

Alibaba Cloud Elastic Container Instance

SDK reference

Issue: 20200525









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 due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall 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 the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. 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 bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, 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 contact Alibaba Cloud directly if you discover any errors in this document.

Document conventions

Style	Description	Example
	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: If the weight is set to 0, the server no longer receives new requests.
	A note indicates supplemental instructions, best practices, tips, and other content.	 Note: 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 <code>cd /d C:/window</code> command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid Instance_ID</code>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>

Style	Description	Example
{ } or {a b}	This format is used for a required value, where only one item can be selected.	switch {active stand}

Contents

Legal disclaimer.....	I
Document conventions.....	I
1 Overview.....	1
2 Use the Go SDK.....	2
3 Use the Java SDK.....	8
4 Use the Python SDK.....	11

1 Overview

This topic describes the Elastic Container Instance (ECI) SDKs in different programming languages and lists the addresses for obtaining the latest SDK versions. Currently, ECI SDKs are available in the following languages:

- Go
- Python
- Java

SDK overview

Alibaba Cloud ECI SDKs encapsulate the API version released on August 8, 2018. The SDKs identify callers based on their AccessKey pairs and provides features such as automatic signature to allow you to create and manage resources by calling API operations.

To use ECI features, you must install both an ECI SDK and the Alibaba Cloud core library. For example, to use the Java SDK, you must download `aliyun-java-sdk-core` and `aliyun-java-sdk-eci`.

Supported SDKs

The following table lists the SDKs supported by ECI. You can view the SDK update history, obtain the installation package, and view the help document on GitHub.

Programming language	GitHub address	GitHub document
Java	aliyun-openapi-java-sdk	README.md
Python	aliyun-openapi-python-sdk	README.md
Go	alibaba-cloud-sdk-go	README.md

2 Use the Go SDK

[Download the Go SDK](#)

Sample code

const.go:

```
package eci

type ContainerGroupStatus string

const (
    Scheduling   = ContainerGroupStatus("Scheduling")
    Pending      = ContainerGroupStatus("Pending")
    Running      = ContainerGroupStatus("Running")
    Succeeded    = ContainerGroupStatus("Succeeded")
    Failed       = ContainerGroupStatus("Failed")
    Restarting   = ContainerGroupStatus("Restarting")
    Updating     = ContainerGroupStatus("Updating")
    ScheduleFailed = ContainerGroupStatus("ScheduleFailed")
)

type ContainerStatus string

const (
    waiting = ContainerStatus("Waiting")
    running  = ContainerStatus("Running")
    terminated = ContainerStatus("Terminated")
)
```

config.conf:

```
[eci_conf]
# The AccessKey pair of the test account.
access_key = xxx
secret_key = xxx

#region
region_id=cn-hangzhou

# The information about the resources in each region, in JSON format.
region_info={"cn-hangzhou": {"securityGroupId": "sg-bp118knl07ymorgzfhyw",
vSwitchId": "vsw-bp1jrgfqy54kg5hcf8bc", "zoneId": "cn-hangzhou-h"}, "cn-shanghai": {"securityGroupId": "sg-uf62jug0dt92nfy630cs", "vSwitchId": "vsw-uf6cb39ub5urw82bv3w70", "zoneId": "cn-shanghai-g"}, "cn-beijing": {"securityGroupId": "sg-2ze6uv80a8s4yk04dmyu", "vSwitchId": "vsw-2zegtxf8q29d8tt1xtcj", "zoneId": "cn-beijing-h"}, "cn-shenzhen": {"securityGroupId": "sg-wz96x4kv4edy97q6jjzn", "vSwitchId": "vsw-wz9rn33jwilodp24klo0q", "zoneId": "cn-shenzhen-a"}, "cn-hongkong": {"securityGroupId": "sg-j6cjajpf7f4ybda7h19w", "vSwitchId": "vsw-j6c9c9fit5owbbc0ids6c", "zoneId": "cn-hongkong-b"}, "us-west-1": {"securityGroupId": "sg-u19hbpj9x", "vSwitchId": "vsw-rj9h5vvzhhb4fjkuzuyi8j", "zoneId": "us-west-1a"}, "us-east-1": {"securityGroupId": "sg-0xih51sfoldxzfbsja5", "vSwitchId": "vsw-0xisefewxoqe4f849jbnl", "zoneId": "us-east-1b"}, "ap-southeast-1": {"securityGroupId": "sg-t4n23ppw7fm5to7cnib9", "vSwitchId": "vsw-t4nr1xt8zupxg00unnmg9", "zoneId": "ap-southeast-1b"}, "cn-zhangjiakou": {"securityGr
```

```
oupld": "sg-8vb96lku1dgq71ckh98g","vSwitchId": "vsw-8vbtnz5878r3bcio7cpyx", "nfs": "9cbb94a474-oav34.cn-zhangjiakou.nas.aliyuncs.com", "zoneId": "cn-zhangjiakou-a"}}}
```

