

阿里云 资源编排

SDK参考

文档版本：20190917

法律声明

阿里云提醒您 在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的”现状“、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含”阿里云”、Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

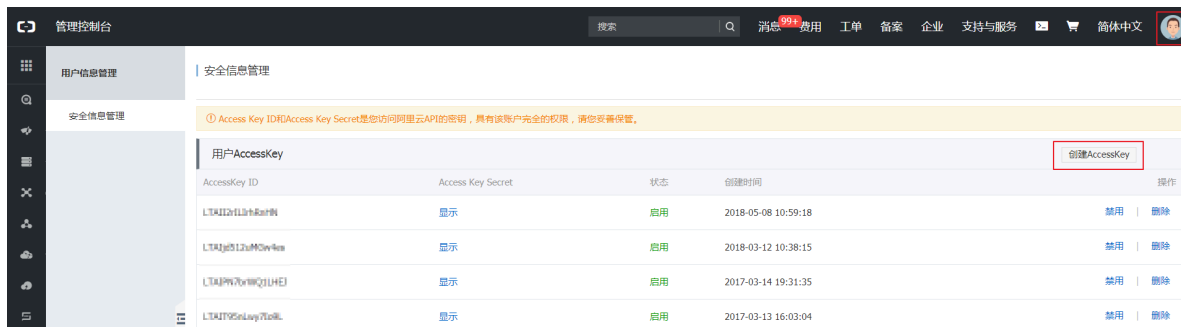
格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 禁止： 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告： 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明： 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 确定 。
<code>courier</code> 字体	命令。	执行 <code>cd /d C:/windows</code> 命令，进入Windows系统文件夹。
<code>##</code>	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
<code>[]</code> 或者 <code>[a b]</code>	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
<code>{ }</code> 或者 <code>{a b}</code>	表示必选项，至多选择一个。	<code>swich {stand slave}</code>

目录

法律声明.....	I
通用约定.....	I
1 创建 AccessKey.....	1
2 SDK下载和使用说明.....	2
3 Python SDK使用示例（新）	3
4 Python SDK使用示例（旧）	8
5 Java SDK使用示例.....	14

1 创建 AccessKey

1. 登录[阿里云控制台](#)。
2. 在控制台页面，将鼠标光标移动至您的用户图标上，在下拉框中，单击accesskeys，进入安全管理页面。
3. 单击创建 AccessKey 按钮，即可进入 AccessKey 的创建流程。



2 SDK下载和使用说明

您可以通过访问以下链接，查看阿里云SDK的下载地址和使用说明。

下载地址：

[阿里云SDK下载地址](#)

使用说明：

- [Python SDK使用说明](#)
- [Java SDK通用说明](#)
- [PHP SDK使用说明](#)
- [.NET SDK使用说明](#)
- [GO SDK使用说明](#)
- [Node.js SDK使用说明](#)

3 Python SDK使用示例 (新)

本文为您介绍如何使用资源编排服务（ROS）的Python SDK来创建和管理资源栈。

背景信息

您除了可以在ROS控制台创建资源栈，还可以使用API代码来创建和管理资源栈。

准备工作

1. 下载及安装Python SDK。

a) 使用pip安装aliyun-python-sdk-core。

```
pip install aliyun-python-sdk-core
```



说明:

*aliyun-python-sdk-core*是所有阿里云官方Python SDK的公共组件。如果安装过程提示权限错误，可能是因为没有Python安装路径的写权限。此命令也可以改为`sudo pip install aliyun-python-sdk-core`。

b) 安装ROS SDK。

```
pip install aliyun-python-sdk-ros
```

2. 初始化SDK。

a) 导入相关的包。

```
from aliyunsdkcore.client import AcsClient
from aliyunsdkros.request.v20190910.CreateStackRequest import
CreateStackRequest
from aliyunsdkros.request.v20190910.GetStackRequest import
GetStackRequest
from aliyunsdkros.request.v20190910.DeleteStackRequest import
DeleteStackRequest
from aliyunsdkros.request.v20190910.DescribeRegionsRequest import
DescribeRegionsRequest
from aliyunsdkros.request.v20190910.ListStacksRequest import
ListStacksRequest
```

b) 初始化SDK客户端对象。

```
AK = '<yourAccessKeyId>'
SECRET = '<yourAccessKeySecret>'
Region = '<yourRegionId>'
client = AcsClient(AK, SECRET, Region)
```



