

RMS Interface Description

Version 1.03

NTDSC RMS data provision interface description

30/11/2022

Hungarian Tourism Agency

Document control

Version	Phase	Published	Responsible person	Reason/Changes
v1.00	Accepted version	04/04/2022	HTA	First published version
v1.01	Updated version	20-08-2022	HTA	Second published version
v1.02	Updated version	20-09-2022	HTA	Expand interface based on feedback, transfer field-level specifications from Excel attachment
v1.03	Updated version	30-11-2022	HTA	Expand interface based on feedback

Change log

Changes introduced in version v1.03

- The validation rule for the field *Order summaries* (rendelesOsszesitok) / *Order end* (rendelesVege) has been updated. The new validation rule is the following: $rendelesVege - rendelesKezdetek \leq 24 \text{ hours}$
- The validation rule for the field *Daily closure* / *Transaction date* (targynap) has been deleted: unique constraint (NTDSC registration number + transaction date + software ID)
- The value set for the field *Order summaries* (rendelesOsszesitok) / *Order items* (rendelesTetelek) / *Subcategory* (alkategoria) has been updated (see Appendices):
 - New subcategories:
 - Main course with side dish (FOETEL_KORETTEL)
 - Food package (ETELCSOMAG)
 - Beverage package (in all three beverage-related main categories: ITALCSOMAG)
 - Noncarbonated beverages (SZENSAVMENTES_UDITO)
 - Non-catering (NEM_VENDEGLATAS)
- The validation rule for the field *Daily closure* / *Data provision classification of the transaction date* (targynapBesorolasa) has been corrected: from this version onwards sending an ADOTT_NAPON_ZARVA value does not limit the processing of further daily closure messages sent for the same transaction date.
- A detailed change log section was added to the document.

Changes introduced in version v1.02

- Data provision message samples have been added to the document.
- A new chapter titled *Verification of submitted data* has been added.
- A new chapter titled *Specification description of messages* has been added.
- The detailed description of the possible error codes on the RMS Interface has been added to the chapter titled *Error keys on the RMS interface*
- Further information has been added to the *Types of RMS messages, messaging process* chapter regarding the rules applicable in case of messages that need to be resent.
- The *Submitting an order summary* chapter has been modified to include that the developers of RMS systems need to prepare their software to handle occasional changes to value sets.
- The validation rule for the field *Daily closure* / *Transaction date* (targynap) has been updated: it will be possible to submit more than one daily closure for one transaction date, provided that the periods of time in the messages do not overlap.
- The validation for maximum character limit in the fields *RMS Order identifier* (rmsRendelesAzonosito) and *Order summary identifier referenced by RMS* (hivatkozottRendelesOsszesito) has been deleted.
- The JSON name for the field *Combined data provision* has been changed. Its new name is *osszesitett*.
- The validation for the field *Order summaries* (rendelesOsszesitok) / *Payment Information* (fizetesInformaciok) / *Total sum of order in HUF* (rendelesVegosszegeHUF) has been updated.

- The validation for *Order summaries* (rendelesOsszesitok) / *Payment Information* (fizetesInformaciok) / *Means of payment* (fizetesModok) / *Sum paid in HUF* (fizetettOsszegHUF) has been updated.
- The value set for the field *Order summaries* (rendelesOsszesitok) / *Payment Information* (fizetesInformaciok) / *Means of payment* (fizetesModok) / *Payment method* (fizetesMod) has been updated:
 - New enum
 - Rounding (KEREKITES)
- The formula for calculating the payment method rounding has been added to the document.
- The value set for the field *Order summaries* (rendelesOsszesitok) / *Order items* (rendelesTetelek) / *Subcategory* (alkategoria) has been updated.
 - New subcategories:
 - Discount (KEDVEZMENY)
 - Plastic packaging, environmentally friendly packaging (MUANYAG_CSOMAGOLAS, KORNYEZETBARAT_CSOMAGOLAS)
 - Delivery fee (KISZALLITASI_DIJ)
- The validation rule for the fields *Order summaries* (rendelesOsszesitok) / *Quantity* (tetelszam), and *Amount* (menyiseg) has been updated to allow the value 0.
- The validation rule for the fields *Order summaries* (rendelesOsszesitok) / *Order items* (rendelesTetelek) / *Item aggregator* (tetelOsszesito) has been updated. Hereafter it is mandatory to use the rounding method described (rounding by merchant).
- The JSON name for the field *Daily closure data* has changed. Its new name is *zarasiInformaciok*.
- The validation rule for the field *NTDSC system unique processing identifier* (feldolgozasAzonosito) has changed. The validation for maximum character limit has been deleted.