The following code displays the `region_info` parameter in an easy-to-read format. Note that the settings are for reference only.

```
{
  "cn-hangzhou": {
    "securityGroupId": "sg-bp118knl07ymorgzfhyw",
    "vSwitchId": "vsw-bp1jrgfqy54kg5hcf8bc",
    "zoneId": "cn-hangzhou-h"
  },
  "cn-shanghai": {
    "securityGroupId": "sg-uf62jug0dt92nfy630cs",
    "vSwitchId": "vsw-uf6cb39ub5urw82bv3w70",
    "zoneId": "cn-shanghai-g"
  },
  "cn-beijing": {
    "securityGroupId": "sg-2ze6uv80a8s4yk04dmyu",
    "vSwitchId": "vsw-2zegtxf8q29d8tt1xtcj",
    "zoneId": "cn-beijing-h"
  },
  "cn-shenzhen": {
    "securityGroupId": "sg-wz96x4kv4edy97q6jjzn",
    "vSwitchId": "vsw-wz9rn33jwilodp24klo0q",
    "zoneId": "cn-shenzhen-a"
  },
  "cn-hongkong": {
    "securityGroupId": "sg-j6cjajpf7f4ybda7h19w",
    "vSwitchId": "vsw-j6c9c9fit5owbbc0ids6c",
    "zoneId": "cn-hongkong-b"
  },
  "us-west-1": {
    "securityGroupId": "sg-u19hbpj9x",
    "vSwitchId": "vsw-rj9h5vvzhh4fjkuzuyi8j",
    "zoneId": "us-west-1a"
  },
  "ap-southeast-1": {
    "securityGroupId": "sg-t4n23ppw7fm5to7cnib9",
    "vSwitchId": "vsw-t4nr1xt8zupxg00unnmg9",
    "zoneId": "ap-southeast-1b"
  },
  "cn-zhangjiakou": {
    "securityGroupId": "sg-8vb96lku1dgq71ckh98g",
    "vSwitchId": "vsw-8vbtnz5878r3bcio7cpyx",
    "zoneId": "cn-zhangjiakou-a"
  }
}
```