注意:

- 示例中的命令行都使用Linux的shell, 如果您是Windows/DOS用户, 则需要根据情况修改。
- ROS Python SDK依赖Python 2.7以上版本。
- ROS Python SDK 3.0.0及以上版本, 支持v20150901版和v20190910版API。

查询可用地域列表

您可以使用Python SDK查询可用地域列表。

```
def describe_region():
    """describe regions list """
    request = DescribeRegionsRequest()
    request.set_accept_format('json')
    response = client.do_action_with_exception(request)

    return response.decode('utf8')
```

创建资源栈

创建资源栈时, 您必须指定以下参数:

- StackName: 将要创建的资源栈的名称。每个用户空间下的资源栈名称不能重复。
- TimeoutInMinutes: 创建过程如果在指定的时间后不能完成则超时失败。单位为分钟。
- TemplateBody: 创建的资源栈使用的模板内容。
- TemplateURL: 模板主体的文件的位置。必须指定TemplateBody或TemplateURL, 但不能同时指定两者。
- Parameters: 创建的资源栈所需要的参数。需要在模板中定义key。

```
stack_name = "MyStack"
timeout = 10
template_body = """
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "VpcName": {
      "Type": "String",
      "Description": "Vpc Name",
      "Label": "Vpc Name"
    },
    "CidrBlock": {
      "Type": "String",
      "Description": "Vpc CidrBlock",
      "Label": "Vpc CidrBlock"
    }
  },
  "Resources": {
    "Vpc": {
      "Type": "ALIYUN::ECS::VPC",
      "Properties": {
        "CidrBlock": {
          "Ref": "CidrBlock"
        },
        "VpcName": {
          "Ref": "VpcName"
        }
      }
    }
  }
}
```



```

    }
    }
    }
    """
    params = [
        {
            "ParameterValue": "192.168.0.0/16",
            "ParameterKey": "CidrBlock"
        },
        {
            "ParameterValue": "TestVpc",
            "ParameterKey": "VpcName"
        }
    ]

def create_stack():
    """create stack"""
    request = CreateStackRequest()
    request.set_accept_format('json')
    request.set_StackName(stack_name)
    request.set_TimeoutInMinutes(timeout)
    request.set_TemplateBody(template_body)
    request.set_Parametersss(params)
    response = client.do_action_with_exception(request)

    return response.decode('utf8')

```

查询资源栈

您需要输入对应资源栈的ID。

```

def get_stack():
    """get descriptions of the stack"""
    request = GetStackRequest()
    request.set_accept_format('json')
    request.set_StackId(stack_id)
    response = client.do_action_with_exception(request)

    return response.decode('utf8')

```

删除资源栈

您需要输入对应资源栈的ID。

```

def delete_stack():
    """delete stack"""
    request = DeleteStackRequest()
    request.set_accept_format('json')
    request.set_StackId(stack_id)
    response = client.do_action_with_exception(request)

    return response.decode('utf8')

```

```

import json
from time import sleep
from aliyunsdkcore.client import AcsClient