GLOSSARY	7
1. INTRODUCTION	9
1.1. THE PURPOSE OF THE DOCUMENT	9
1.2. TECHNICAL DESCRIPTION OF THE RMS DATA PROVISION INTERFACE	10
2. CONNECTION PROCESS OF RMS SOFTWARE	11
2.1. SUBMITTING A CONNECTION REQUEST FOR THE RMS SOFTWARE	12
2.2. PREPARING THE TEST ENVIRONMENT FOR CONNECTING THE RMS SYSTEM	12
2.3. GENERATING AND SENDING A DIGITAL CERTIFICATE SIGNING REQUEST	13
2.4. PROVIDING THE INFORMATION NECESSARY FOR CONNECTING THE RMS SYSTEM	13
2.5. DEVELOPMENT OF THE RMS SYSTEM	15
2.6. COMPLETING TESTS IN THE NTDSC VENDÉGLÁTÁS SYSTEM'S TEST ENVIRONMENT	15
2.7. ASSESSMENT OF VALIDATION TESTS	15
2.8. ISSUING A CERTIFICATE	16
3. CONFIGURATION FOR LIVE DATA TRANSMISSION	17
3.1. STORING THE NTDSC VENDÉGLÁTÁS SYSTEM IDENTIFIER OF THE RMS SOFTWARE	17
3.2. REGISTRATION OF SERVICE PROVIDER	17
3.3. REGISTRATION OF THE CATERING ESTABLISHMENT	18
3.4. ISSUING A CERTIFICATE AND ASSOCIATING A CATERING ESTABLISHMENT WITH THE RMS SYSTEM	18
3.5. RMS SYSTEM CONFIGURATION IN A LIVE ENVIRONMENT	19
3.6. COMMENCEMENT OF LIVE DATA TRANSMISSION	20
4. TYPES OF RMS MESSAGES, MESSAGING PROCESS.....	21
4.1. SUMMARY	21
4.2. ORDER OF DATA SUBMISSIONS	22
4.3. SUBMITTING AN ORDER SUMMARY	22
4.4. SUBMITTING DAILY CLOSURE.....	24
4.5. VERIFICATION OF SUBMITTED DATA.....	24
4.6. SPECIFICATION DESCRIPTION OF MESSAGES.....	25
4.6.1. <i>Structure of message header.....</i>	25
4.6.2. <i>Structure of order summary message</i>	27
4.6.3. <i>Daily closure message structure</i>	38
4.6.4. <i>Verification endpoint message structure.....</i>	40
4.6.5. <i>Response messages</i>	41
5. PROCESS DESCRIPTION, TROUBLESHOOTING AND FORMAL REQUIREMENTS	43
5.1. DATA PROVISION PROCESS	43
5.2. ERROR KEYS ON THE RMS INTERFACE.....	46
5.2.1. <i>Field value error keys.....</i>	46
5.2.2. <i>NTDSC registration error keys.....</i>	47
5.2.3. <i>Signature error keys.....</i>	47
5.2.4. <i>Certificate error keys</i>	47
6. APPENDICES	49

6.1.	REQUESTING A CERTIFICATE	49
6.1.1.	<i>Standards and Recommendations</i>	49
6.1.2.	<i>Creating a certificate signing request in a Windows environment</i>	49
6.1.3.	<i>Create a certificate signing request in a Linux environment</i>	50
6.2.	INTERFACE MESSAGE EXAMPLES	51
6.2.1.	<i>Normal order summary message</i>	51
6.2.2.	<i>Corrective order summary message</i>	54
6.2.3.	<i>Cancelled order summary message</i>	55
6.2.4.	<i>Daily closure message</i>	57
6.2.5.	<i>Verification message</i>	58
6.3.	CATEGORIES	59

Glossary

This chapter describes the terms and definitions that are necessary to understand and implement the data reception interface.

Term, abbreviation	Definition
CSR – Certificate Signing Request	Certificate Signing Request. A unique electronic certificate needs to be used for the purpose of identifying catering establishments and the software version used there to enable data provision in the NTDSC System. The User who represents the service provider generates a certificate signing request (CSR) – with support from the provider of the RMS system – and uploads it to the NTDSC interface.
JKÜ	Authorization Manager Agent (JKÜ), a central service of NISZ Zrt., which allows to check the permissions of the logged-in person to manage transactions for a given tax number (for a registered company or sole trader).
HTA	Hungarian Tourism Agency Limited
Daily closure	A closing/summary dataset to be submitted to NTDSC for inspection when a given catering establishment closes a business day or at least every 24 hours.
NTDSC RMS certificate	An electronic certificate issued by the NTDSC system based on the certificate signing request (CSR) uploaded into the NTDSC system which is used by the RMS software running at the service location for identification and authentication purposes.
Catering establishments under a data provision obligation towards NTDSC	A catering establishment under a data provision obligation towards NTDSC is an establishment that is required to fulfil its obligation to issue receipts with the use of online cash registers pursuant to Ministry for National Economy Decree 48/2013. (XI. 15.) NGM on the technical requirements of cash registers, the marketing, use, and servicing of cash registers used for issuing receipts, and the reporting of data recorded by cash registers to the tax authority. Based on this decree, the use of online cash registers is mandatory for activities bearing NACE codes 56.1 and 56.3, except where services are provided on the move.
NTDSC VENDÉGLÁTÁS Portal	The interface of the NTDSC VENDÉGLÁTÁS system where the operators of catering establishments register.
NTDSC VENDÉGLÁTÁS system	A system receiving the data and messages – determined herein – sent by the catering establishments.
Order summary	The sum of items consumed by one person or a company of persons in a given catering establishment.
Order item	The sum of sales and descriptive data generated in relation to a product consumed or purchased at a given time.
RMS	Restaurant Management System.

RMS interface	An interface provided by the NTDSC VENDEGLÁTÁS system for submitting data.
ROA	Registry of authorizations (https://rendelkezes.gov.hu/). A person authorized for representing a given service provider may use this to authorize another person to act in cases related to the service provider in the NTDSC system.
Service location	A service location is a collective place located at the address of a service provider where various touristic services are provided. A service location may be associated with more NTDSC systems.
Service location identification number	Every service location registered in the NTDSC system is issued an identification number which comprises the last two digits of the year of the registration and a rolling serial number. E.g.: 22000055. This is the identification number to be provided in the certificate signing request.
Service provider	Pursuant to effective legislation, the operator, i.e. owner or tenant, of any catering establishment operating in Hungary who has the right to represent the catering establishment and register it at NTDSC.
Catering establishment	Catering establishments with an obligation to register at NTDSC under Government Decree 210/2009. (IX. 29.) on the Conditions for Performing Commercial Activities. More than one catering establishments can be registered at one service location (e.g. restaurant, bar, café).
NTDSC registration number of a catering establishment	Every catering establishment registered in NTDSC is issued a registration number that the NTDSC system generates from the identification number of the service location associated with it, indicating the type of the catering establishment. E.g.: ET22000055

1. Introduction

The National Tourism Data Supply Centre (NTDSC) supports the digitalization, development and whitening of the tourism industry and is therefore indispensable for improving the competitiveness of the industry. Currently, not every enterprise directly associated with tourism is included in the system, therefore, widening the scope is in the interest of the national economy. The catering industry plays a significant role in the national economy, although a considerable part of the industry is involved in the black and grey economy, too.

