WORLD-CHECK ONE API

HMAC Walkthrough – July 2022

REFINITIV®

An LSEG Business

CONTENTS

BACKGROUND

All requests made by an API user to the World-Check One API service need to include the user's API key, as well as be signed with a message authentication code (MAC). Specifically, this is expected to be a keyed-hash message authentication code (HMAC), which needs to be calculated by the client application using a combination of the request message contents and the API user's secret key. The purpose of this document is to show World-Check One API developers how to manually calculate this value correctly.

Assistance Refinitiv <u>can</u> provide: Refinitiv can show our client developers how to manually calculate the signature value properly and compare it to the signature generated within the provided Postman samples. Refinitiv also provides simple sample applications, written in Java, Node.js and C# that demonstrate code written in these languages to calculate HMAC. Refinitiv will also show within this document how a sending application can use a public site to determine what data is being sent – this can be used to verify the signature that is being calculated and being sent by the client application.

Assistance Refinitiv <u>cannot</u> provide: Refinitiv is not able to write code, do a code review or troubleshoot issues within our clients' proprietary software code.

Resources referenced within this document:

- World-Check One API HMAC Documentation: HMAC Document Link
- Sample applications download page: Sample Code Download
- HMAC-SHA256 Online Generator
- Online Character Counter
- Use Webhook.site to view what is being sent from your application in requests.

HMAC Steps:

1. Develop function/method in client code to generate HMAC signature from *HMAC signing text.*
2. HMAC signature is then sent in the authorization header of the World-Check One API request along with the API Key, algorithm used, hostname, and date. Content type and content length values are sent for any requests containing a body or payload, but omitted for requests without any body or payload.

A full textual description of Refinitiv's HMAC architecture and process are documented more fully in the HMAC Document. As a high-level description, when World-Check One API receives a client request, the HMAC signature calculated by the client including message body, API secret and **time** must match what is being simultaneously being generated by World-Check One API. An important note on this is that the sending application **must have a clock that is synched within 30 seconds of atomic time.** This can be checked on the sending machine via https://time.is. If time on the sending system is not within 30 seconds of global atomic time, World-Check One API will always return a 401 unauthorized error.

HMAC signature should be sent as a Base64 string.

GENERATE HMAC SIGNATURE

Following is the format of the *signing text* used as input to the HMAC calculation, designed and coded by each of our clients, to generate the HMAC signature for messages with and without bodies/payloads. These examples are *representative*. In the POST request below, client GroupID must be inputted correctly and the exact message size, including spaces, must be calculated and sent properly.

Without Payload:

(request-target): get /v2/groups
host: api-worldcheck.refinitiv.com
date: Wed, 13 July 2022 18:07:56 GMT

With Payload:

(request-target): post /v2/cases/screeningRequest
host: api-worldcheck.refinitiv.com
date: Wed, 13 Jul 2022 13:59:59 GMT
content-type: application/json
content-length: 192

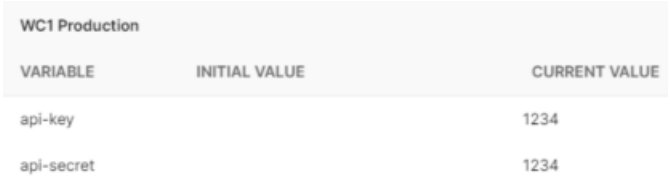
```
{
   "groupId": "client group ID",
   "entityType": "INDIVIDUAL",
   "providerTypes": ["WATCHLIST"],
   "caseScreeningState": {"WATCHLIST": "INITIAL"},
   "name": "John Smith"
}
```

COMPARE SIGNATURE WITH POSTMAN

The first recommended troubleshooting step if HTTP 401 errors are being returned to *client application* requests is to verify the sending system's clock is within 30 seconds of global atomic time. This can be verified on the sending system by visiting https://time.is. Once time synch is confirmed, the next troubleshooting step is to generate the signature with the appropriate World-Check One API Postman Sample Request. "Get My Top Level Groups" is representative of a GET request without payload, and "Perform Synchronous Screening: Simple" is representative of a POST request with a body.

Walkthrough of this comparison follows. Please note that these examples are using a dummy API Secret of "1234" for security. Although this is not a valid API Secret and any requests with an invalid API secret will return HTTP 401, the validity of the signature calculated by the client application can still be checked against the known valid signature calculated by Refinitv's Postman sample collection.

1. For purposes of this example we set the API Secret in Postman to be "1234". Remember that sending this request will always return an HTTP 401 error response, but we can still verify whether the client application's HMAC signature calculation is correct.