eci.go:

```
package eci

import (
  "encoding/json"
  "fmt"
  "github.com/Unknwon/goconfig"
  "github.com/aliyun/alibaba-cloud-sdk-go/sdk/requests"
  "github.com/aliyun/alibaba-cloud-sdk-go/services/eci"
  "math"
  "os"
```

```
"reflect"
"strconv"
"time"
)

var accessKey string
var secretKey string
var regionId string
var zoneId string
var securityGroupId string
var vSwitchId string

var client *eci.Client

/**
Obtain the configurations.
*/
func init() {
var cfg *goconfig.ConfigFile
config, err := goconfig.LoadConfigFile("./eci/config.conf") // Load the configuration file.
if err != nil {
fmt.Println("get config file error:", err.Error())
os.Exit(-1)
}
cfg = config

accessKey, _ = cfg.GetValue("eci_conf", "access_key")
secretKey, _ = cfg.GetValue("eci_conf", "secret_key")
regionId, _ = cfg.GetValue("eci_conf", "region_id")

var regionInfo map[string](map[string](string));
value, _ := cfg.GetValue("eci_conf", "region_info")

json.Unmarshal([]byte(value), &regionInfo)

zoneId = regionInfo[regionId]["zoneId"]
securityGroupId = regionInfo[regionId]["securityGroupId"]
vSwitchId = regionInfo[regionId]["vSwitchId"]

fmt.Printf("init success[ access_key:%s, secret_key:%s, region_id:%s, zoneId:%s,
vSwitchId:%s, securityGroupId:%s]\n",
accessKey, secretKey, regionId, zoneId, vSwitchId, securityGroupId)

// Initialize the client.
client, err = eci.NewClientWithAccessKey(regionId, accessKey, secretKey)
if err != nil {
panic(err)
}
}

/**
Create a container group.
*/
func createContainerGroup()(string) {
// Create Container Group
createContainerRequest := eci.CreateCreateContainerGroupRequest()
// required
createContainerRequest.RegionId = regionId;
createContainerRequest.SecurityGroupId = securityGroupId
createContainerRequest.VSwitchId = vSwitchId
createContainerRequest.ContainerGroupName = "test-go-sdk"
createContainerRequest.RestartPolicy = "Never"

createContainerRequestVolume := make([]eci.CreateContainerGroupVolume, 1)
```

```

volume1 := &eci.CreateContainerGroupNFSVolume{
    Path: "/",
    Server: "0a1bf4a604-jaq68.cn-hangzhou.nas.aliyuncs.com",
}

createContainerRequestVolume[0].Name = "volume1"
createContainerRequestVolume[0].Type = "NFSVolume"
createContainerRequestVolume[0].NFSVolume = *volume1

createContainerRequest.Volume = &createContainerRequestVolume

createContainerRequestContainer := make([]eci.CreateContainerGroupContainer, 1)
createContainerRequestContainer[0].Image = "nginx"
createContainerRequestContainer[0].Name = "nginx-liu"
// option
createContainerRequestContainer[0].Cpu = requests.NewFloat(0.25)
createContainerRequestContainer[0].Memory = requests.NewFloat(0.5)
createContainerRequestContainer[0].ImagePullPolicy = "IfNotPresent"

createContainerRequest.Container = &createContainerRequestContainer

// The default number of retries of sdk-core is 3. Without the settings to ensure
// idempotence, the underlying operation does not need to automatically retry to create
// resources.
client.GetConfig().MaxRetryTime = 0
createContainerGroupResponse, err := client.CreateContainerGroup(createContainerRequest)
if err != nil {
    panic(err)
}

containerGroupId := createContainerGroupResponse.ContainerGroupId

fmt.Println(containerGroupId)

return containerGroupId
}

/**
Delete a container group.
*/
func deleteContainerGroup(containerGroupId string) {
    deleteContainerGroupRequest := eci.CreateDeleteContainerGroupRequest()
    deleteContainerGroupRequest.RegionId = regionId
    deleteContainerGroupRequest.ContainerGroupId = containerGroupId

    _, err := client.DeleteContainerGroup(deleteContainerGroupRequest)
    if err != nil {
        panic(err)
    }

    fmt.Println("DeleteContainerGroup ContainerGroupId :", containerGroupId)
}

/**
Query a container group.
*/
func describeContainerGroup(containerGroupId string) (eci.DescribeContainerGroupsContainerGroup0) {
    // Describe Container Groups
    describeContainerGroupsRequest := eci.CreateDescribeContainerGroupsRequest()
    describeContainerGroupsRequest.RegionId = regionId