During the development of the NTDSC system, in addition to analysing the turnover data of accommodation establishments, unified registration, data provision and statistical data analysis will be implemented for catering establishments in Hungary. The NTDSC VENDÉGLÁTÁS system is a part of the NTDSC information technology system that receives data from catering establishments, stores, processes these data, and prepares structured reports, risk analyses, and makes these available to the actors of the segment providing the data and to the authorities specified in law.

To achieve the defined goals, an RMS data provision interface will be established within the NTDSC VENDÉGLÁTÁS system through which catering establishments will be able to submit their turnover data to the NTDSC system.

1.1. The purpose of the document

This document describes the prerequisites for integrating the NTDSC VENDÉGLÁTÁS system with the catering software (hereinafter as “RMS system”), the necessary configuration for integration on the RMS side, as well as its communication channels and the rules for data transmission. The scope of the document does not include guidelines for creating the internal data structure, architecture, or modules of the RMS software; it only contains the information strictly necessary for connecting to the NTDSC VENDÉGLÁTÁS system.

Prerequisites for the commencement of data provision by a given catering establishment:

- NTDSC registration;
- Use of RMS software with a certificate on NTDSC data provision capability;
- Requesting, downloading an NTDSC RMS certificate, and making it available to RMS software.

This document contains the following chapters:

- Integration process of RMS software;
- Configuration for live data transmission;
- Types of RMS messages;
- Formal requirements;
- Sample messages

1.2. Technical description of the RMS data provision interface

RMS interface is a data reception service based on REST, which, to allow large workloads, processes data transmissions asynchronously.

The sender RMS system must sign the submitted messages with the NTDSC RMS certificate requested on and downloaded from NTDSC's interface, thus ensuring the authenticity and integrity of the data. The same certificate is to be used to authenticate and identify the RMS version when the TLS/SSL connection is established.

The submitted data will be received by a front-end application that will check the digital signature of the message. Afterwards, it will be checked whether the data submitted contain the fields corresponding to the interface description, and whether every mandatory field has been filled out. If a message is acceptable, the front-end application will generate a **unique processing identifier** which the calling system will receive immediately in the response. The front-end application will forward the message received to NTDSC core systems for processing. This identifier can be used subsequently to check status of the processing of the message submitted.

After this, the messages will be forwarded to NTDSC VENDEGLÁTÁS system for processing.

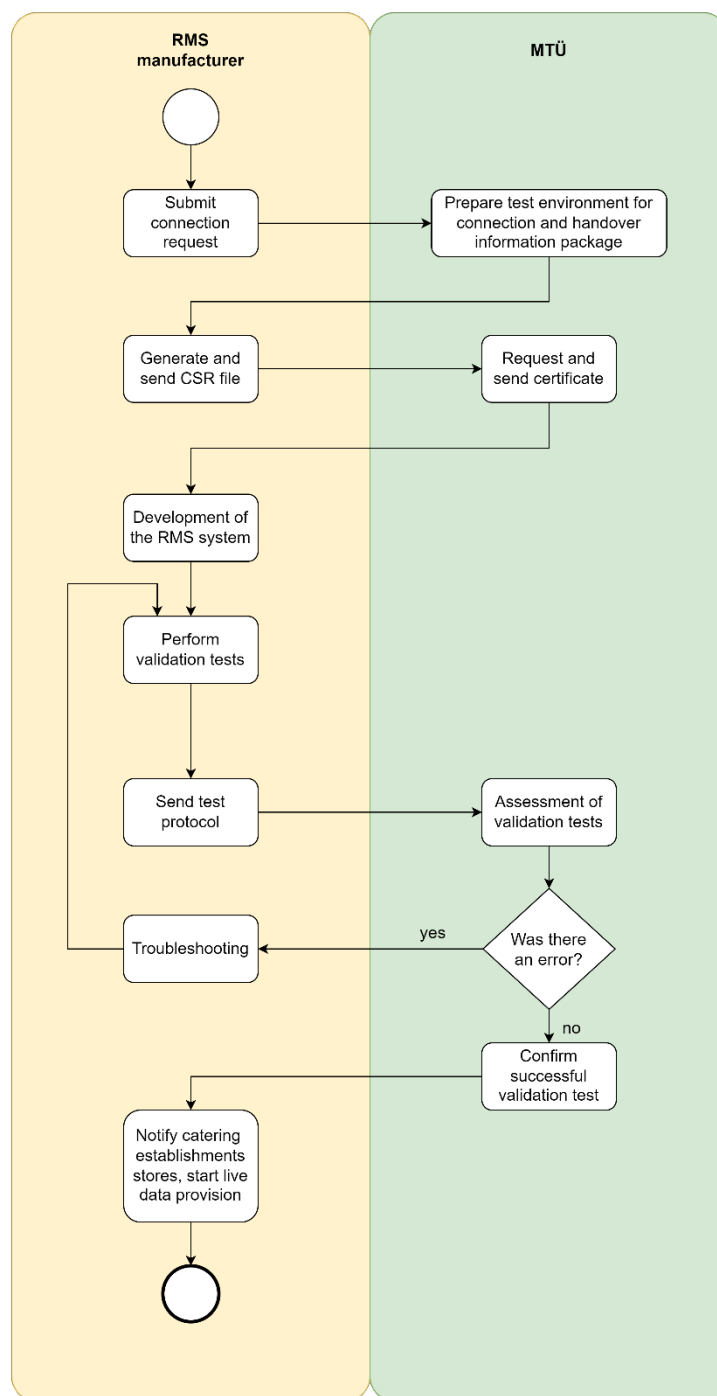
The benefits of this solution:

- due to asynchronous processing, the RMS interface can serve a higher load than services implementing synchronous processing;
- with the help of the verification endpoint, the call systems may query the final processing result of the data submitted. This ensures that the data have been properly received and processed;
- asynchronous processing allows for more complicated, albeit more time-consuming, checks to be performed on the messages received.

