header does not contain Content-MD5, leave this field blank.
- CONTENT-TYPE indicates the request content type.
- DATE indicates the time of request occurred. It cannot be blank and only currently supports the GMT format.
    - Note: The difference between the DATE that is provided by clients and the system time of the FC server processing the request must be less than 15 minutes. Otherwise, FC rejects the request.
- CanonicalizedFCHeaders indicates a string that consists of all the HTTP headers prefixed with x-fc-. The following section describes how the string is generated.
- CanonicalizedResource indicates the request URL path, for example, /2016-08-15/services/my-service/functions?limit=100.
- hmac-sha256 uses your AccessKeySecret as its key.

CanonicalizedFCHeaders

Follow these steps to generate CanonicalizedFCHeaders:

1. Locate all the fields starting with x-fc- in request headers (case insensitive).
    - For fields that match the specified prefix, convert the field names into lowercase letters, and then sort these field names by ascending order.
2. Generate a string ${key}:${value}\n for each field,
    - ${key} is the key of the HTTP header (in lowercase).
    - ${value} is the value of the HTTP header.
    - For example: X-Fc-Invocation-Type: Sync is converted to x-fc-invocation-type:Sync\n.
3. Combine all previously generated strings into an new string.

The following pseudocode details CanonicalizedFCHeaders generation:

```
// javascript
// prefix = 'x-fc-'
function buildCanonicalHeaders(headers, prefix) {
var list = [];
var keys = Object.keys(headers);
for (let i = 0; i < keys.length; i++) {
var key = keys[i];
```

```
if (key.startsWith(prefix)) {
list.push(key);
}
}
list.sort();

var canonical = '';
for (let i = 0; i < list.length; i++) {
const key = list[i];
canonical += `${key}:${headers[key]}\n`;
}

return canonical;
}
```

Sample request

Request:

```
GET /2016-08-15/services?limit=100&nextToken=&prefix=&startKey= HTTP/1.1
Host: 1237050315505682.fc.cn-shanghai.aliyuncs.com
User-Agent: go-sdk-0.1
Accept: application/json
Authorization: FC LTAIUyt0Yeq1rgqo:GBmoz6OwC7bobTlD1jboBZ9PkaZ1e4cKsQ+5/dlLTns=
Date: Mon, 08 May 2017 03:08:31 GMT
X-User-Agent: go-resty v0.11 - https://github.com/go-resty/resty
Accept-Encoding: gzip
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
X-Fc-Request-Id: ab7c7602-0922-f04f-b4ee-923cd7df7fb0
Date: Mon, 08 May 2017 03:08:31 GMT
Transfer-Encoding: chunked
```

Sample code

You can also see released SDKs for FC signature generation:

- fc-nodejs-sdk

API specification

Alibaba Cloud Function Compute API Specification

Overview

Version information

Version : 0.1

URI scheme

Host : $account-id.$region.fc.aliyuncs.com*BasePath* : /2016-08-15*Schemes* : HTTP, HTTPS

Tags

- Service : Service operations
- Function : Function operations
- Trigger : Trigger operations

Consumes

- application/json

Produces

- application/json

Resources

Service

Service operations

CreateService

ListServices

GetService

UpdateService

DeleteService

Function

Function operations

CreateFunction

ListFunctions

GetFunction

UpdateFunction

DeleteFunction

GetFunctionCode

InvokeFunction

Trigger

Trigger operations

CreateTrigger

ListTriggers

GetTrigger

UpdateTrigger

DeleteTrigger

Operations

CreateFunction

```
POST /services/{serviceName}/functions
```

Description

Create a function

Parameters

Type	Name	Description	Schema
Header	**Host** *required*	Host name, $account-id.$region.fc.aliyuncs.com	string
Path	**serviceName** *required*	Service name	string
Body	**body** *required*	Function resource	Function

Responses

HTTP Code	Description	Schema
200	Successfully created the resource. **Headers** : ETag (string) : Ensures that the function that will be updated or deleted matches the etag.	FunctionResponse
400	Invalid arguments. Error code is InvalidArgument.	Error
403	Denied requests. Error code is AccessDenied , InvalidAccessKeyId , SignatureNotMatch , RequestTimeTooSkewed.	Error

404	Service does not exist. Error code is ServiceNotFound.	Error
409	Function already exists. Error code is FunctionAlreadyExists.	Error
500	Internal server error. Error code is InternalServerError.	Error

CreateService

```
POST /services
```

Description

create service

Parameters

Type	Name	Description	Schema
Header	Host *required*	Host name, $account-id.$region.fc.aliyuncs.com	string
Body	body *required*	Service resource	Service

Responses

HTTP Code	Description	Schema
200	Successfully created the resource. **Headers** : ETag (string) : Ensures that the service that will be updated or deleted matches the etag.	ServiceResponse
400	Invalid arguments. Error code is InvalidArgument.	Error
403	Denied requests. Error code is AccessDenied , InvalidAccessKeyId , SignatureNotMatch , RequestTimeTooSkewed.	Error
409	Service already exists. Error code is ServiceAlreadyExists.	Error
500	Internal server error. Error	Error

	code is InternalServerError.	