```

```
from aliyunsdkros.request.v20190910.CreateStackRequest
import CreateStackRequest
from aliyunsdkros.request.v20190910.GetStackRequest import
GetStackRequest
from aliyunsdkros.request.v20190910.DeleteStackRequest
import DeleteStackRequest
from aliyunsdkros.request.v20190910.DescribeRegionsRequest
import DescribeRegionsRequest
from aliyunsdkros.request.v20190910.ListStacksRequest import
ListStacksRequest

AK = '<yourAccessKeyId>'
SECRET = '<yourAccessKeySecret>'
Region = '<yourRegionId>' # 例如: 'cn-beijing', 'cn-hangzhou'

client = AcsClient(AK, SECRET, Region)

def describe_region():
    """describe regions list """
    request = DescribeRegionsRequest()
    request.set_accept_format('json')
    response = client.do_action_with_exception(request)

    return response.decode('utf8')

def create_stack(stack_name, timeout, template_body, params
=[]):
    """create stack"""
    request = CreateStackRequest()
    request.set_accept_format('json')
    request.set_StackName(stack_name)
    request.set_TimeoutInMinutes(timeout)
    request.set_TemplateBody(template_body)
    response = client.do_action_with_exception(request)

    return response.decode('utf8')

def get_stack(stack_id):
    """get descriptions of the stack"""
    request = GetStackRequest()
    request.set_accept_format('json')
    request.set_StackId(stack_id)
    response = client.do_action_with_exception(request)

    return response.decode('utf8')

def delete_stack(stack_id):
    """delete stack"""
    request = DeleteStackRequest()
    request.set_accept_format('json')
    request.set_StackId(stack_id)
    response = client.do_action_with_exception(request)

    return response.decode('utf8')

if __name__ == '__main__':
    test_template = """
    {
```

```
    "ROSTemplateFormatVersion": "2015-09-01",
    "Parameters": {
      "VpcName": {
        "Type": "String",
        "Description": "Vpc Name",
        "Label": "Vpc Name"
      },
      "CidrBlock": {
        "Type": "String",
        "Description": "Vpc CidrBlock",
        "Label": "Vpc CidrBlock"
      }
    },
    "Resources": {
      "Vpc": {
        "Type": "ALIYUN::ECS::VPC",
        "Properties": {
          "CidrBlock": {
            "Ref": "CidrBlock"
          },
          "VpcName": {
            "Ref": "VpcName"
          }
        }
      }
    }
  }
}

"""

parameters = [
    {"ParameterKey": "CidrBlock", "ParameterValue": "192
.168.0.0/16"},
    {"ParameterKey": "VpcName", "ParameterValue": "
TestVpc"}
]

describe_region()
stack = create_stack('MyStack', 10, test_template,
parameters)
get_stack(json.loads(stack)["StackId"])
sleep(3) # 等待创建资源栈任务执行完毕
delete_stack(json.loads(stack)["StackId"])
```

4 Python SDK使用示例 (旧)

本文为您介绍如何使用资源编排服务（ROS）的Python SDK来创建和管理资源栈。

背景信息

您除了可以在ROS控制台创建资源栈，还可以使用API代码来创建和管理资源栈。

准备工作

1. 下载及安装Python SDK。

a) 使用pip安装aliyun-python-sdk-core。

```
pip install aliyun-python-sdk-core
```



说明:

aliyun-python-sdk-core是所有阿里云官方Python SDK的公共组件。如果安装过程提示权限错误，可能是因为没有Python安装路径的写权限。此命令也可以改为sudo pip install aliyun-python-sdk-core。

b) 安装ROS SDK。

```
pip install aliyun-python-sdk-ros
```

2. 初始化SDK。

a) 导入相关的包。

```
import json
from aliyunsdkcore import client
from aliyunsdkros.request.v20150901.CreateStacksRequest import
CreateStacksRequest
from aliyunsdkros.request.v20150901.DescribeStackDetailRequest
import DescribeStackDetailRequest
from aliyunsdkros.request.v20150901.DeleteStackRequest import
DeleteStackRequest
from aliyunsdkros.request.v20150901.DescribeRegionsRequest import
DescribeRegionsRequest
```

b) 初始化SDK客户端对象。

```
AK = '<yourAccessKeyId>'
SECRET = '<yourAccessKeySecret>'
Region = '<yourRegionId>'
clt = client.AcsClient(AK, SECRET, Region)
```



注意:

- 示例中的命令行都使用Linux的shell，如果您是Windows/DOS用户，则需要根据情况修改。

- ROS Python SDK依赖Python 2.7以上版本。
- ROS Python SDK 3.0.0及以上版本，支持v20150901版和v20190910版API。

查询可用地域列表

您可以使用Python SDK查询可用地域列表。

```
def describe_region():
    """describe regions list"""
    req = DescribeRegionsRequest() # 构造请求对象
    status, headers, body = clt.get_response(req)
    if status == 200:
        regions = json.loads(body)
        return regions
    else:
        return 'Unexpected errors: status=%d, error=%s' % (status,
body)
```

创建资源栈

创建资源栈时，您必须指定以下参数：

- Name：将要创建的资源栈的名称。每个用户空间下的资源栈名称不能重复。
- TimeoutMins：创建过程如果在指定的时间后不能完成则超时失败。单位为分钟。
- Template：创建的资源栈使用的模板内容。
- Parameters：创建的资源栈所需要的参数。需要在模板中定义key。
- status：请求的返回状态，与HTTP状态码对应。一般2xx表示成功，4xx和5xx表示出错。
- headers：请求的返回头信息，与HTTP头对应。
- body：请求的返回内容，与HTTP请求返回的body对应。

```
test_template = '''
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "VpcName": {
      "Type": "String",
      "Description": "Vpc Name",
      "Label": "Vpc Name"
    },
    "CidrBlock": {
      "Type": "String",
      "Description": "Vpc CidrBlock",
      "Label": "Vpc CidrBlock"
    }
  },
  "Resources": {
    "Vpc": {
      "Type": "ALIYUN::ECS::VPC",
      "Properties": {
        "CidrBlock": {
          "Ref": "CidrBlock"
        },
        "VpcName": {
```

```

        "Ref": "VpcName"
    }
}
}
}
'''