2. Connection process of RMS software

This chapter details the steps to be taken for integrating the RMS software intended for data transmission into the NTDSC VENDÉGLÁTÁS system.

The flowchart below indicates the steps required for a successful connection. The process steps shown in the diagram will be detailed later on.



Steps of the RMS software connection process:

- a. Submitting the connection request – Responsible Person: RMS manufacturer
- b. Preparing the test environment – Responsible Person: HTA
- c. Generating the CSR files necessary for requesting the certificates – Responsible Person: RMS manufacturer
- d. Requesting certificates in the NTDSK VENDÉGLÁTÁS system test environment – Responsible Person: HTA
- e. Sending the information package and reference client to RMS manufacturer to support integration – Responsible Person: HTA
- f. Development of the RMS system – Responsible Person: RMS manufacturer
- g. Performing validation tests and sending test report – Responsible Person: RMS manufacturer
- h. Assessment of validation tests – Responsible Person: HTA
- i. Troubleshooting if necessary – Responsible Person: RMS manufacturer
- j. If tests are successful, issuing a certificate to verify successful connection – Responsible Person: HTA

2.1. Submitting a connection request for the RMS software

Service locations registering to the NTDSK VENDÉGLÁTÁS system may select an RMS system to implement their data provision with. The RMS software manufacturer must submit a preregistration request to the HTA for the RMS system to become available among the selectable elements. The request must contain all necessary information of the RMS software.

List of data to be submitted in advance:

- RMS manufacturer data (company details, contact details, public information);
- IP address or domain of the RMS system's development and test environment. Download the connection datasheet from the following link and send it to dev.support@ntak.hu when completed.

 [Download connection form](#)

2.2. Preparing the test environment for connecting the RMS system

Once a connection datasheet has been approved, HTA will implement the following settings in the NTDSK VENDÉGLÁTÁS system's testing environment to enable the manufacturer of the RMS system to begin the integration:

- The RMS system is added to the NTDSC VENDEGLÁTÁS system's testing environment as an eligible RMS system.
- Following this, a test provider and a test service location will be registered for the connection process. The data relating to these, including the tax number of the testing provider and the identification number of the service location of the test and the registration number of the catering establishment, will be sent to the manufacturer of the RMS system. The manufacturer will need these data for the data provision.

2.3. Generating and sending a digital certificate signing request

The next step in the process is to generate a digital certificate signing request that is required for requesting a certificate, further help is included in the Annex.

 *CSR files are to be sent via e-mail to dev.support@ntak.hu. Please, indicate in the subject the name of the RMS system to be connected and the identification number of the test service location.*

2.4. Providing the information necessary for connecting the RMS system

After receiving the CSR files, HTA employees will request a certificate for the RMS manufacturer.

When the certificate has been generated, HTA will send the information necessary to access the test environment as well as the configuration for data transmission to the RMS software manufacturer.

These information are the following:

- Contact details of NTDSC VENDEGLÁTÁS testing systems;
- The certificate of the service location that can be used for testing;
- Identification data necessary for testing the RMS software;
- Swagger description URL;
- Reference client;
- Description of validation tests and a test report template

Contact details of NTDSC VENDEGLÁTÁS testing systems

The contact details of NTDSC VENDEGLÁTÁS testing systems include the REST endpoints to be used for testing data transmissions and the address of the testing system. HTA may provide access for the RMS manufacturer to the test systems in two ways: With a VPN profile or with IP filtering.

If a VPN profile is used, the VPN profile included in the reply message needs to be correctly configured to access the environment. HTA maintains the right to *whitelist* the IP domain indicated by the RMS manufacturer instead of using a VPN profile, and to provide access to the test environment this way. HTA will notify the RMS

manufacturer via e-mail about the method by which access will be provided to the RMS interface (VPN or IP address filtering).

The certificate of the service location that can be used for testing

A prerequisite for transmitting data to NTDSC VENDÉGLÁTÁS system's testing environment is to use a unique NTDSC RMS (test) certificate for every service location to identify the sender when establishing the SSL connection, as well as the electronic signature of the message and attaching a certificate to the message sent. Subsequent chapters will provide more detail on the use of the certificate.

Providing test environment identification data

In data transmissions to the NTDSC VENDÉGLÁTÁS system, the RMS software must identify, in the body of the message sent, the sender's service location, its service provider and the RMS software itself.

These test identifiers will be sent to the RMS software manufacture by HTA during the software testing phase in its reply to the RMS software connection request.

These identification data are the following:

- **NTDSC registration number of a catering establishment:** (e.g. ET22012345) The registration number of the catering establishment includes the identifier of the service location (e.g.: 22012345), too, and therefore can be compared to the identifier of the service location in the certificate. This data should be sent to the RMS interface using the "szolgponRegSzam" field.
- **Service provider's tax number:** Service provider's tax number is the tax number used for identifying the entity associated with the test service location. This data should be sent to the RMS interface using the "adoszam" field.
- **NTDSC system identifier of the RMS system:** It is also necessary for the testing that the RMS software identify itself. This data should be sent to the RMS interface using the "rmsRendszerNTAKazonosito" field.