CreateTrigger

```
POST /services/{serviceName}/functions/{functionName}/triggers
```

Description

Create a trigger

Parameters

Type	Name	Description	Schema
Header	**Host** *required*	Host name, $account-id.$region.fc.aliyuncs.com	string
Path	**functionName** *required*	Function name	string
Path	**serviceName** *required*	Service name	string
Body	**body** *required*	Trigger resource	Trigger

Responses

HTTP Code	Description	Schema
200	Successfully created the resource **Headers** : ETag (string) : Ensures that the trigger that will be updated or deleted matches the etag.	TriggerResponse
400	Invalid arguments. Error code is InvalidArgument.	Error
403	Denied requests. Error code is AccessDenied , InvalidAccessKeyId , SignatureNotMatch , RequestTimeTooSkewed.	Error
404	Service or function does not exist. Error code is ServiceNotFound , FunctionNotFound.	Error
409	Trigger already exists. Error	Error

	code is TriggerAlreadyExists.	
500	Internal server error. Error code is InternalServerError.	Error

DeleteFunction

```
DELETE /services/{serviceName}/functions/{functionName}
```

Description

Deletes the function

Parameters

Type	Name	Description	Schema
Header	**Host** *required*	Host name, $account-id.$region.fc.aliyuncs.com	string
Header	**If-Match** *optional*	Ensures the resource that’ s to be updated is the requested one. The value is included in the Create/Get/Update Service/Function/Trigger API response.	string
Path	**functionName** *required*	Function name	string
Path	**serviceName** *required*	Service name	string

Responses

HTTP Code	Description	Schema
204	Successfully deleted the requested resource.	No Content
400	Invalid arguments. Error code is InvalidArgument.	Error
403	Denied requests. Error code is AccessDenied，InvalidAccessKeyId，SignatureNotMatch，RequestTimeTooSkewed.	Error
404	Service or function does not	Error

	exist. Error code is ServiceNotFound，Function NotFound.	
409	Function has at least one trigger. Error code is FunctionNotEmpty.	Error
500	Internal server error. Error code is InternalServerError.	Error

DeleteService

```
DELETE /services/{serviceName}
```

Description

Deletes the service

Parameters

Type	Name	Description	Schema
Header	**Host** *required*	Host name, $account-id.$region.fc.aliyuncs.com	string
Header	**If-Match** *optional*	Ensures the resource that’ s to be updated is the requested one. The value is included in the Create/Get/Update Service/Function/Trigger API response.	string
Path	**serviceName** *required*	Service name	string

Responses

HTTP Code	Description	Schema
204	Successfully deleted the requested resource.	No Content
400	Invalid arguments. Error code is InvalidArgument.	Error
403	Denied requests. Error code is AccessDenied，InvalidAccessKeyId，SignatureNotMatch，	Error

	RequestTimeTooSkewed.	
404	Service does not exist. Error code is ServiceNotFound.	Error
409	Service has at least one function. Error code is ServiceNotEmpty.	Error
500	Internal server error. Error code is InternalServerError.	Error

DeleteTrigger

```
DELETE /services/{serviceName}/functions/{functionName}/triggers/{triggerName}
```

Description

Deletes the trigger

Parameters

Type	Name	Description	Schema
Header	**Host** *required*	Host name, $account-id.$region.fc.aliyuncs.com	string
Header	**If-Match** *optional*	Ensures the resource that' s to be updated is the requested one. The value is included in the Create/Get/Update Service/Function/Trigger API response.	string
Path	**functionName** *required*	Function name	string
Path	**serviceName** *required*	Service name	string
Path	**triggerName** *required*	Trigger name	string

Responses

HTTP Code	Description	Schema
204	Successfully deleted the requested resource.	No Content

400	Invalid arguments. Error code is InvalidArgument.	Error
403	Denied requests. Error code is AccessDenied , InvalidAccessKeyId , SignatureNotMatch , RequestTimeTooSkewed.	Error
404	Service , function , trigger does not exist. Error code is ServiceNotFound , FunctionNotFound , TriggerNotFound.	Error
500	Internal server error. Error code is InternalServerError.	Error

