CONTENTS

CHANGE LOG

Version	Modifier	Main changes
1	Amit Nevo	Created document
2	Amit Nevo	Added code samples for PHP, .net and Node.js JavaScript
3	Amit Nevo	Removed all references to existing services end-points

SERVICES AUTHENTICATION

Tranzila uses a secure access-token to ensure authentication prevent processing of the same request more than once and preventing man-in-the-middle attacks.

In order to achieve that, every merchant must enroll to Tranzila API Services and get both public and secret keys from Tranzila

While public key is used in each request and is exposed in the request header, secret key is only used internally by both merchant application and server application for each merchant.

Please make sure secret key is kept safe and not shared with anyone at all time. This restriction means you cannot call Tranzila API from within web client application, but rather via a proxy service on your server.

Authentication information is sent via custom http request headers and not the actual payload itself

Header	Note
X-tranzila-api-app-key	Application key supplied by Tranzila
X-tranzila-api-request-time	Request time sent in Unix format (large integer counting milliseconds from Jan 1st, 1970 00:00:00
X-tranzila-api-nonce	A 40 bytes NONCE – unique random string generated with any random bytes function
X-tranzila-api-access-token	hash_hmac using 'sha256' on application key with secret + request-time + nonce. hash_hmac is available for all programming languages with samples found here: https://www.jokecamp.com/blog/examples-of-creating-base64-hashes-using-hmac-sha256-in-different-languages/

Client action to create a valid access token (variables names and code sample are in PHP)

1. Take the public app-key (`$appKey`)
2. Take the secret key (`$secretKey`)
3. Generate a NONCE – a random large string the can only be used once (`$nonce = bin2hex(random_bytes(40))`)
4. Generate a Unix timestamp (`$timestamp = time()`)
5. Generate the access-token (`$accessToken = hash_hmac('sha256', $appKey, $secretKey . $timestamp . $nonce)`);
6. Generate http custom headers

CODE SAMPLES

Important Notice:

Referred URL in following samples are Trazila Api endpoints. Each service has its own endpoint that should be taken from its documentation.

PHP

```
$json = trim($this->input->post('jsontext'));
$time = time();
$appKey = '<public app key>';
$secret = '<private app key>';
$nonce = bin2hex(random_bytes(40)); //actually 80 characters string
$accessToken = hash_hmac('sha256',$appKey, $secret . $time . $nonce);

$ch = curl_init('<<please replace this with service endpoint>>');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLINFO_HEADER_OUT, true);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $json);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
        'Content-Type: application/json',
        'Content-Length: ' . strlen($json),
        'X-tranzila-api-app-key: ' . $appKey,
        'X-tranzila-api-request-time:' . $time,
        'X-tranzila-api-nonce:' . $nonce,
        'X-tranzila-api-access-token:' . $accessToken
    )
);

$data = curl_exec($ch);
curl_close($ch);
return $json;
```