Reference client

The reference client is a complete environment created in the Postman application with the help of which you can become familiar with the structure of the messages to be sent. Furthermore, it contains ready sample messages with which data provision can be tried and tested.

Description of validation tests

After the development of the RMS system, the manufacturer is to verify in a validation test that the RMS system created by it is able to properly send the necessary data to the NTDSC VENDÉGLÁTÁS system and can query the results of the processing of such data and, in case of an error, resend corrected data.

During the validation test, the manufacturer of the RMS software needs to perform test cases determined by HTA. HTA will provide the description of the tests within an information package.

Test report

During validation testing, the manufacturer of the RMS software will need to record the following data in a test report:

- Name of the RMS software;
- Identifier of the RMS software in the test NTDSC system;
- identification data of the test service location;
- testing window;
- marking the test cases completed;
- processing identifiers received for the messages;
- screenshots of the RMS system that prove that this was where the transactions occurred;
- other relevant information.

HTA will provide the RMS manufacturer with a sample test report within the information package.

2.5. Development of the RMS system

The next step in the process is to develop the RMS system, to establish data transmission capability towards NTDSC. After the development, and once the aforementioned configuration settings have been implemented, the RMS software can send data to the NTDSC VENDÉGLÁTÁS system's testing environment, use all the test endpoints described in the RMS interface document.

During the development, the reference client may provide support, or you may request assistance at dev.support@ntak.hu.

Please, indicate in the subject of the e-mail the name of the RMS system to be connected and the identification number of the test service location.

2.6. Completing tests in the NTDSC VENDÉGLÁTÁS system's test environment

Data submissions will be subjected by HTA to technological validation tests and content verification, too.

Validation tests will be performed in the timeframe selected by the RMS manufacturer, the test cases required must be completed within this window. During the validation tests, the required information must be recorded in the test report, the test report and the information to be recorded therein are included in the information package received during the connection. The test report is to be sent to the following e-mail address: dev.support@ntak.hu. Please, indicate in the subject of the e-mail the name of the RMS system to be connected and the identification number of the test service location used for the connection.

2.7. Assessment of validation tests

HTA staff will examine the messages received for form and content.

If an error is detected, they will communicate it to the manufacturer of the RMS software. After the errors have been corrected, the process will continue from the chapter Development of the RMS system.

2.8. Issuing a certificate

If a particular version of the RMS software successfully passes the validation test, HTA will issue a relevant certificate. Following this, the RMS software will be certified to be deployed for live data provision.

3. Configuration for live data transmission

If the RMS software successfully passes the required validation test, HTA will enter the RMS software in the registry of the NTDSC VENDÉGLÁTÁS system and send the live environment identifier of the RMS software to the manufacturer of the RMS software.

After this, catering establishments registering to the NTDSC VENDÉGLÁTÁS system may select the RMS software that has a Certificate and a live environment identifier for data transmission.

3.1. Storing the NTDSC VENDÉGLÁTÁS system identifier of the RMS software

The RMS software needs to store the RMS identification number received after the successful validation. The identification number is a string of characters that identifies the RMS software and does not depend on the service provider. Every catering establishment must attach the same identifier to every data submission that uses the same RMS. The certificate used for data provision includes, among others, the RMS identifier, too.

The steps to be taken by the service provider using a particular RMS are explained in the following. These consist of the following steps:

- NTDSC registration of the service provider
- NTDSC registration of service locations and catering establishments;
- Providing / selecting the RMS used;
- Generating a digital certificate signing request (if necessary, with support from the RMS manufacturer);
- Uploading a certificate signing request to the NTDSC VENDÉGLÁTÁS system;
- Downloading the completed certificate and forwarding it to the RMS manufacturer.

The User Guide of the NTDSC VENDÉGLÁTÁS system, which can be found of info.ntak.hu, describes these steps in detail.

3.2. Registration of service provider

Registering a service provider in the NTDSC VENDÉGLÁTÁS system is a prerequisite for data transmissions. Registration is completed by the person authorized to represent the catering establishment (or the person authorized in the ROA). It consists of the following steps:

- 1) Adding the service provider to the NTDSC Registration module:
 - a) Login requires Client Gate identification;
 - b) Registration of service provider:
 - i) Access the interface: It may be accessed through the menu at Service provider / New service provider
 - ii) During registration, select the form of the economic entity to be registered, this may be Company, Sole Trader, or Other Organization.

- (1) For Companies and Sole Traders, the system checks whether the logged-in user is authorised to act on behalf of the organisation, based on the Authorization Manager Agent (JKÜ) service.
 - (2) For other entities, this authentication will take place after registration has been completed, based on the documents uploaded by the person completing the registration.
 - iii) Once authorization has been successfully checked, add the address of the catering establishment and the contact details of the contact person. After this, press the Save button to complete the registration. (Verification for other organisations takes place at this time, the results of which will be communicated to the user initiating the registration in an e-mail.)
 - c) Adding a service location and indicating the NTDSC VENDÉGLÁTÁS system.
- 2) Switching to NTDSC VENDÉGLÁTÁS system.

3.3. Registration of the catering establishment

After registering the service provider and the service location, the catering establishment needs to be registered, too. A catering establishment is an independent catering unit of the service location. A service location may have more than one catering establishment. You may start the registration of a catering establishment on the page under Service locations, by clicking the Add new catering establishment button. After registering a catering establishment, the user will be redirected to the Service locations page where the newly added catering establishment will be displayed. There is only one remaining requirement for data transmissions: obtaining a certificate necessary for data transmission from the NTDSC VENDÉGLÁTÁS system. The following chapter will describe the relevant steps.

3.4. Issuing a certificate and associating a catering establishment with the RMS system

RMS systems used by catering establishments add an electronic signature to the data submitted during the data provision. To generate an authentic electronic signature, a certificate is required that is provided by a certificate authority (CA).