GetFunction

```
GET /services/{serviceName}/functions/{functionName}
```

Description

Get function resource

Parameters

Type	Name	Description	Schema
Header	**Host** *required*	Host name, $account-id.$region.fc.aliyuncs.com	string
Path	**functionName** *required*	Function name	string
Path	**serviceName** *required*	Service name	string

Responses

HTTP Code	Description	Schema
200	Successfully retrieved the function **Headers** : ETag (string) : Ensures that the function that will be updated or deleted matches the etag.	FunctionResponse
400	Invalid arguments. Error code is InvalidArgument.	Error

403	Denied requests. Error code is AccessDenied , InvalidAccessKeyId , SignatureNotMatch , RequestTimeTooSkewed.	Error
404	Service or function does not exist. Error code is ServiceNotFound , FunctionNotFound.	Error
500	Internal server error. Error code is InternalServerError.	Error

GetFunctionCode

```
GET /services/{serviceName}/functions/{functionName}/code
```

Description

Get function code

Parameters

Type	Name	Description	Schema
Header	**Host** *required*	Host name, $account-id.$region.fc.aliyuncs.com	string
Path	**functionName** *required*	Function name	string
Path	**serviceName** *required*	Service name	string

Responses

HTTP Code	Description	Schema
200	Successfully retrieved function	FunctionCodeResponse
403	Denied requests. Error code is AccessDenied , InvalidAccessKeyId , SignatureNotMatch , RequestTimeTooSkewed.	Error
404	Service or function does not exist. Error code is ServiceNotFound , FunctionNotFound.	Error

500	Internal server error. Error code is InternalServerError.	Error

GetService

```
GET /services/{serviceName}
```

Description

Get service information

Parameters

Type	Name	Description	Schema
Header	**Host** *required*	Host name, $account-id.$region.fc.aliyuncs.com	string
Path	**serviceName** *required*	Service name	string

Responses

HTTP Code	Description	Schema
200	Successfully retrieved the service **Headers** : ETag (string) : Ensures that the service that will be updated or deleted matches the etag.	ServiceResponse
400	Invalid arguments. Error code is InvalidArgument.	Error
403	Denied requests. Error code is AccessDenied , InvalidAccessKeyId , SignatureNotMatch , RequestTimeTooSkewed.	Error
404	Service does not exist. Error code is ServiceNotFound.	Error
500	Internal server error. Error code is InternalServerError.	Error

GetTrigger

```
GET /services/{serviceName}/functions/{functionName}/triggers/{triggerName}
```

Description

Get trigger resource

Parameters

Type	Name	Description	Schema
Header	**Host** *required*	Host name, $account-id.$region.fc.aliyuncs.com	string
Path	**functionName** *required*	Function name	string
Path	**serviceName** *required*	Service name	string
Path	**triggerName** *required*	Trigger name	string

Responses

HTTP Code	Description	Schema
200	Successfully retrieved trigger **Headers** : ETag (string) : Ensures that the trigger that will be updated or deleted matches the etag.	TriggerResponse
400	Invalid arguments. Error code is InvalidArgument.	Error
403	Denied requests. Error code is AccessDenied , InvalidAccessKeyId , SignatureNotMatch , RequestTimeTooSkewed.	Error
404	Service , function , trigger does not exist. Error code is ServiceNotFound , FunctionNotFound , TriggerNotFound.	Error
500	Internal server error. Error code is InternalServerError.	Error

InvokeFunction

```
POST /services/{serviceName}/functions/{functionName}/invocations
```

Description

Invoke function synchronously or asynchronously

Parameters

Type	Name	Description	Schema	Default
Header	**Host** *required*	Host name, $account-id.$region.fc.aliyuncs.com	string	
Header	**X-Fc-Invocation-Type** *optional*	Invocation type, Sync or Async. Defaults to Sync.	string	"Sync"
Path	**functionName** *required*	Function name	string	
Path	**serviceName** *required*	Service name	string	
Body	**body** *required*	Event，binary type. This value is passed to function without any transformation. It’s function’s responsibility to interpret the value.	object	

Responses

HTTP Code	Description	Schema
200	Successfully invoked function synchronously. **Headers** : X-Fc-Error-Type (string) : Error type, including HandledInvocationError and UnhandledInvocationError.	InvokeResponse
202	The async invocation request has been accepted, and the corresponding function will be invoked as soon as possible.	No Content
403	Denied requests. Error code	Error

	is AccessDenied , InvalidAccessKeyId , SignatureNotMatch , RequestTimeTooSkewed.	
404	Service or function does not exist. Error code is ServiceNotFound , FunctionNotFound.	Error
500	Internal server error. Error code is InternalServerError.	Error