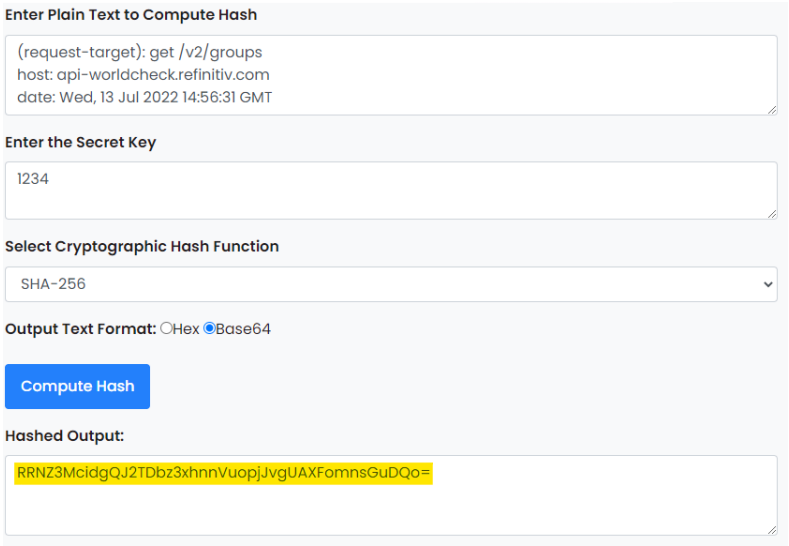
WC1 Production

VARIABLE	INITIAL VALUE	CURRENT VALUE
api-key		1234
api-secret		1234

2. Enable Postman console logging (view/Show Postman Console).
3. Send "Get My Top Level Groups" request from the Postman sample, and view the signature Postman generated:

Request Headers
Date: Wed, 13 Jul 2022 14:56:31 GMT
Authorization: Signature keyId="CLIENT API KEY",algorithm="hmac-sha256",headers="(request-target) host date",signature="RRNZ3McidgQJ2TDbz3xhnnVuopjJvgUAXFomnsGuDQo="
User-Agent: PostmanRuntime/7.29.0
*Accept: */**
Postman-Token: bd9df068-5505-44c4-9953-977e112b372b
Host: api-worldcheck.refinitiv.com
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

4. Using HMAC-SHA256 Online Generator Tool, type in the proper signing text, our dummy API Secret of "1234", SHA-256 for hash function, and output of Base64. Keep in mind that we are checking the validity of the signature as compared to Refinitiv's Postman sample, which we know is valid. So the "date" value must match exactly to what was sent from Postman. Here we can see that the HMAC-SHA256 generator tool generates a matching signature:



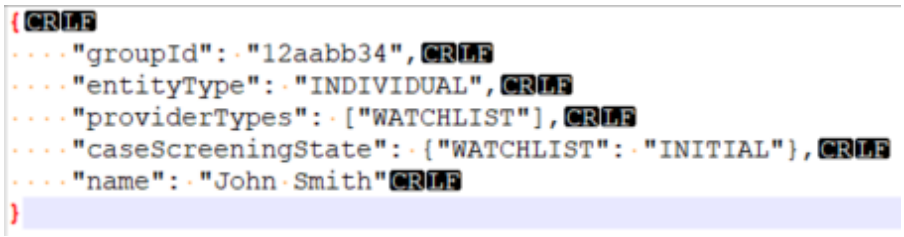
If the client application code's signature value does not match the signature generated by the Refinitiv Postman sample, which is known to be valid, <u>something is wrong with the HMAC signature calculation in the client application code.</u> This would need to be investigated by the client development team.

HTTP POST HMAC NOTES

While requests with no payload can omit values for content-type and content-length, requests with a payload need to include these values as well as include the actual payload in the HMAC signing text. For this example we are comparing our calculation with the Postman “Perform synchronous screening: simple” example and the following as the body/payload. Keep in mind a real group ID must be used in an actual request. This body is using a dummy groupID as well as the dummy API secret “1234”.

```
{
   "groupId": "12aabb34",
   "entityType": "INDIVIDUAL",
   "providerTypes": ["WATCHLIST"],
   "caseScreeningState": {"WATCHLIST": "INITIAL"},
   "name": "John Smith"
}
```

This payload is 175 characters in size, checked using [this online character counter](). A text utility such as Notepad ++ can be used to observe blank spaces (red dots), which are counted, and to check for extraneous new lines.



Note that Notepad ++ shows that this text is “length 181” including the new lines. Subtracting those six gives the actual content length of 175 characters.

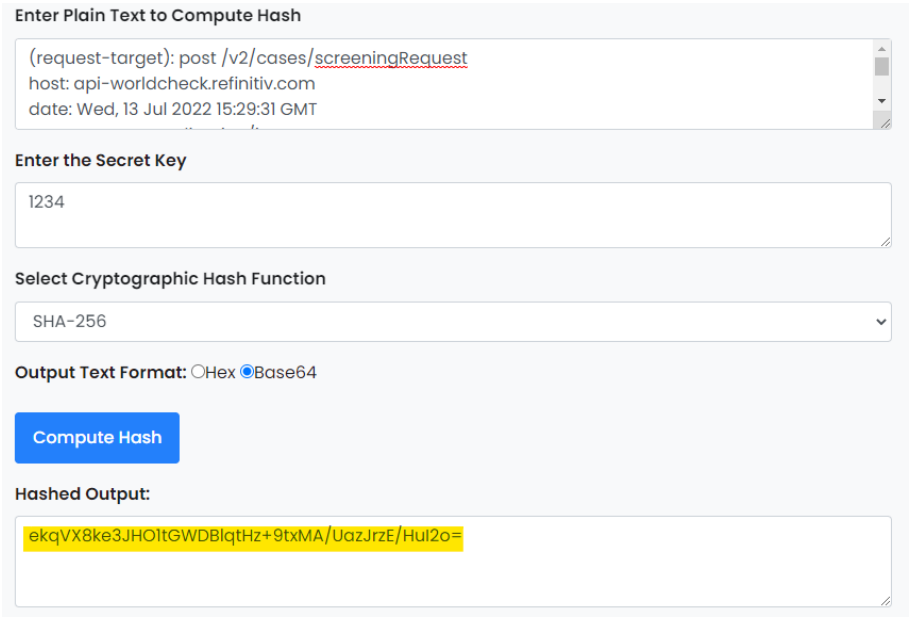
Sending this request using the Postman sample shows the following signature in the Postman console:

Request Headers
Date: Wed, 13 Jul 2022 15:29:31 GMT
Content-Type: application/json
Authorization: Signature keyId="ACTUAL API KEY REMOVED" algorithm="hmac-sha256",headers="(request-target) host date content-type content-length",signature="ekqVX8ke3JHO1tGWDBlqtHz+9txMA/UazJrzE/HuI2o="
Content-Length: 175
User-Agent: PostmanRuntime/7.29.0
*Accept: */**
Postman-Token: fca5735d-a0a5-409f-b5a4-8c5b4aeb1ed8
Host: api-worldcheck.refinitiv.com
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

Format of the signing text for the HMAC calculation in this example is:

(request-target): post /v2/cases/screeningRequest
host: api-worldcheck.refinitiv.com
date: Wed, 13 Jul 2022 15:29:31 GMT
content-type: application/json
content-length: 175

```
{
   "groupId": "12aabb34",
   "entityType": "INDIVIDUAL",
   "providerTypes": ["WATCHLIST"],
   "caseScreeningState": {"WATCHLIST": "INITIAL"},
   "name": "John Smith"
}
```

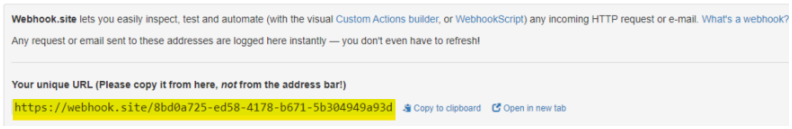



This calculation matches. If the client application is not calculating a matching HMAC signature, the World-Check One API client's development team will need to troubleshoot the issue within their application code.

TROUBLESHOOTING WITH WEBHOOK

Refinitiv has recommended https://webhook.site to verify what signature is being sent across the wire from client applications. As in the previous examples, sending a request to the Webhook site will not return a successful response to the client application, but in a browser our client can see the signature being created if this will help to prove what signature is being sent.

Visiting https://webhook.site will provide a unique API endpoint to which requests should be sent for viewing. For example, the highlighted URL below is what should be configured as the API endpoint for the request to be sent to:



Details on the request that was sent from the client application can then be verified in the browser:

Headers

connection	close
accept-encoding	gzip, deflate, br
host	webhook.site
postman-token	5a3ab41d-2aee-4ea2-87df-773c044456a6
accept	*/*
user-agent	PostmanRuntime/7.29.0
content-length	175
authorization	Signature keyId="",algorithm="hmac-sha256",headers="(request-target) host date content-type content-length",signature="YQVLUUVC+vi13HSWr9LxeZ89Ueo9jU2VnoeY2GC9ZsY="
content-type	application/json
date	Wed, 13 Jul 2022 15:46:16 GMT

Raw Content Format JSON Word-Wrap Copy


```
{
  "groupId": "12aabb34",
  "entityType": "INDIVIDUAL",
  "providerTypes": [
    "WATCHLIST"
  ],
  "caseScreeningState": {
    "WATCHLIST": "INITIAL"
  },
  "name": "John Smith"
}
```

SUMMARY

- Clients should utilize the resources provided to troubleshoot HTTP 401 errors received while developing their application. These resources include Refinitiv's Postman sample applications (which include HMAC code within the "prerequest script" section), the three HMAC code examples, and the HMAC document available on Refinitiv Developer Community.
- The Date value sent by client applications must be within 30 seconds of global atomic time or an HTTP 401 error will always be returned.
- The content-size value sent for requests with payloads must be exact, including spaces and without trailing characters/line breaks.
- Client applications must be developed to properly calculate the HMAC signature to match what is calculated at the same time by the World-Check One API platform.
- Refinitiv is glad to review/present this information to our client developers during implementation of World-Check One API, however we cannot write code outside the sample applications that are provided, or troubleshoot/debug code that has been written by clients.
- Most issues encountered by clients are either due to clock being out of synch on the sending application, improper creation of HMAC signing text, message size calculation being incorrect, or challenges within very specific development platforms. Refinitiv can assist with an overview of creating the signing text, but not very specific issues within specific application environments.
- The authentication header including the signature must be sent in restful HTTP – Refinitiv has found in the past that CORS Environments are generally not compatible.