The manufacturer/service provider of the RMS software is responsible for supporting the catering establishments in generating the certificate signing request (CSR) necessary for issuing the key pair and the certificate.

- a. RMS system manufacturers will receive guidance on how to create certificate signing requests in chapter 6.1. of this document.
- b. The certificate signing requests will be generated in accordance with the above.

Submitting the certificate signing request – requesting a certificate

Catering establishments may request the certificate through the NTDSC VENDÉGLÁTÁS system and can download the ready certificate from there, too. They can do this on the page they can access under menu item Select software, after the required software has been added, by clicking the button Certificate management in the appropriate line of the list. Certificate signing requests may be uploaded at any time, i.e. the registration of the catering establishment may be closed down even if

the required certificate is not yet available. A catering establishment may (theoretically) use more RMS systems. The steps to apply for a certificate are as follows:

- a. The administrator of the service provider uploads the requests generated by the RMS system (CSR file in the appropriate field) on the certificate requesting page available as described above.
- b. The certificate signing request is submitted by clicking on the Send certificate signing request button. This is how requests are forwarded to the certificate authority (CA).
- c. The request supplemented with the data of the catering establishment recorded in the system will be forwarded by the NTDSC VENDÉGLÁTÁS system to the CA, which saves the extra work that would be necessary for the developers of the RMS system to create a separate request.

Downloading certificates and status check

The user associated with the catering establishment should download the ready certificate from the NTDSC VENDÉGLÁTÁS system and forward it to the providers of the RMS system.

1. The ready certificate will be shown in the table below the upload panel.
2. Generating the certificate will commence immediately, however, depending on system load it may take up to a few minutes – it is not a problem, therefore, if the required certificate does not show immediately.
3. The representative of the catering establishment may check the status of certificate requests and certificates in the table, which may be the following:
 - i. Request in process;
 - ii. Certificate ready;
 - iii. Error in issuing the certificate;
 - iv. Withdrawal in process;
 - v. Withdrawn.
4. The catering establishment must download the ready certificate from the interface of the NTDSC VENDÉGLÁTÁS system.
5. After this, it needs to be imported to the certificate repository of the RMS software runtime environment (together with the private key, if it was generated elsewhere). The RMS manufacturer/provider is responsible for providing support to this step.

In the following cases, a certificate needs to be requested:

- Replacing the RMS system (switching to another manufacturer's software);
- Obtaining a new certificate because of an expired certificate (the certificate is valid for two years from the date it is issued);
- Replacing a certificate due to corrupted or damaged private keys.

3.5. RMS system configuration in a live environment

For the live deployment of the RMS software, the following configuration steps need to be taken:

- Storing the NTDSC VENDÉGLÁTÁS system identifier of the RMS software detailed above;
- Storing service provider's tax number;

- Storing the registration number of the catering establishment;
- Storing the NTDSC certificate.

3.6. Commencement of live data transmission

After implementing the above configuration changes, RMS software is ready for live data submissions.

4. Types of RMS messages, messaging process

4.1. Summary

The NTDSC VENDÉGLÁTÁS system's RMS interface is a collection of endpoints serving RMS data communication. The data are received at the individual data submission endpoints, while the processing status of the data is queried at a common and major verification endpoint.

The NTDSC VENDÉGLÁTÁS system's RMS interface distinguishes between three endpoints for communication with RMS software:

- Order summary endpoint;
- Daily closure endpoint
- Verification endpoint.

Receiving endpoints: daily closure and order summary

The NTDSC VENDÉGLÁTÁS system saves the messages received at the receiving endpoints in a temporary storage and sends a synchronous response of the result of the reception to the RMS software. In this synchronous response, if the reception is successful, the application sends 200 OK responses and one submission identifier to the calling party. In the case of an incorrect request, it returns an error key and an error message, which are described in chapter 5.2. Receiving a message does not mean that the message has been successfully processed.

Verification endpoint

The sender systems may query the processing status of the messages submitted with the help of the verification endpoint. If an error is discovered during the processing of the message, the call systems receive detailed feedback also through these endpoints.

In addition to the technical details mentioned above, the processing identifiers received after submitting the order summaries or the daily closure messages must be submitted to the verification endpoint.

The RMS interface will send a detailed response to every processing identifier sent to the verification endpoint.

Messages can have the following statuses:

- Received
 - this means that the message has been received by the NTDSC system but it is still being processed.
- Completely incorrect
 - in this case the message did not contain any item that the NTDSC system could have processed successfully. The NTDSC system will send a detailed list of errors for unsuccessfully processed items.

- Partially processed
 - in this case the message contains a mix of successfully and unsuccessfully processed items. The NTDSC system will send a detailed list of errors for unsuccessfully processed items.
- Successfully processed
 - in this case every item in the message has been processed successfully.
- Resend
 - in this case the NTDSC system requests the sending system to resend the given message without changes.

4.2. Order of data submissions

Since the data to be submitted to NTDSC VENDÉGLÁTÁS system are interdependent, it is necessary to apply the following order when submitting the data. Any deviance from the order will mean that the processing system will park the messages it received based on their dependencies until these dependencies have been resolved. If the dependencies are not resolved within a short time (the message the system is waiting for does not arrive), parked messages will be marked as incorrect submissions and, therefore, they will need to be resent. Order of processing of each message type:

1. Order summary;
2. Corrective order summary;
3. Cancelled order summary;
4. Daily closure.

4.3. Submitting an order summary

To manage the industry with a data-driven approach, the NTDSC system needs to receive as detailed and as accurate data about the turnover of catering establishments as possible. Another important objective is that data provision should not inhibit or slow the serving of the guests.

In the order summaries, it is necessary to provide general data about the order and the items consumed in addition to the technical details.

