zentera

Zentera Systems, Inc.

# *Zapi.py*

Zentera Cloud Over IP® API Helper Functions
Quick Start Guide

Version 3.5.1 v1

## Table of Contents

## Zentera Application Programming Interface Helper Functions

This guide outlines the use of Zentera demonstration software showing the usefulness of the Zentera API for managing multi-cloud network and security environments.

The native API calls are available from any controller appliance running version 3.4.2 or greater. The API is reached through web-based Representational state transfer (REST)[1] calls using either the POST or GET methods and passing a well-formed JavaScript Object Notation (JSON)[2]

### Web API Security

In order to maintain security, the API REST calls include a keyed-hash message authentication code (HMAC) method that combines keys for accessing the API. The HMAC key must be computed for each REST interface call and essentially combines the user's API key with the contents of the message for both data integrity and message authentication[3].

### Purpose of the *Zapi.py* Helper Functions

Creating one or two native Zentera API calls is a relatively straightforward process. However, as the number of API calls grows, it is helpful to encapsulate some of that complexity into functions that simplify the process. This also leads to more readable code.

Written in Python, a primary design goal of the *Zapi.py* was to reduce any given API call into a single line of user code.

*Zapi.py* encapsulates the process of building API calls by:

1) Assembling the appropriate argument data structures in preparation for JSON conversion
2) Providing the API command and parameters to the payload construction process
3) Calculate the HMAC key from the payload string
4) Assembling the overall request including the payload, API and HMAC keys
5) POST or GET the request to the web API
6) Recover and optionally decode the results

[1] "Representational state transfer," *Wikipedia, The Free Encyclopedia.*
[2] "JSON," Wikipedia, The Free Encyclopedia,
[3] "Hash-based message authentication code," *Wikipedia, The Free Encyclopedia*.

## Overall Structure of a Python API Script using *Zapi.py*

The overall structure of a script that uses the *Zapi.py* is as follows:

```
#!/usr/bin/python

import Zapy
import sys
import os

# Other Imports

# Globals
apiURL = 'https://<ControllerURL>/zCenter/webapi/doApi'

# User Code
print 'Creating Application Profile "APITesting"'
apId = Zapi.AppProfile_create(apiURL, 'APITesting', 'dynamic')
print '   apId =', apId
```

This example creates an Application Profile (and auto-generated Cloud Project) in the Customer's zCenter environment and returns a response with status and other information.

Running the script produces this output in the terminal window:

```
Creating Application Profile "APITesting"
   apId = 1e22697e70944811baa405556a6e0aa9
```

## Under the Hood

A native call to the zCenter API command '*$AppProfile.create'* requires the JSON request:

```
{
      "request":{
            "cmd": "$AppProfile.create",
            "args": {
                  "coipAssignment": "dynamic",
                  "appProfileName":"APITesting"
            }
      },
      "kid":"<KeyID>",
      "hmac":"<HMAC>"
}
```

The API returns the following in response[4]:

```
{
        u'status': u'Ok',
        u'message': u'OK',
        u'data': {
                u'appProfileId': u'1e22697e70944811baa405556a6e0aa9'
        }
}
```

The *Zapi.py* call packages the command and extracts out the App Profile ID string using the following Python code:

```
return(info['data']['appProfileId'])
```

Taking this approach simplifies the use of the native API.

Error situations are handled by the *Zapi.py* function, *makeZapiRequest()*.

## *Zapi.py* API Key Functions

Because the web API requires the use of API keys (generated by zCenter), it is not a good idea to code them directly into a script. Directly coding keys into a script is a potential security breach as well as making a given API script useful only with a single customer account.

The *Zapi.py* module relies on pulling these keys directly out of the user's runtime shell environment variables:

```
ZENTERA_API_SKEY        : Customer's API Secret Key
ZENTERA_KID_SKEY        : Customer's API Public Key
```

These two values can be set in the user's environment (in this case Bash):

```
export ZENTERA_API_SKEY='<SecretKey>'
export ZENTERA_KID_SKEY='<PublicKey>'
```

Debugging *Zapi.py* Scripts
After importing the *Zapi.py* module, a developer can at anytime in the script set or unset the building debug code.

The module's hidden variable, `_ZapiDEBUG` is set to False by default but issuing the script line:

```
_ZapiDEBUG = True
```

[4] Note that the return strings carry an initial '*u*' on the front of the string. Thus the string `u'This is a Unicode String'` indicates to Python that this string is formatted in Unicode.

Turns on the debugging output. Likewise, setting it to False turns it off. Running the example script from above with debugging on looks like this:

```
Creating Application Profile "APITesting"

>>> Zapi Request  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
{"request":{"cmd": "$AppProfile.create", "args":
{"coipAssignment": "dynamic", "appProfileName":
"APITesting"}},"kid":"<KeyID>","hmac":"<HMAC>"}
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>


<<< Zapi Response <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
{u'status': u'Ok', u'message': u'OK', u'data': {u'appProfileId':
u'1e22697e70944811baa405556a6e0aa9'}}
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

   apId = 1e22697e70944811baa405556a6e0aa9
```

## Zapi Python Library Import

*Zapi.py* uses a number of standard, but possibly not installed, Python library packages. If you receive a warning message during a *Zapi.py* based script run of the form:

```
File <Path>/Zapi.py", line 32, in <module>
     import httplib
```

Please refer to the official Python Packaging User's Guide for more information on how to install the missing packages.

The full list of required packages can be found in the first few lines of the *Zapi.py* file.