```

```

def create_stack():
    """create stack"""
    global result
    req = CreateStacksRequest()
    create_stack_body = dict() # 构造请求的消息体内容
    test_parameters = {
        "CidrBlock": "192.168.0.0/16",
        "VpcName": "test_vpc"
    }
    create_stack_body["Name"] = 'template-test-vpc'
    create_stack_body["Template"] = test_template
    create_stack_body["Parameters"] = test_parameters
    create_stack_body["TimeoutMins"] = 60
    req.set_content(json.dumps(create_stack_body))
    req.set_content_type('application/json')
    status, headers, body = clt.get_response(req)
    if status == 201: # 成功时返回状态为201
        result = json.loads(body)
        return result
    else:
        return 'Unexpected errors: status=%d, error=%s' % (status,
body)

```

当创建资源栈的请求成功时，返回的body中包含被创建资源栈的ID、Name。

```

{'Id': '2ffcf5e5d-d35c-4f35-858d-dda78922b7c6', 'Name': 'template-test-vpc'}

```

创建资源栈的请求会同步返回，但资源栈内的资源创建是由资源编排服务在后台异步执行的，创建请求返回并不表示所有资源已经创建完成。您可以通过ROS的Web控制台或者API来查询资源栈的创建状态、创建过程中的事件等。

查询资源栈

您需要输入对应资源栈的ID、Name。

```

def describe_stack():
    """describe stack"""
    req = DescribeStackDetailRequest()
    req.set_StackName(result['Name'])
    req.set_StackId(result['Id'])
    status, headers, body = clt.get_response(req)
    if status == 200: # 成功时返回状态为200
        res = json.loads(body)
        if res['Status'] != 'CREATE_IN_PROGRESS': # 资源栈的状态不是创建
中时
            return res
        else:
            return describe_stack()
    else:

```

```
return 'Unexpected errors: status=%d, error=%s' % (status,
body)
```

删除资源栈

您需要输入对应资源栈的ID、Name。

```
def delete_stack():
    """delete stack"""
    req = DeleteStackRequest()
    req.set_StackName(result['Name'])
    req.set_StackId(result['Id'])
    status, headers, body = clt.get_response(req)
    if status == 204: # 成功时返回状态为204
        return body
    else:
        return 'Unexpected errors: status=%d, error=%s' % (status,
body)
```

```
import json
from aliyunsdkcore import client
from aliyunsdkros.request.v20150901.CreateStacksRequest
import CreateStacksRequest
from aliyunsdkros.request.v20150901.DescribeStackDetailR
equest import DescribeStackDetailRequest
from aliyunsdkros.request.v20150901.DeleteStackRequest
import DeleteStackRequest
from aliyunsdkros.request.v20150901.DescribeRegionsRequest
import DescribeRegionsRequest

AK = '<yourAccessKeyId>'
SECRET = '<yourAccessKeySecret>'
Region = '<yourRegionId>' # 例如: 'cn-beijing', 'cn-hangzhou'

clt = client.AcsClient(AK, SECRET, Region)

test_template = '''
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "VpcName": {
      "Type": "String",
      "Description": "Vpc Name",
      "Label": "Vpc Name"
    },
    "CidrBlock": {
      "Type": "String",
      "Description": "Vpc CidrBlock",
      "Label": "Vpc CidrBlock"
    }
  },
  "Resources": {
    "Vpc": {
      "Type": "ALIYUN::ECS::VPC",
      "Properties": {
        "CidrBlock": {
          "Ref": "CidrBlock"
        },
        "VpcName": {
```

```
        "Ref": "VpcName"
    }
}
}
}
}
'''

def describe_region():
    """describe regions list """
    req = DescribeRegionsRequest()
    status, headers, body = clt.get_response(req)
    if status == 200:
        regions = json.loads(body)
        return regions
    else:
        return 'Unexpected errors: status=%d, error=%s' % (
status, body)

def create_stack():
    """create stack"""
    global result
    req = CreateStacksRequest()
    create_stack_body = dict()
    test_parameters = {
        "CidrBlock": "192.168.0.0/16",
        "VpcName": "test_vpc"
    }
    create_stack_body["Name"] = 'template-test-vpc'
    create_stack_body["Template"] = test_template
    create_stack_body["Parameters"] = test_parameters
    create_stack_body["TimeoutMins"] = 60
    req.set_content(json.dumps(create_stack_body))
    req.set_content_type('application/json')
    status, headers, body = clt.get_response(req)
    if status == 201:
        result = json.loads(body)
        return result
    else:
        return 'Unexpected errors: status=%d, error=%s' % (
status, body)

def describe_stack():
    """describe stack"""
    req = DescribeStackDetailRequest()
    req.set_StackName(result['Name'])
    req.set_StackId(result['Id'])
    status, headers, body = clt.get_response(req)
    if status == 200:
        res = json.loads(body)
        if res['Status'] != 'CREATE_IN_PROGRESS':
            return res
        else:
            return describe_stack()
    else:
        return 'Unexpected errors: status=%d, error=%s' % (
status, body)

def delete_stack():
    """delete stack"""
```



```
req = DeleteStackRequest()
req.set_StackName(result['Name'])
req.set_StackId(result['Id'])
status, headers, body = clt.get_response(req)
if status == 204:
    return body
else:
    return 'Unexpected errors: status=%d, error=%s' % (
status, body)

if __name__ == '__main__':
    describe_region()
    create_stack()
    describe_stack()
    delete_stack()
```