ListFunctions

```
GET /services/{serviceName}/functions
```

Description

Retrieve function records

Parameters

Type	Name	Description	Schema	Default
Header	**Host** *required*	Host name, $account-id.$region.fc.aliyuncs.com	string	
Path	**serviceName** *required*	Service name	string	
Query	**limit** *optional*	Limits the number of returned resource records. Defaults to 20 and cannot exceed 100.	integer (int32)	20
Query	**nextToken** *optional*	Is used to query more resource records. The token is included in the List API response.	string	
Query	**prefix** *optional*	Limits the resource names that begin with the specified	string	

		prefix.		
Query	**startKey** *optional*	Specifies the resource name where to start the query.	string	

Responses

HTTP Code	Description	Schema
200	Successfully retrieved functions	Response 200
400	Invalid arguments. Error code is InvalidArgument.	Error
403	Denied requests. Error code is AccessDenied , InvalidAccessKeyId , SignatureNotMatch , RequestTimeTooSkewed.	Error
404	Service does not exist. Error code is ServiceNotFound.	Error
500	Internal server error. Error code is InternalServerError.	Error

Response 200

Name	Schema
functions *optional*	<**functions** > array
nextToken *optional*	NextToken

functions

Name	Description	Schema
codeChecksum *optional*	The checksum (crc64) of the function code. **Example** : "5434025278388143772"	string
codeSize *optional*	The size of the function code, in bytes. **Example** : 1024	integer (int64)
createdTime *optional*	The UTC time string indicating the time the function was created **Example** : "2016-08-15T15:00:00.000+0000"	string

description *optional*	**Example** : "This is a demo hello world function."	string
functionId *optional*	A unique id generated by Function Compute to identify a function. **Example** : "2d28e0e9-9ba5-4eed-8b1a-d3d9cd24e737"	string
functionName *optional*	**Example** : "helloworld"	string
handler *optional*	The function execution entry point. **Example** : "hello_world.main"	string
lastModifiedTime *optional*	The UTC time string indicating the last time the function was updated **Example** : "2016-08-15T17:00:00.000+0000"	string
memorySize *optional*	The amount of memory that’ s used to execute function, in MB. Function Compute uses this value to allocate CPU resources proportionally. **Example** : 512	integer (int32)
runtime *optional*	The function runtime environment. Supporting nodejs6, nodejs8, python2.7, python3, java8 **Example** : "nodejs4.4"	string
timeout *optional*	The maximum time duration a function can execute, in seconds. After which Function Compute terminites the execution. **Example** : 10	integer (int32)

ListServices

```
GET /services
```

Description

Retrieve service records

Parameters

Type	Name	Description	Schema	Default
Header	**Host**	Host name,	string	

	required	$account-id.$region.fc.aliyuncs.com		
Query	**limit** *optional*	Limits the number of returned resource records. Defaults to 20 and cannot exceed 100.	integer (int32)	20
Query	**nextToken** *optional*	Is used to query more resource records. The token is included in the List API response.	string	
Query	**prefix** *optional*	Limits the resource names that begin with the specified prefix.	string	
Query	**startKey** *optional*	Specifies the resource name where to start the query.	string	

Responses

HTTP Code	Description	Schema
200	Successfully retrieved services	Response 200
400	Invalid arguments. Error code is InvalidArgument.	Error
403	Denied requests. Error code is AccessDenied , InvalidAccessKeyId , SignatureNotMatch , RequestTimeTooSkewed.	Error
500	Internal server error. Error code is InternalServerError.	Error

Response 200

Name	Schema
nextToken *optional*	NextToken

services *optional*	<services > array

services

Name	Description	Schema
createdTime *optional*	The UTC time string indicating the time the service was created. **Example** : "2016-08-15T16:06:05.000+0000"	string
description *optional*	Service description **Example** : "This is a demo service."	string
internetAccess *optional*	Set it true to enable internet access. **Example** : true	boolean
lastModifiedTime *optional*	The UTC time string indicating the last time the function was updated **Example** : "2016-08-16T18:00:00.000+0000"	string
logConfig *optional*	Log configuration. Function Compute pushes function execution logs to the configured log store.	LogConfig
role *optional*	The role grants Function Compute the permission to access user’ s cloud resources, such as pushing logs to user’ s log store. The temporary STS token generated from this role can be retrieved from function context and used to access cloud resources. **Example** : "acs:ram::1234567890:role/fc-test"	string
serviceId *optional*	A unique id generated by Function Compute to identify a service. **Example** : "2d28e0e9-9ba5-4eed-8b1a-d3d9cd24e737"	string
serviceName *optional*	**Example** : "demo-service"	string
vpcConfig *optional*	VPC configuration. Function Compute uses these information to setup ENI in the specific VPC.	VPCConfig

