Certification Audit Report

AlpineBits HotelData 2022-10 Server

Certified Organization	**NOI Techpark**
Certification Organization	
Auditor	**Davide Montesin / Catch Solve**
Version	**2022-10**
	Server
Date/Period of audit	**Sep 26, 2024 - Dec 9, 2024**
Location	**Online**

General

The audit took place between 26th September and 9th December with the purpose to prove the AlpineBits V2022-10 standard conformity of NOI Techpark Open Data Hub server endpoint https://alpinebits.opendatahub.com/AlpineBit.

NOI Techpark Open Data Hub server has been tested against all related fundamental rules and characteristics described by the AlpineBits Standard V2022-10. All functional tests have been executed with a dedicated client software (https://development.alpinebits.org/backend/certification/web), which permits to invoke test request messages for each single rule and to evaluate their relative responses from the tested server. All XML requests and responses have been validated through the official AlpineBits XSD + RNG files and OTA XSD files provided by the tool..

The customer also provided us by email (13/09/24, 16:38) the data we sent to the database to double check that was exactly what we sent.

What hasn't been validated

The certification audit doesn't consider following criterias:

- software and hardware stability
- software and hardware security
- request frequency and stress tests

Actions validated

Following action and message types have been validated in order they are listed:

- HANDSHAKING
    - action_OTA_Ping
- INVENTORY
    - 4.4.1 action_OTA_HotelDescriptiveContentNotif_Inventory (Inventory/Basic push)
        - OTA_HotelDescriptiveContent_Notif_Inventory_use_rooms
        - OTA_HotelDescriptiveContentNotif_Inventory_occupancy_children
    - 4.4.3 action_OTA_HotelDescriptiveInfo_Inventory (Inventory/Basic pull)

    - 4.4.5 action_OTA_HotelDescriptiveContentNotif_Info (Inventory/Hotelinfo push)
    - 4.4.7 action_OTA_HotelDescriptiveInfo_Info (Inventory/Hotelinfo pull
- FREEROOMS
    - 4.1.1 action_OTA_HotelInvCountNotif
        - OTA_HotelInvCountNotif_accept_rooms
        - OTA_HotelInvCountNotif_accept_categories
        - OTA_HotelInvCountNotif_accept_complete_set
        - OTA_HotelInvCountNotif_accept_out_of_order
        - OTA_HotelInvCountNotif_accept_out_of_market
        - OTA_HotelInvCountNotif_accept_closing_seasons

Server does not support gzip: ERROR:GZIP compression unsupported (Content-Encoding header with value "gzip" found in request header

Server credentials

AlpineBits Server URL	https://alpinebits.opendatahub.com/AlpineBits
Credentials used:	
HotelCode	123
X_AlpineBits_ClientID	noi-client-test
X_AlpineBits_ProtocolVersion	2022-10

Result

While the audit on the customer's server has been tested on about ~100 rules described by the AlpineBits standard 2022-10 and listed by the tool (some rules were out of scope in this case). Only some minor details were found. This was already reported by e-mail at 26/09/24, 18:40.

1. **authentication**: in the server implementation, empty or wrong authentication means that the response contains open data

2. **hotel-info pull before push**: reading hotel information before the first push will generate an error, but is not a typical use case
3. **missing action or request**: missing action or request the server will return an internal server error (500) like he has handled the missing parameter unexpectedly

We have assumed that this server does not support reading of EXIF copyright informations from images and does not implement "server may request an update of the capabilities to client"[1]

Given all this, we declare the implementation compliant

Suggestions and future steps

The issue related to the open data reported in the previous section is the most important condition for compliance. The AlpineBits standard does not specify what exactly to do in these cases. However, it would be recommended to look into this issue with the AlpineBits alliance to find a way regulated by the standard. The current implementation is not very convenient for developers and debugging. The handshake function is used precisely to verify the correctness of the authentication, before proceeding with the other calls. If the response has no difference whether the authentication is passed or failed, there is no information to debug any user-related issues that are not there maybe in the beginning but may appear unexpectedly over time.

If every user and password returns the same data for handshake, this is likely to create difficulties at the organizational level. We recommend limiting the open data situation to a more restricted user/password combination, such as “open” or empty user/password. Or return at least two somewhat different responses in case of passed or failed authentication, such as a non-blocking warning message in the second case.

Bolzano, 9 December 2024

Daide Montesin

[1] if not please report us for do this check too