5 Java SDK使用示例

本文为您介绍如何使用资源编排服务（ROS）的Java SDK来创建和管理资源栈。

背景信息

您除了可以在ROS控制台创建资源栈，还可以使用API代码来创建和管理资源栈。

准备工作

1. 下载及安装Java SDK。



注意:

建议您使用JRE 1.8以上版本。

2. 在pom.xml中，添加aliyun-java-sdk-core及其它依赖包。

```
<dependencies>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <version>4.4.1</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-ros</artifactId>
    <version>2.2.8</version>
  </dependency>
  <dependency>
    <groupId>org.json</groupId>
    <artifactId>json</artifactId>
    <version>20180813</version>
  </dependency>
  <dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
    <version>4.5.5</version>
  </dependency>
  <dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.8.5</version>
  </dependency>
</dependencies>
```

3. 初始化SDK。

a) 导入必要的库。

```
import java.util.HashMap;
import java.util.Map;
import org.json.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.http.FormatType;
```

```
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.ros.model.v20150901.CreateStacksRequest;
import com.aliyuncs.ros.model.v20150901.DeleteStackRequest;
import com.aliyuncs.ros.model.v20150901.DescribeRegionsRequest;
import com.aliyuncs.ros.model.v20150901.DescribeStacksRequest;
```

b) 初始化SDK客户端对象。

```
private static String ACCESSKEYID = "<yourAccessKeyId>";
private static String SECRET = "<yourAccessKeySecret>";
private static String REGIONID = "<yourRegionId>";

private static IAcsClient initClient() {
    DefaultProfile profile = DefaultProfile.getProfile(REGIONID,
ACCESSKEYID, SECRET);
    return new DefaultAcsClient(profile);
}
```

查询可用地域列表

您可以使用Java SDK查询可用地域列表。

```
public HttpResponse describeRegion() throws ServerException,
ClientException {
    DescribeRegionsRequest request = new DescribeRegionsRequest();
    return initClient().doAction(request);
}
```

创建资源栈

创建资源栈时，您必须指定以下参数：

- Name：将要创建的资源栈的名称。每个用户空间下的资源栈名称不能重复。
- TimeoutMins：创建过程如果在指定的时间后不能完成则超时失败。单位为分钟。
- Template：创建的资源栈使用的模板内容。

```
public HttpResponse createStack(String stackname) throws ServerException, ClientException {
    CreateStacksRequest request = new CreateStacksRequest();
    Map<String, Object> map = new HashMap<String, Object>();
    String template = "{\n\"ROSTemplateFormatVersion\": \"2015-09-01\n\"}";
    // 去掉模版中的"\n"
    JSONObject jsonObject = new JSONObject(template);
    map.put("Name", "empty-template-test");
    map.put("Template", jsonObject.toString());
    map.put("TimeoutMins", 60);
    // 转byte
    JSONObject jsonTemplate = new JSONObject(map);
    String string = jsonTemplate.toString();
    byte[] b = string.getBytes();
    request.putHeaderParameter("x-acs-region-id", "cn-hangzhou");
    request.setHttpContent(b, "utf-8", FormatType.JSON);
    return initClient().doAction(request);
}
```

```
}
```