ListTriggers

```
GET /services/{serviceName}/functions/{functionName}/triggers
```

Description

Retrieve trigger records

Parameters

Type	Name	Description	Schema	Default
Header	**Host** *required*	Host name, $account-id.$region.fc.aliyuncs.com	string	
Path	**functionName** *required*	Function name	string	
Path	**serviceName** *required*	Service name	string	
Query	**limit** *optional*	Limits the number of returned resource records. Defaults to 20 and cannot exceed 100.	integer (int32)	20
Query	**nextToken** *optional*	Is used to query more resource records. The token is included in the List API response.	string	
Query	**prefix** *optional*	Limits the resource names that begin with the specified prefix.	string	
Query	**startKey** *optional*	Specifies the resource name where to start the query.	string	

Responses

HTTP Code	Description	Schema

200	Successfully retrieved functions	Response 200
400	Invalid arguments. Error code is InvalidArgument.	Error
403	Denied requests. Error code is AccessDenied , InvalidAccessKeyId , SignatureNotMatch , RequestTimeTooSkewed.	Error
404	Service or function does not exist. Error code is ServiceNotFound , FunctionNotFound.	Error
500	Internal server error. Error code is InternalServerError.	Error

Response 200

Name	Schema
nextToken *optional*	NextToken
triggers *optional*	<TriggerResponse > array

UpdateFunction

```
PUT /services/{serviceName}/functions/{functionName}
```

Description

Update the function

Parameters

Type	Name	Description	Schema
Header	**Host** *required*	Host name, $account-id.$region.fc.aliyuncs.com	string
Header	**If-Match** *optional*	Ensures the resource that’ s to be updated is the requested one. The value is included in the Create/Get/Update Service/Function/Tri	string

		gger API response.	
Path	**functionName** *required*	Function name	string
Path	**serviceName** *required*	Service name	string
Body	**function** *required*	Function object that’ s to be updated	FunctionUpdateFields

Responses

HTTP Code	Description	Schema
200	Successfully updated function **Headers** : ETag (string) : Ensures that the function that will be updated or deleted matches the etag.	FunctionResponse
400	Invalid arguments. Error code is InvalidArgument.	Error
403	Denied requests. Error code is AccessDenied , InvalidAccessKeyId , SignatureNotMatch , RequestTimeTooSkewed.	Error
404	Service or function does not exist. Error code is ServiceNotFound , FunctionNotFound.	Error
500	Internal server error. Error code is InternalServerError.	Error

UpdateService

```
PUT /services/{serviceName}
```

Description

Update the service

Parameters

Type	Name	Description	Schema
Header	**Host** *required*	Host name, $account-	string

		id.$region.fc.aliyuncs.com	
Header	**If-Match** *optional*	Ensures the resource that’ s to be updated is the requested one. The value is included in the Create/Get/Update Service/Function/Trigger API response.	string
Path	**serviceName** *required*	Service name	string
Body	**body** *required*	Service object that will be updated.	ServiceUpdateFields

Responses

HTTP Code	Description	Schema
200	Successfully updated the service **Headers** : ETag (string) : Ensures that the service that will be updated or deleted matches the etag.	ServiceResponse
400	Invalid arguments. Error code is InvalidArgument.	Error
403	Denied requests. Error code is AccessDenied , InvalidAccessKeyId , SignatureNotMatch , RequestTimeTooSkewed.	Error
404	Service does not exist. Error code is ServiceNotFound.	Error
500	Internal server error. Error code is InternalServerError.	Error

UpdateTrigger

```
PUT /services/{serviceName}/functions/{functionName}/triggers/{triggerName}
```

Description

Update the trigger

Parameters

Type	Name	Description	Schema
Header	**Host** *required*	Host name, $account-id.$region.fc.aliyuncs.com	string
Header	**If-Match** *optional*	Ensures the resource that’ s to be updated is the requested one. The value is included in the Create/Get/Update Service/Function/Trigger API response.	string
Path	**functionName** *required*	Function name	string
Path	**serviceName** *required*	Service name	string
Path	**triggerName** *required*	Trigger name	string
Body	**body** *required*	Trigger object that will be updated.	TriggerUpdateFields

Responses

HTTP Code	Description	Schema
200	Successfully updated the trigger **Headers** : ETag (string) : Ensures that the trigger that will be updated or deleted matches the etag.	TriggerResponse
400	Invalid arguments. Error code is InvalidArgument.	Error
403	Denied requests. Error code is AccessDenied , InvalidAccessKeyId , SignatureNotMatch , RequestTimeTooSkewed.	Error
404	Service , function , trigger does not exist. Error code is ServiceNotFound , FunctionNotFound , TriggerNotFound.	Error
500	Internal server error. Error	Error

	code is InternalServerError.	