```

```

containerGroupIds := append([]string{}, containerGroupId)
containerGroupIdsString, err := json.Marshal(containerGroupIds)
describeContainerGroupsRequest.ContainerGroupIds = string(containerGroupIdsString)

describeContainerGroupsResponse, err := client.DescribeContainerGroups(describeCo
ntainerGroupsRequest)
if err != nil {
    panic(err)
}

describeContainerGroupNumber := len(describeContainerGroupsResponse.ContainerG
roups)

if describeContainerGroupsResponse.TotalCount != 1 && describeContainerGro
upNumber != 1 {
    fmt.Println("Invalid ContainerGroups count", describeContainerGroupsResponse.
TotalCount, describeContainerGroupNumber)
    panic("Invalid ContainerGroups count")
}

fmt.Println("ContainerGroup status:", describeContainerGroupsResponse.ContainerG
roups[0].Status, containerGroupId,)

// container groups
return describeContainerGroupsResponse.ContainerGroups[0]
}

func describeContainerGroupMetric(containerGroupId string) {
describeContainerGroupMetricRequest :=eci.CreateDescribeContainerGroupMe
tricRequest()
describeContainerGroupMetricRequest.RegionId = regionId
describeContainerGroupMetricRequest.ContainerGroupId=containerGroupId

response, err := client.DescribeContainerGroupMetric(describeContainerGroupMetricRe
quest)
if err != nil {
    panic(err)
}

data, _ :=json.Marshal(response)

fmt.Printf("Metrics for %s:%s\n", containerGroupId, string(data))
}

func describeMultiContainerGroupMetric(containerGroupIds []string) {
describeContainerGroupMetricRequest
:=eci.CreateDescribeMultiContainerGroupMetricRequest()
describeContainerGroupMetricRequest.RegionId = regionId

data, _ :=json.Marshal(containerGroupIds)
describeContainerGroupMetricRequest.ContainerGroupIds=string(data)

response, err := client.DescribeMultiContainerGroupMetric(describeContainerGro
upMetricRequest)
if err != nil {
    panic(err)
}

data, _ = json.Marshal(response)

fmt.Printf("Metrics:%s\n", string(data))

```

```
}

func Test() {
    containerGroupIds := make(chan string)

    go func() {
        for i := 0; i < 1; i++ {
            containerGroupId := createContainerGroup()
            containerGroupIds <- containerGroupId
        }
    }()

    go func() {
        for containerGroupId := range containerGroupIds {
            for i := 0; i < 10; i++ {
                status := describeContainerGroup(containerGroupId).Status
                if Running == ContainerGroupStatus(status) {
                    break
                } else {
                    time.Sleep(5 * time.Second)
                }
            }
            //deleteContainerGroupById(containerGroupId)
        }
    }()

    // Block the program until the asynchronous operation is completed. Otherwise, the
    program exits before the operation is completed.
    var input string
    fmt.Println("waiting for input to finish:")
    fmt.Scanln(&input)
    fmt.Println("test done!")
}
```

main.go:

```
func main() {
    eci.Test()
}
```

3 Use the Java SDK

Overview

This topic describes how to install the Elastic Container Instance (ECI) Java SDK and Alibaba Cloud SDK core library.

Before you begin

Install the ECI Java SDK

You can download `aliyun-java-sdk-eci` or use the SDK through Maven coordinates. The latter method is recommended.

You can download the SDK at <https://search.maven.org/search?q=a:aliyun-java-sdk-eci>.

To use the SDK through Maven coordinates, add the following dependencies to the pom.xml file of Maven:

```
<dependencies>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <optional>true</optional>
    <version>[4.3.2,5.0.0)</version>
  </dependency>
  <dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.8.5</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-eci</artifactId>
    <version>1.0.12</version>
  </dependency>
</dependencies>
```

Sample code

Create an ECI

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.eci.model.v20180808.*;

public class CreateContainerGroup {
```



```
public static void main(String[] args) {
    DefaultProfile profile = DefaultProfile.getProfile("cn-beijing", "<accessKeyId>", "<
accessSecret>");
    IAcsClient client = new DefaultAcsClient(profile);

    CreateContainerGroupRequest request = new CreateContainerGroupRequest();
    request.setRegionId("cn-beijing");
    request.setSecurityGroupId("sg-2ze50xxve5raqetki8at");
    request.setVSwitchId("vsw-2ze462895gttskw2yc6wh");
    request.setContainerGroupName("test");

    List<CreateContainerGroupRequest.Container> containerList = new ArrayList<
CreateContainerGroupRequest.Container>();