当创建资源栈的请求成功时，返回的body中包含被创建资源栈的ID、Name。

```
{"Id": "1cab9199-2fc5-4608-9672-014811d073a9", "Name": "empty-template-test"}
```

创建资源栈的请求会同步返回，但资源栈内的资源创建是由资源编排服务在后台异步执行的，创建请求返回并不表示所有资源已经创建完成。您可以通过ROS的Web控制台或者API来查询资源栈的创建状态、创建过程中的事件等等。

查询资源栈

您需要输入对应资源栈的ID、Name。

```
public String describeStack(String stackname, String stackid) throws
ServerException, ClientException {
    DescribeStackDetailRequest request = new DescribeStackDetailR
equest();
    request.setStackId(stackid);
    request.setStackName(stackname);
    return initClient().doAction(request).getHttpContentString();
}
```

删除资源栈

您需要输入对应资源栈的ID、Name。

```
public String deleteStack(String stackname, String stackid) throws
ServerException, ClientException {
    DeleteStackRequest request = new DeleteStackRequest();
    request.putHeaderParameter("x-acs-region-id", "cn-hangzhou");
    request.setStackId(stackid);
    request.setStackName(stackname);
    return initClient().doAction(request).getHttpContentString();
}
```

```
package com;

import java.util.HashMap;
import java.util.Map;
import org.json.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.ros.model.v20150901.CreateStacksRequest;
import com.aliyuncs.ros.model.v20150901.DeleteStackRequest;
import com.aliyuncs.ros.model.v20150901.DescribeRegionsReque
st;
import com.aliyuncs.ros.model.v20150901.DescribeStacksReques
t;

public class Stack {
```

```
private static String ACCESSKEYID = "<yourAccessKeyId>";
private static String SECRET = "<yourAccessKeySecret>";
private static String REGIONID = "<yourRegionId>";

private static IAcsClient initClient() {
    DefaultProfile profile = DefaultProfile.getProfile(
REGIONID, ACCESSKEYID, SECRET);
    return new DefaultAcsClient(profile);
}

public String createStack(String stackname) throws
ServerException, ClientException {
    CreateStacksRequest request = new CreateStacksRequest();
    Map<String, Object> map = new HashMap<String, Object>
>();
    String template = "{\"ROSTemplateFormatVersion\": \"
2015-09-01\"}";
    // 去掉模版中的"\\"
    JSONObject jsonObject = new JSONObject(template);
    map.put("Name", stackname);
    map.put("Template", jsonObject.toString());
    map.put("TimeoutMins", 60);
    // 转byte
    JSONObject jsonTemplate = new JSONObject(map);
    String string = jsonTemplate.toString();
    byte[] b = string.getBytes();
    request.putHeaderParameter("x-acs-region-id", "cn-
hangzhou");
    request.setHttpContent(b, "utf-8", FormatType.JSON);
    return initClient().doAction(request).getHttpCon
tentString();
}

public String describeRegion() throws ServerException,
ClientException {
    DescribeRegionsRequest request = new DescribeRe
gionsRequest();
    return initClient().doAction(request).getHttpCon
tentString();
}

public String describeStack(String stackname, String
stackid) throws ServerException, ClientException {
    DescribeStackDetailRequest request = new DescribeSt
ackDetailRequest();
    request.setStackId(stackid);
    request.setStackName(stackname);
    return initClient().doAction(request).getHttpCon
tentString();
}

public String deleteStack(String stackname, String
stackid) throws ServerException, ClientException {
    DeleteStackRequest request = new DeleteStackRequest
();
    request.putHeaderParameter("x-acs-region-id", "cn-
hangzhou");
    request.setStackId(stackid);
    request.setStackName(stackname);
    return initClient().doAction(request).getHttpCon
tentString();
}
```

```
    public static void main(String[] args) throws ServerException, ClientException {
        System.out.println(new Stack().createStack("test_stack1112"));
    }
}
```