Definitions

Code

Provides two ways to upload code 1. By specifying the OSS BucketName and ObjectName 2. By specifying zipFile with base64 encoded zip file content.

Name	Description	Schema
ossBucketName *optional*	OSS bucket name	string
ossObjectName *optional*	OSS object name	string
zipFile *optional*	Base64 encoded zip file content	string

Error

Error

Name	Description	Schema
errorCode *optional*	Error code	string
errorMessage *optional*	Error message	string

Function

Name	Description	Schema
code *required*	The code that contains the function implementation.	Code
description *optional*		string
functionName *required*	Function name	string
handler *required*	The function execution entry point.	string
memorySize	The amount of memory	integer

optional	that’ s used to execute function, in MB. Function Compute uses this value to allocate CPU resources proportionally. Defaults to 128MB. It should be multiple of 64 MB and between 128MB and 3072MB.	
runtime *required*	The function runtime environment. Supporting nodejs6, nodejs8, python2.7, python3, java8	string
timeout *optional*	The maximum time duration a function can execute, in seconds. After which Function Compute terminates the execution. Defaults to 3 seconds, and should be between 1 to 300 seconds.	integer

FunctionCodeResponse

Name	Description	Schema
checksum *optional*	crc64 checksum **Example** : "1234567890"	string
url *optional*	A pre-signed OSS url that can be used to download function code. **Example** : "http://func-code.oss-cn-shanghai.aliyuncs.com/1a2b3c4d5e6f"	string

FunctionResponse

Name	Description	Schema
codeChecksum *optional*	The checksum (crc64) of the function code. **Example** : "5434025278388143772"	string
codeSize *optional*	The size of the function code, in bytes. **Example** : 1024	integer (int64)
createdTime *optional*	The UTC time string indicating the time the function was created	string

	Example : "2016-08-15T15:00:00.000+0000"	
description *optional*	**Example** : "This is a demo hello world function."	string
functionId *optional*	A unique id generated by Function Compute to identify a function. **Example** : "2d28e0e9-9ba5-4eed-8b1a-d3d9cd24e737"	string
functionName *optional*	Function Name **Example** : "helloworld"	string
handler *optional*	The function execution entry point. **Example** : "hello_world.main"	string
lastModifiedTime *optional*	The UTC time string indicating the last time the function was updated **Example** : "2016-08-15T17:00:00.000+0000"	string
memorySize *optional*	The amount of memory that’ s used to execute function, in MB. Function Compute uses this value to allocate CPU resources proportionally. **Example** : 512	integer (int32)
runtime *optional*	The function runtime environment. Supporting nodejs6, nodejs8, python2.7, python3, java8. **Example** : "nodejs4.4"	string
timeout *optional*	The maximum time duration a function can execute, in seconds. After which Function Compute terminates the execution. **Example** : 10	integer (int32)

FunctionUpdateFields

Function update fields

Name	Description	Schema
code *optional*	The code that contains the function implementation.	Code
description *optional*		string

handler *optional*	The function execution entry point.	string
memorySize *optional*	The amount of memory that’ s used to execute function, in MB. Function Compute uses this value to allocate CPU resources proportionally. Defaults to 128MB. It should be multiple of 64 MB and between 128MB and 3072MB.	integer
runtime *optional*	The function runtime environment. Supporting nodejs6, nodejs8, python2.7, python3, java8	string
timeout *optional*	The maximum time duration function can execute, in seconds. After which Function Compute terminates the execution. Defaults to 3 seconds, and should be between 1 to 300 seconds.	integer

InvokeResponse

The response of invoke function. It’ s in binary format. Users should interpret the data according to their function implementation.

Type : binary

LogConfig

Log configuration

Name	Description	Schema
logstore *optional*	The log store name of Logs service	string
project *optional*	The project name of Logs service	string

NextToken

Is used to query more resource records.

Type : string

Service

Service resource

Name	Description	Schema
description *optional*	Service description	string
internetAccess *optional*	Set it true to enable internet access. **Example** : true	boolean
logConfig *optional*	Log configuration. Function Compute pushes function execution logs to the configured log store.	LogConfig
role *optional*	The role grants Function Compute the permission to access user’ s cloud resources, such as pushing logs to user’ s log store. The temporary STS token generated from this role can be retrieved from function context and used to access cloud resources. **Example** : "acs:ram::1234567890:role/fc-test"	string
serviceName *required*	Service name	string
vpcConfig *optional*	VPC configuration. Function Compute uses these information to setup ENI in the specific VPC.	VPCConfig