Technical details:

- service provider's tax number;
- NTDSC registration number of the catering establishment;
- NTDSC identifier of the RMS system;
- RMS system version number;
- date and time of data transmission;

General data about the order:

- type of order summary (normal / cancellation / correction);

- RMS order summary identifier (unique local identification number)
- for cancellation and correction, the identifier of the referenced RMS order summary;
- order start (first item) and finish (closure of payment);
- payment method(s) and sums.

Data related to the items ordered:

- name;
- main category (by a standard value set);
- subcategory (by a standard value set);
- VAT category;
- gross unit price;
- unit of measurement;
- amount;
- quantity;
- date and time when the item was ordered;
- gross total sum of the item.

Order details recorded since the previous data submission need to be forwarded to the data submission endpoint of the NTDSC VENDÉGLÁTÁS system associated with order summaries. If no orders have been made since the previous data provision, it is not necessary to send a message to the endpoint. An order summary message may contain one or more order summaries, and particular attention needs to be paid to the fact that one message does not contain more than 500 order items to ensure successful transmission. 500 order items may be the result of more than one order summary.

A valid order summary that has been submitted with inaccurate data may be corrected or cancelled. In these cases, enter the identifier of the order summary that needs to be corrected or cancelled in the “hivatkozottRendelesOsszesito” field, and set the “rendelesOsszesitoBesorolasa” field to “sztorno” or “helyesbito”.

A corrective order summary completely replaces the corrected summary, and therefore it must contain all the accurate data that were sent earlier, it is not enough to only send the corrected items.

A corrective order summary may also be corrected or cancelled: In this case, the identifier of the last corrective order summary must be provided as a reference. One order summary may be modified (corrected or cancelled) only once. A cancelled order summary may not be corrected, cancellations may not be withdrawn, in this case the data must be submitted in a new order summary.

When developing integration for the RMS interface, the developer of the RMS system must prepare the application that the value sets of the fields containing ENUMs defined in the interface may change in the future due to adaptation to the changes of market environment.

4.4. Submitting daily closure

The purpose of the daily closure message is to perform a controlling function for both the data providing catering establishments and the NTDSC system. On the other hand, it provides an opportunity for the catering establishment concerned to report days with no sales or days when it was closed, or to submit data via this endpoint that it cannot submit in the order summaries.

Daily closure data are comprised of two main blocks:

- technical details;
- summary details.

Technical details:

- Service provider's tax number;
- NTDSC registration number of the catering establishment;
- NTDSC identifier of the RMS system;
- RMS system version number;
- date and time of data transmission.

Summary details:

- transaction date;
- classification of transaction date (normal / closed on the day concerned / day with no sales);
- time of closing and opening;
- total tips.

Every RMS software must send a daily closure message even if no order summary was sent on the transaction date concerned. In this case, it must be indicated in the daily closure message whether the catering establishment was closed on the given day or had a day with no sales. Multiple daily closure messages can be sent for one transaction date if a current day is open, the midday closure function can be treated in this way, too.

4.5. Verification of submitted data

For each of the messages sent to the above two data communication endpoints, the calling party receives a unique identifier from the NTDSC Vendéglátás system in a synchronous response. This identifier will be used later to retrieve the result of the processing of the data sent in that particular message.

Multiple unique identifiers can be sent to the verification endpoint.

In the response, the result of the data processing is reported broken down by unique identifiers, and within that into successfully and unsuccessfully processed items. In case of unsuccessful items, the system sends an error key and a description.

If the sending party receives a response from the verification endpoint that contains items that have not been processed successfully, the items must be examined, the error corrected and the data resubmitted.

Otherwise, the processing of items sent in subsequent data transmissions based on the failed items will also fail. The sending system is responsible for correcting and resubmitting incorrect items.

4.6. Specification description of messages

4.6.1. Structure of message header

All the messages that can be submitted on the interface uniformly have a field group that contains the technical details of the Service Provider, the RMS system and the message.

Field name		JSON name
Service provider data		szolgaltatoAdatok
	Service provider's tax number	adoszam
	Catering establishment registration number	vendeglatoUzletRegSzam
Message details		uzenetAdatok
	Message send time	uzenetKuldesIdeje
Sending system details		kuldoRendszerAdatok
	RMS system's NTDS system identifier	rmsRendszerNTAKazonosito
	RMS system version number	rmsRendszerVerzioSzam

4.6.1.1. Service provider data

Type	object
JSON name	szolgaltatoAdatok
Size	-
Mandatory	yes
Value set	-
Comment	-

4.6.1.1.1. Service provider tax number

Type	String
JSON name	adoszam
Size	11
Mandatory	yes
Value set	-
Comment	Tax number of entity submitting the data, without dashes

Error messages

Validation	Type	Error key
The service provider with the tax number indicated has no VENDÉGLÁTÁS interests	Asynchronous	MismatchSzolgaltatoAdatokAdoszam
Mandatory	Synchronous	NotNull
Mandatory	Synchronous	NotEmpty
Length = 11	Synchronous	Size
Must be identical in cert and in the field	Synchronous	MismatchCertificateCnCheck

4.6.1.1.2. Catering establishment registration number

Type	String
-------------	--------

JSON name	vendeglatoUzletRegSzam
Size	10
Mandatory	yes
Value set	-
Comment	The catering establishment registration number contains the service location identifier, too, and it can be compared with the service location identifier included in the certificate.

