PelicanCorp

USA North 811
(Northern California)

Webhook Member Guide

Version 1.0

Table of Contents

Statement of Confidentiality & Non- Disclosure

Version History

Version	Details	Authors	Date
N/A	Pre-analysis	Vincent Moreau	14/09/2021
1.0	Document creation	Mikaël Routhier	01/04/2022

Overview

The result of a request being created in OneCallAccess is a number of Notifications being created and sent to members of the service. A Member can choose to receive their Notifications via Web Hooks, this document outlines how to integrate with the OneCallAccess Web Hooks to receive the Notifications.

Specifications

To use the Web Hook option, you need to provide a HTTPs endpoint to which OneCallAccess will send a JSON structure containing the relevant files that make up a Notification (XML, GML and GIF documents).

Members using this feature will need to provide the following values:

1.1. URL

This is the URL which OneCallAccess will call for each Notification. The URL must accept a POST request. The URL must support HTTPS and must be reachable from the OneCallAccess environments.

1.2. Secret Key

A secret string that allows the receiver to verify that the request was sent from OneCallAccess and that the body of the message has been unchanged.

1.3. Custom HTTP headers (optional)

Some firewalls may require the inclusion of custom HTTP headers in requests to allow them to be proxied through the firewall.

OneCallAccess allows the inclusion of any number of custom HTTP headers in each Notification request (key/value).

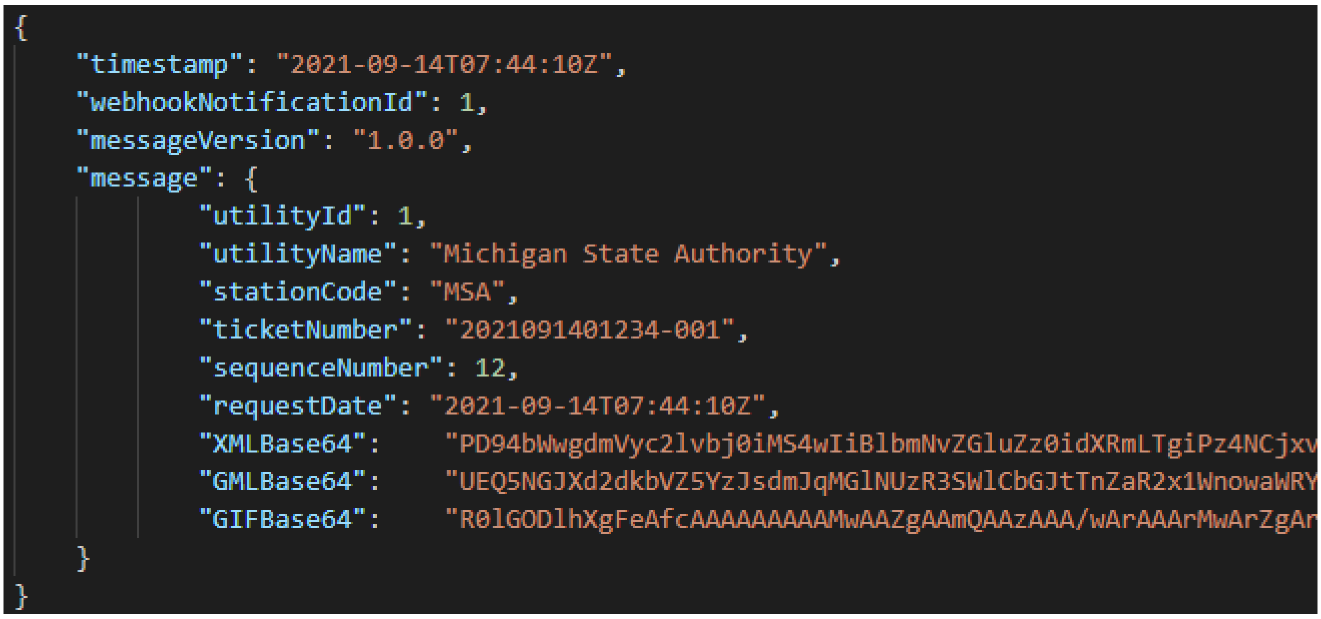
Payload

The payload is comprised of the message transmitted by OneCallAccess to the members of the external system. The following items are part of the payload:

2.1. Header

- **X-OneCall-Webhook-Signature (required)** : sha256=<hashvalue>
- **Content-Type (required)** : application/json

2.2. Request Body



```
{
    "timestamp": "2021-09-14T07:44:10Z",
    "webhookNotificationId": 1,
    "messageVersion": "1.0.0",
    "message": {
            "utilityId": 1,
            "utilityName": "Michigan State Authority",
            "stationCode": "MSA",
            "ticketNumber": "2021091401234-001",
            "sequenceNumber": 12,
            "requestDate": "2021-09-14T07:44:10Z",
            "XMLBase64":    "PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0idXRmLTgiPz4NCjxv
            "GMLBase64":    "UEQ5NGJXd2dkbVZ5YzJsdmJqMGlNUzR3SWlCbGJtTnZaR2x1WnowaWRY
            "GIFBase64":    "R0lGODlhXgFeAfcAAAAAAAAAMwAAZgAAmQAAzAAA/wArAAArMwArZgAr
    }
}
```

Figure 1: Request Body

Signature verification

The signature key that is provided allows the receiver to confirm that the Notification came from OneCallAccess and that the body of the message has not been altered since OneCallAccess issued the Notification.

The signing key is used to create a base64-encoded HMAC SHA-256 hash signature with each payload.

Each request includes an HTTP header:

- X-OneCall-Webhook-Signature: sha256=EXyLcM67FBwFXkyFu+qzy7UwEc5ytPCQK8UBFJJ/UsM=

The code sample provided below (C#) can be used to re-generate the hash (variable part of the signature) by passing the request payload (request body) and your secret key.

If the computed hash is different from the header, there will be a problem because the signature does not match.

```
/// <summary>
/// Create a Base64 hash in HMAC SHA256
/// </summary>
/// <param name="message">Message to hash</param>
/// <param name="secret">Secret key</param>
/// <remarks>
/// Example :
///    var hash = CreateBase64HashHMACSHA256("BodyMessage", "ThisIsMySecret");
///    will return: EXyLcM67FBwFXkyFu+qzy7UwEc5ytPCQK8UBFJJ/UsM=
/// </remarks>
/// <returns>base64 hash</returns>
private string CreateBase64HashHMACSHA256(string message, string secret)
{
   secret = secret ?? "";
   var encoding = new System.Text.ASCIIEncoding();
   byte[] keyByte = encoding.GetBytes(secret);
   byte[] messageBytes = encoding.GetBytes(message);
   using (var hmacsha256 = new HMACSHA256(keyByte))
   {
      byte[] hashmessage = hmacsha256.ComputeHash(messageBytes);
      return Convert.ToBase64String(hashmessage);
   }
}
```

Figure 2: Code Sample

For any other type of language, we would recommend this external resource. It may provide you with answers to get the same hash result. Please note that we are neither the author nor the owner of this resource and that the information contained herein is subject to change at any time.

HTTP Response

The receiver is expected to return an HTTP response code of 2XX within 3 seconds of the Notification being sent.

If the HTTP response code is not 2XX, or if the response is not received within 3 seconds, OneCallAccess will mark the Notification for retry.

OneCallAccess will keep retrying the request until a successful response is received or it has failed 10 times. The failed Notifications will appear in the OneCallAccess failed transmission queue which can be accessed through the Damage Prevention Portal.