ServiceResponse

Name	Description	Schema
createdTime *optional*	The UTC time string indicating the time the service was created **Example** : "2016-08-15T16:06:05.000+0000"	string
description *optional*	Service description **Example** : "This is a demo service."	string

internetAccess *optional*	Set it true to enable internet access. **Example** : true	boolean
lastModifiedTime *optional*	The UTC time string indicating the last time the service was updated **Example** : "2016-08-16T18:00:00.000+0000"	string
logConfig *optional*	Log configuration. Function Compute pushes function execution logs to the configured log store.	LogConfig
role *optional*	The role grants Function Compute the permission to access user’ s cloud resources, such as pushing logs to user’ s log store. The temporary STS token generated from this role can be retrieved from function context and used to access cloud resources. **Example** : "acs:ram::1234567890:role/fc-test"	string
serviceId *optional*	A unique id generated by Function Compute to identify a service. **Example** : "2d28e0e9-9ba5-4eed-8b1a-d3d9cd24e737"	string
serviceName *optional*	**Example** : "demo-service"	string
vpcConfig *optional*	VPC configuration. Function Compute uses these information to setup ENI in the specific VPC.	VPCConfig

ServiceUpdateFields

Service resource that can be updated.

Name	Description	Schema
description *optional*		string
internetAccess *optional*	Set it true to enable internet access. **Example** : true	boolean
logConfig	Log configuration. Function	LogConfig

optional	Compute pushes function execution logs to the configured log store.	
role *optional*	The role grants Function Compute the permission to access user’ s cloud resources, such as pushing logs to user’ s log store. The temporary STS token generated from this role can be retrieved from function context and used to access cloud resources. **Example** : "acs:ram::1234567890:role/fc-test"	string
vpcConfig *optional*	VPC configuration. Function Compute uses these information to setup ENI in the specific VPC.	VPCConfig

Trigger

Name	Description	Schema
invocationRole *required*	The role grants event source the permission to invoke function on behalf of user. This is optional for some triggers. **Example** : "acs:ram::1234567890:role/fc-test"	string
sourceArn *required*	The Aliyun Resource Name (ARN) of event source. This is optional for some triggers. **Example** : "acs:oss:cn-shanghai:12345:mybucket"	string
triggerConfig *required*	Event source specific trigger configuration. The value is different according to trigger type.	object
triggerName *required*	Trigger name **Example** : "image_resize"	string
triggerType *required*	Trigger type, e.g. oss, timer, logs. This determines how the trigger config is interpreted. **Example** : "oss"	string

TriggerResponse

Name	Description	Schema
createdTime *optional*	The UTC time string indicating the time the trigger was created **Example** : "2016-08-15T15:00:00.000+0000"	string
invocationRole *optional*		string
lastModifiedTime *optional*	The UTC time string indicating the last time the trigger was updated **Example** : "2016-08-15T17:00:00.000+0000"	string
sourceArn *optional*		string
triggerConfig *optional*		object
triggerName *optional*		string
triggerType *optional*		string

TriggerUpdateFields

Name	Description	Schema
invocationRole *optional*	The role grants event source the permission to invoke function on behalf of user. This is optional for some triggers. **Example** : "acs:ram::1234567890:role/fc-test"	string
triggerConfig *optional*	Event source specific trigger configuration. The value is different according to trigger type.	object

VPCConfig

VPC configuration

Name	Description	Schema

securityGroupId *optional*	Security group ID	string
vSwitchIds *optional*	List of VSwitch IDs	<string > array
vpcId *optional*	VPC ID	string

SDK

SDK list

Language	Current version
nodejs	
java	
python	

SDKs for more languages are coming soon.

Tools

fcli

fcli is a powerful command line tool that helps you manage your Function Compute resources conveniently.

You can click here to download fcli. After the installation, you can run fcli shell to go to the shell

mode and complete the initial configurations.

```
fcli shell
Please input the endpoint (example: https://your_account_id.cn-shanghai.fc.aliyuncs.com):
https://123456.cn-shanghai.fc.aliyuncs.com
Please input the access key id:
fakeaccesskeyid
Please input the access key secret:
fakeaccesskeysecret
Store the configuration in: /Users/testuser/.fcli
Welcome to the function compute world. Have fun!
```

Note:

The account ID can be obtained in the console.

If you do not have a valid AccessKey, you can create one in RAM. You can use the AccessKey of the primary account, but this action is not recommended.

