Aliyun SMS SDK Documentation

Release 1.1.0

Hou, Lu

Mar 05, 2018

Contents

CHAPTER 1

Installation

The *alisms* package is available on *pypi*, so use:

```
pip install alisms
```

to install it.

CHAPTER 2

AliyunSMS Object

AliyunSMS Object has the following attributes and methods:

```
class aliyun_sms.AliyunSMS(config_file='', access_key_id='', access_key_secret='',␣
↪region_id='', host='http://dysmsapi.aliyuncs.com')
```

Class

- *config_file*: Configure file name.
- *access_key_id*: *accessKeyId* of Aliyun *sub-account* (*subaccount* is recommended for security).
- *access_key_secret*: *accessKeySecret* of the id (vital).
- *region_id*: Region name of your SMS service.
- *host*: API host, default is enough.

% A new instance of AliyunSMS class.

CHAPTER 3

AliyunSMS Public Attributes

```
AliyunSMS.sms_params
```

This is the *dict* of parameters for SMS request. It can be get and set directly (A *dict* is mandatory)

CHAPTER 4

AliyunSMS Methods

```
AliyunSMS.generate_signature(params=None, method='GET', url='/')
```

This function can generate signature based on the *params*, *method* and *url*. Of course *access_key_secret* is necessary!

- *params*: A *dict* parameters for the request, *OrderedDict* is better since the sequence is of importance.
- *method*: HTTP method for the request, default 'GET'.
- *url*: Url endpoint of the request, default is '/' if using *send_sms()*.

% The signature string

```
AliyunSMS.send_sms(phone_numbers, sign_name, template_code, template_params=None,␣
↪raw=True, **kwargs)
```

This function is used to send SMS via Aliyun API.

- *phone_numbers*: The list of phone numbers, can be a string if only one number
- *sign_name*: Sign name configured in Aliyun console
- *template_code*: The template code defined in Aliyun console
- *template_params*: The params that need to be used in template
- *raw*: If to return the original *requests* instance

% Status: success or failure

```
AliyunSMS.query_details(phone_number, serial_number='', send_date='', page_size='10',␣
↪current_page='1', raw=True, **kwargs)
```

This function is used to query sending histories specified by *phone_number* and *send_date*.

- *phone_number1*: Only one phone number.
- *serial_number*: Serial number of a SMS message, can be received from return of *send_sms*.
- *send_date*: Search date, less than 30 days, form: *20170801*.

- *page_size*: Paging, max 50 items a page.
- *current_page*: Current page.
- *raw*: If to return the original *requests* instance.

% Details of the response

CHAPTER 5

Useful Functions

```
utils.hmac64(object_str, secret, alg='sha1')
```

Compute the *HMAC-{alg}* of the *object_str* with *secret* and get the return after *base64* encoding

- *object_str*: Original string that needs to be encrypted.
- *secret*: secret string.
- *alg*: HMAC algorithm, default *sha1*.

% The encrypted string

```
utils.parse_config(config_file, part='_all')
```

Parse config file, a *YAML* file is mandatory

- *config_file*: Configure file name.
- *part*: Return part of the configures, default is *_all*.

% The configures in *dict*