.NET

```
using System;
using System.Text;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Security.Cryptography;
using System.Security.Permissions; //.SecurityPermission;

namespace Rextester {

 public class Program {
  public static void Main(string[] args) {

   System.Net.ServicePointManager.DefaultConnectionLimit = 100;

   var url = "<<please replace this with service endpoint>>";

   var json = "{<request json}";
   using(var client = CreateClient(url)) {
    var httpContent = new StringContent(json, Encoding.UTF8, "application/json");
    var res = client.PostAsync(url, httpContent).Result;

    Console.WriteLine(res.Content.ReadAsStringAsync().Result);
   }
  }


  private static HttpClient CreateClient(string p_url) {
   var client = new HttpClient() {};
   client.DefaultRequestHeaders.Accept.Add(
    new MediaTypeWithQualityHeaderValue("application/json"));

   var mb = new MB().Build();

   client.DefaultRequestHeaders.Add("X-tranzila-api-app-key", mb.publicK);
   client.DefaultRequestHeaders.Add("X-tranzila-api-request-time", mb.unixTS.ToString());
   client.DefaultRequestHeaders.Add("X-tranzila-api-nonce", mb.hex);
   client.DefaultRequestHeaders.Add("X-tranzila-api-access-token", mb.access_token);
   Console.WriteLine(mb.unixTS.ToString());
   Console.WriteLine(mb.hex);
   Console.WriteLine(mb.access_token);

   return client;
  }

  public class MB {

   public string publicK = "yourpulickey";
   public long unixTS {
    get;
    set;
   }
   public string hex {
    get;
    set;
   }
   public string access_token {
    get;
    set;
   }
   Random rnd = new Random();
   string key = "yourprivatekey";

   public MB Build() {
```

```
    Byte[] b = new Byte[40];
    rnd.NextBytes(b);
    hex = ByteToString(b);
    unixTS = DateTimeOffset.UtcNow.ToUnixTimeSeconds();

    var encoding = Encoding.UTF8;
    var message = key + unixTS + hex;

    byte[] keyByte = encoding.GetBytes(publicK);
    byte[] messageBytes = encoding.GetBytes(message);
    using(var hmacsha256 = new HMACSHA256(messageBytes)) {
     byte[] hashmessage = hmacsha256.ComputeHash(keyByte);
     access_token = ByteToString(hashmessage);
    }

    return this;
   }

   string ByteToString(byte[] buff) {
    string sbinary = "";

    for (int i = 0; i < buff.Length; i++) {
     sbinary += buff[i].ToString("X2"); // hex format
    }
    return (sbinary.TrimEnd());
   }


   public string bin2Hex(string strBin)

   {

    int decNumber = bin2Dec(strBin);

    return dec2Hex(decNumber);

   }
   public int bin2Dec(string strBin) {
    return Convert.ToInt16(strBin, 2);

   }
   private string dec2Hex(int val)

   {

    return val.ToString("X");
   }
  }
 }
}
```

NODE.JS JAVASCRIPT

```
var CryptoJS = require("crypto-js");

function makeid(length) {
    var result           = '';
    var characters       = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
    var charactersLength = characters.length;
    for ( var i = 0; i < length; i++ ) {
       result += characters.charAt(Math.floor(Math.random() * charactersLength));
    }
    return result;
 }

var time = Math.round((new Date()).getTime() / 1000);
var nonce = makeid(80)
var key = '<app public key>';
var private = '<app private key>';
var hash = CryptoJS.HmacSHA256(key, private + time + nonce).toString(CryptoJS.enc.Hex)

var headers = {
  'X-tranzila-api-app-key':key,
  'X-tranzila-api-request-time':time,
  'X-tranzila-api-nonce':nonce,
  'X-tranzila-api-access-token':hash
}

var data = {
  <request json>
  }

request.post({'url': '<<please replace this with service endpoint>>',
              'headers': headers,
              'body':data,
              'json':true },
       function (err, data, response) {
              console.log(response)
              return callback(response);
    });
```

PYTHON

```
#!/usr/bin/python3
import hashlib
import hmac

import requests
import time
import binascii
import secrets
import json

public_key = 'app public key'
private_key = 'app private key'
timestamp = int(time.time())
nonce = str(binascii.hexlify(secrets.token_bytes(40)), 'utf-8')

access_key = hmac.new(bytes(private_key + str(timestamp) + nonce, 'utf-8'), bytes(public_key, 'utf-8'),
hashlib.sha256).hexdigest()


headers = {
    'X-tranzila-api-app-key': public_key,
    'X-tranzila-api-request-time': str(timestamp),
    'X-tranzila-api-nonce': nonce,
    'X-tranzila-api-access-token': access_key
}
url = '<<please replace this with service endpoint>>'

data = {
    "terminal_name": "terminalname",
    "document_id": "11823"
}

req = requests.post(url, headers=headers, data=json.dumps(data))
result = json.loads(req.text)
print(result)
```