You can run the help command to view available commands.

```
>>> help

Commands:
attach attach the policy to a role
cd change the current resource
clear clear the screen
config config the fcli
detach detach the policy from a role
exit exit the program
grant grant the permission
help display help
info display the resource detail info
invk invoke the function
logs display the service/function logs
ls list the child resources of the current resource
mkf create the function
mkir create the invocation role
mkl create the log project and store
mkrp create the ram policy
mks create the service
mksr create the service role
mkt create the trigger
pwd display the current resource
rm delete the resource
sbox a sandbox environment for installing the 3rd party libararies and trouble shooting
upf update the function
ups update the service
upt update the trigger
```

For every command, you can run cmd --help to view detailed help information.

```
>>> mks --help
-d, --description string service description
--etag string service etag for update
--help
-p, --log-project string loghub project for logging
-l, --log-store string loghub logstore for logging
-r, --role string role arn for oss code copy, function execution and logging
```

In the Service Management, Function Management, and Trigger Management documents under the Developer guide, you can learn about more detailed descriptions and examples. If you have any questions, go to the project for a discussion.

fun

Fun is a tool used to support Function Compute and API Gateway. It helps you manage your Function Compute and API Gateway resources. It uses a resource configuration file (faas.yaml) to help you develop, create, and deploy functions with API gateway as its frontend.

For a Node.js user, this tool can automatically install and pack the references, which means you can still use package.json to manage your dependencies.

Installation

Although fun is a command line tool written in Node.js, it can also deploy functions written in other languages such as Python and Java. You can run the following npm command to install fun:

```
$ npm install @alicloud/fun -g
```

After installation, a command line tool called fun is created. You can enter this fun command to view the help information:

```
$ which fun
/Users/xxx/.tnvm/versions/node/v8.9.0/bin/fun
```

Usage

After the command line tool is installed, you can start to develop your application. To use the fun

tool, you must create a project directory and a faas.yaml file in it so that it can recognize this directory as a project.

You must define project-related configurations in this file. The information mainly consists of four parts:

- credentials
- role
- function compute
- api gateway

Note that the API Gateway settings are optional, which means that one can release a Function Compute without configuring API Gateway.

Credentials

The credential section is mainly used to list the owner identities as it includes accountid, AccessKeyId, and AccessKeySecret. As the information is sensitive, they can be imported as an environmental variable. This section of configuration is optional when the credentials are imported as an environmental variable.

The following provides a configuration example:

```
# faas.yaml
accountid: '<account id, default is process.env.ACCOUNT_ID>'
accessKeyId: '<ACCESS KEY ID, default is process.env.ACCESS_KEY_ID>'
accessKeySecret: '<ACCESS KEY SECRET, default is process.env.ACCESS_KEY_SECRET>'
```

Role

API Gateway must be granted the permission to access Function Compute.

The role configuration is as follows:

```
# faas.yaml
role:
name: 'apigatewayAccessFC'
```

Function compute

This part of the configuration is mainly used to describe the organization structure of Function Compute. The following example describes how to perform the configuration.

Create a hello.js file under the project directory:

```
'use strict';
```

```
const hook = require('fc-helper');

exports.index = hook((req, res) => {
res.send('Hello world!\n');
});
```

Click here to find the previous fc-helper module, and manage it using package.json.

Then, perform the following configuration:

```
# faas.yaml
function-compute:
region: 'cn-shanghai'
services:
- name: 'fc'
description: 'fc test'
functions:
- name: 'helloworld'
description: 'hello!'
# entry point
handler: hello.index
# which files should be ziped
codes:
- 'hello.js'
```

We omit the detailed description of each setting here as they are self-explanatory.

API gateway

This section is about how to set up API gateway so that one can call the functions through API gateway.

```
# faas.yaml
api-gateway:
endpoint: 'http://apigateway.cn-hangzhou.aliyuncs.com'
groups:
- name: 'apigw_fc'
description: 'API Gateway & Function Compute'
apis:
- name: 'helloworld'
method: 'GET'
path: '/helloworld'
function: 'cn-shanghai/fc/helloworld'
```

Deployment

Now we can deploy the code onto Function Compute with the configurations so far. The deployment operation only requires one command.

```
$ fun deploy
Function compute(region):
service service_name ok.
function function_name ok.
API gateway(region):
URL: POST http://<groupid>-<region>.alicloudapi.com/<the api path>
=> cn-hangzhou/service_name/function_name
stage: RELEASE, deployed, version: 20171101125034887
stage: PRE, deployed, version: 20171101125107780
stage: TEST, deployed, version: 20171101144618017
```

You can see that Function Compute is deployed. If the related settings of API Gateway are present, then fun also sets up an API gateway service to route any web traffic to the function. You can also add a stage parameter to the command to deploy the service to a test environment or pre-release environment:

```
$ fun deploy TEST
$ fun deploy PRE
```

Click here for a complete example.

By setting environment variables, you can release it to your own API Gateway and Function Compute environment.

Click here for more complicated examples.

Feedback

If you have any questions, create an issue.