

CDN

开发指南

开发指南

SDK 手册

CLI 手册

CDN命令介绍

aliyuncli 安装

阿里云命令行工具(aliyuncli)是基于阿里云开放API建立的手边管理工具。借助这个工具，您可以轻松的调用阿里云开放API，管理您的阿里云产品。阿里云命令行工具与阿里云open API——对应，灵活性高且易于扩展。您可以基于阿里云命令行工具对阿里云原子的API进行封装，扩展出自己想要的功能。

pip 安装阿里云命令行工具会自动帮助您处理好各种依赖。

您可以直接采用下面的命令安装aliyuncli:

```
$sudo pip install aliyuncli
```

aliyuncli 升级:

```
$sudo pip install --upgrade aliyuncli
```

更多产品细节, 请参考：[CLI产品文档](#)

CDN SDK安装

使用CDN的功能，需要与CDN SDK配合使用，因此需要您安装CDN SDK。

pip 安装SDK会自动帮助您处理好各种依赖还有升级, 因此强烈建议您采用pip方式安装SDK.

您可以直接采用下面的命令安装CDN SDK:

```
$ sudo pip install aliyun-python-sdk-cdn
```

SDK升级：

```
$ sudo pip install --upgrade aliyun-python-sdk-cdn
```

aliyuncli cdn 命令说明

下面文档将通过命令说明和使用示范进行cdn 命令的展示

服务操作命令

OpenCdnService

命令说明：

开通CDN服务，只有开通后，才能进行域名操作，一个用户只能开通一次。开通条件：帐号已通过实名认证。

使用示范：

```
aliyuncli cdn OpenCdnService --InternetChargeType PayByTraffic
```

DescribeCdnService

命令说明：

查询CDN服务状态。包括当前计费类型，服务开通时间，下次生效的计费类型，当前业务状态等。

- 使用示范：

```
aliyuncli cdn DescribeCdnService
```

ModifyCdnService

命令说明：

变更CDN服务的计费类型。

使用示范：

```
aliyuncli cdn ModifyCdnService --InternetChargeType PayByTraffic
```

更多 API 接口详细信息，请进入[API 手册](#)了解详情

周边工具

鉴权代码示例

概述

URL鉴权规则请查阅 [URL鉴权文档](#)，通过这个 demo 您可以根据业务需要，方便的对URL进行鉴权处理。以下 Python Demo包含三种鉴权方式：A鉴权方式、B鉴权方式、C鉴权方式，分别描述了三种不同鉴权方式的请求URL构成、哈希字符串构成等内容

Python版本

```
import re
import time
import hashlib
import datetime

def md5sum(src):
    m = hashlib.md5()
    m.update(src)
```

```

return m.hexdigest()

def a_auth(uri, key, exp):
    p = re.compile("^(http://|https://)?([^/?]+)(/[^?]*)?(\\?.*)?$")
    if not p:
        return None
    m = p.match(uri)
    scheme, host, path, args = m.groups()
    if not scheme: scheme = "http://"
    if not path: path = "/"
    if not args: args = ""
    rand = "0" # "0" by default, other value is ok
    uid = "0" # "0" by default, other value is ok

    sstring = "%s-%s-%s-%s-%s" %(path, exp, rand, uid, key)
    hashvalue = md5sum(sstring)
    auth_key = "%s-%s-%s-%s-%s" %(exp, rand, uid, hashvalue)

    if args:
        return "%s%s%s%s&auth_key=%s" %(scheme, host, path, args, auth_key)
    else:
        return "%s%s%s%s?auth_key=%s" %(scheme, host, path, args, auth_key)

def b_auth(uri, key, exp):
    p = re.compile("^(http://|https://)?([^/?]+)(/[^?]*)?(\\?.*)?$")
    if not p:
        return None
    m = p.match(uri)
    scheme, host, path, args = m.groups()
    if not scheme: scheme = "http://"
    if not path: path = "/"
    if not args: args = ""

    # convert unix timestamp to "YYmmDDHHMM" format
    nexp = datetime.datetime.fromtimestamp(exp).strftime('%Y%m%d%H%M')
    sstring = key + nexp + path
    hashvalue = md5sum(sstring)

    return "%s%s/%s/%s%s%s" %(scheme, host, nexp, hashvalue, path, args)

def c_auth(uri, key, exp):
    p = re.compile("^(http://|https://)?([^/?]+)(/[^?]*)?(\\?.*)?$")
    if not p:
        return None
    m = p.match(uri)
    scheme, host, path, args = m.groups()
    if not scheme: scheme = "http://"
    if not path: path = "/"
    if not args: args = ""

    hexexp = "%x" %exp
    sstring = key + path + hexexp
    hashvalue = md5sum(sstring)

    return "%s%s/%s/%s%s%s" %(scheme, host, hashvalue, hexexp, path, args)

```

```
def main():

    uri = "http://xc.cdnpe.com/ping?foo=bar" # original uri
    key = "<input private key>" # private key of authorization
    exp = int(time.time()) + 1 * 3600 # expiration time: 1 hour after current itme

    authuri = a_auth(uri, key, exp) # auth type: a_auth / b_auth / c_auth

    print("URL : %s\nAUTH: %s" %(uri, authuri))

if __name__ == "__main__":
    main()
```