    CreateContainerGroupRequest.Container container1 = new CreateContainerGroup
Request.Container();
    container1.setImage("nginx");
    container1.setName("nginx");
    containerList.add(container1);
    request.setContainers(containerList);

    try {
        CreateContainerGroupResponse response = client.getAcsResponse(request);
        System.out.println(new Gson().toJson(response));
    } catch (ServerException e) {
        e.printStackTrace();
    } catch (ClientException e) {
        System.out.println("ErrCode:" + e.getErrCode());
        System.out.println("ErrMsg:" + e.getErrMsg());
        System.out.println("RequestId:" + e.getRequestId());
    }
}
}
```

Response

```
{"containerGroupId":"eci-bp1h4oxf96kajk0kdf11","requestId":"8E93976E-E73B-410F-B048
-F6E8488E6D51"}
```

Delete an ECI

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.eci.model.v20180808.*;

public class DeleteContainerGroup {

    public static void main(String[] args) {
        DefaultProfile profile = DefaultProfile.getProfile("cn-beijing", "<accessKeyId>", "<
accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);

        DeleteContainerGroupRequest request = new DeleteContainerGroupRequest();
        request.setRegionId("cn-beijing");
        request.setContainerGroupId("eci-2ze4uq1ecwkm2nwf8odf");
    }
}
```

```
try {
    DeleteContainerGroupResponse response = client.getAcsResponse(request);
    System.out.println(new Gson().toJson(response));
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    System.out.println("ErrCode:" + e.getErrCode());
    System.out.println("ErrMsg:" + e.getErrMsg());
    System.out.println("RequestId:" + e.getRequestId());
}
}
```

Response

```
{"requestId":"30E08AC3-5BEE-4721-9E9C-BE0EC63AB494"}
```

4 Use the Python SDK

Before you begin

Install the ECI Python SDK

You can use Python Package Index (PyPI) to install the Python SDK from <https://pypi.org/project/aliyun-python-sdk-eci/>.

Run the following command to install the Python SDK:

```
pip install aliyun-python-sdk-eci==1.0.3
```

If a message appears indicating that the installation times out, add the default timeout period in the command as follows:

```
pip --default-timeout=100 install aliyun-python-sdk-eci==1.0.3
```

Run the following command to upgrade the Python SDK to the latest version:

```
pip install --upgrade aliyun-python-sdk-eci
```

Sample code

Create an ECI

```
#!/usr/bin/env python
#coding=utf-8

from aliyunsdkcore.client import AcsClient
from aliyunsdkeci.request.v20180808.CreateContainerGroupRequest import CreateContainerGroupRequest

client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-hangzhou')
request = CreateContainerGroupRequest()
request.set_accept_format('json')

request.set_SecurityGroupId("sg-bp13xej624v0pfqn****")
request.set_VSwitchId("vsw-bp1dmqtg608k99wmb****")
request.set_ContainerGroupName("test")
request.set_Containers([
    {
        "Image": "nginx",
        "Name": "nginx",
        "Cpu": 2,
        "Memory": 4,
        "VolumeMount": [
            "emptyVolume"
        ],
        "Commands": [
            "/bin/bash"
        ],
        "Args": [
            "-c",
```

```
        "date"
    ]
}
])
request.set_Tags([
    {
        "Key": "env",
        "Value": "test"
    }
])
request.set_Volumes([
    {
        "Name": "emptyVolume",
        "Type": "EmptyDirVolume"
    }
])

response = client.do_action_with_exception(request)
print(str(response))
```

Success response

```
{"ContainerGroupId":"eci-bp1brvmc6xtpz2ex****","RequestId":"ED7ED3FA-AA2F-47CE-B45A-122B446751D8"}
```

Delete an ECI

```
#!/usr/bin/env python
#coding=utf-8

from aliyunsdkcore.client import AcsClient
from aliyunsdkkeci.request.v20180808.DeleteContainerGroupRequest import DeleteContainerGroupRequest

client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-hangzhou')

request = DeleteContainerGroupRequest()
request.set_accept_format('json')

request.set_ContainerGroupId("eci-bp1brvmc6xtpz2ex****")

response = client.do_action_with_exception(request)
# python2: print(response)
print(str(response))
```

Success response

```
{"RequestId":"81F15B16-B5AD-4081-924A-F91D9FF8E402"}
```