Error messages

Validation	Type	Error key
The service location can be found but the catering establishment identifier cannot be found	Asynchronous	NotFoundInDbVUzlet
There is no such service location at all	Asynchronous	NotFoundInDbSzolgaltatasiHely
The service provider does but the given service location does not have a VENDÉGLÁTÁS interest	Asynchronous	MissingIdSzolgHelyAzonositoSzamVendeglatasban
Mandatory	Synchronous	NotNull
Mandatory	Synchronous	NotEmpty
Length = 10	Synchronous	Size
Must be identical in cert and in the field	Synchronous	MissmatchCertificateCnCheck

4.6.1.2. Message details

Type	object
JSON name	uzenetAdatok
Size4	-
Mandatory	yes
Value set	-
Comment	-

4.6.1.2.1. Message send time

Type	timestamp (RFC 3339 full-time, pl 2022-01-07T19:20:30.45+01:00)
JSON name	uzenetKuldesIdeje
Size	-
Mandatory	yes
Value set	-
Comment	The time of sending the data provision message. Specified by the sender.

Error messages

Validation	Type	Error key
Mandatory	Synchronous	NotNull
Timestamp	Synchronous	MissmatchType
<= sysDate + 2h	Synchronous	Future
>= sysDate - 7d	Synchronous	Past

4.6.1.3. Sending system details

Type	object
JSON name	kuldoRendszerAdatok
Size	-
Mandatory	yes
Value set	-
Comment	-

4.6.1.3.1. RMS system's NTDSC system identifier

Type	String
JSON name	rmsRendszerNTAKazonosito
Size	20
Mandatory	yes
Value set	-
Comment	RMS identifier recorded in the NTDSC system by HTA

Error messages

Validation	Type	Error key
RMS software must be among the validated systems	Asynchronous	MissingRMSSystemInVendeglatas
No RMS software is registered for the service location provided	Asynchronous	ErrorKuldorendszer
Mandatory	Synchronous	NotNull
Mandatory	Synchronous	NotEmpty
Length <= 20	Synchronous	Size

4.6.1.3.2. RMS system version number

Type	String
JSON name	rmsRendszerVerzioszam
Size	20
Mandatory	yes
Value set	-
Comment	Version number of the given RMS system

Error messages

Validation	Type	Error key
Mandatory	Synchronous	NotNull
Mandatory	Synchronous	NotEmpty
Length <= 20	Synchronous	Size

4.6.2. Structure of order summary message

Field name	JSON name
Header (see "Structure of message header")	
Order summaries	rendelesOsszesitok
Order summary classification	rendelesBesorolasa
RMS Order identifier	rmsRendelesAzonosito
Order summary identifier referenced by RMS	hivatkozottRendelesOsszesito

	Transaction date	targynap
	Order start	rendelesKezdet
	Order end	rendelesVege
	Is it consumed on site?	helybenFogyasztott
	Combined data provision	osszesített
	Reason for combined data provision	osszesítettIndoklása
	Payment Information	fizetesiInformaciok
	Total sum of order in HUF	rendelesVegosszegeHUF
	Means of payment	fizetesiModok
	Payment method	fizetesiMod
	Sum paid in HUF	fizetettOsszegHUF
	Order items	rendelesTetelek
	Item name	megnevezes
	Main category	fokategoria
	Subcategory	alkategoria
	VAT category	afaKategoria
	Gross unit price	bruttoEgysegar
	Unit of measurement	mennyisegiEgyseg
	Amount	mennyiseg
	Quantity	tetelszam
	Date and time of the order	rendelesIdopontja
	Item aggregator	tetelOsszesito

4.6.2.1. Order summaries

Type	array
JSON name	rendelesOsszesitok
Size	1..n
Mandatory	yes
Value set	-
May be repeated	At least one block, but it may be repeated.
Comment	-

4.6.2.1.1. Order summary classification

Type	enum
JSON name	rendelesBesorolasa
Size	-
Mandatory	yes
Value set	NORMAL SZTORNO HELYESBITO
Comment	-

Error messages

Validation	Type	Error key
Mandatory	Synchronous	NotNull
Enum	Synchronous	ErrorEnum

4.6.2.1.2. RMS Order identifier

Type	UUID
JSON name	rmsRendelesAzonosito
Size	36
Mandatory	yes
Value set	-
Comment	Order summary identifier in the RMS system

Error messages

Validation	Type	Error key
Mandatory	Synchronous	NotNull
Uniqueness check	Asynchronous	UniqueConstraint
Uniqueness check	Synchronous	DuplicatedId
uuid	Synchronous	MissmatchType

4.6.2.1.3. Order summary identifier referenced by RMS

Type	UUID
JSON name	hivatkozottRendelesOsszesito
Size	36
Mandatory	conditional
Value set	-
Comment	mandatory if rendelesOsszesitoBesorolasa is not NORMAL

Error messages

Validation	Type	Error key
is mandatory if rendelesBesorolasa is not NORMAL	Synchronous	NotNull
	Asynchronous	NotFoundInDb
There is already a Corrective or Cancellation message for the referenced order summary	Asynchronous	ReferenceConflict
uuid	Synchronous	MissmatchType
The referenced order summary cannot be a Cancellation	Asynchronous	ReferenceConflict

4.6.2.1.4. Transaction date

The transaction date of the order in every case is the transaction date of the current open day at the closing of the order.

Type	date
JSON name	targynap
Size	-
Mandatory	yes
Value set	-
Comment	RFC 3339 standard full-time, e.g. 2021-01-07

	Which day the order is related to The transaction date of the order in every case is (in case of cancellation and correction as well) the date of the current open day at the time the request was sent.
--	--

Error messages

Validation	Type	Error key
Mandatory	Synchronous	NotNull
rfc 3339 full date	Synchronous	MissmatchType
<= sysDate	Synchronous	Future

4.6.2.1.5. Order start

Type	timestamp
JSON name	rendelesKezdete
Size	-
Mandatory	conditional
Value set	-
Comment	First item recorded at RFC 3339 standard full-time, e.g. 2022-01-07T19:20:30.45+01:00

Error messages

Validation	Type	Error key
is mandatory if rendelesBesorolasa is not SZTORNO	Synchronous	NotNull
Timestamp	Synchronous	MissmatchType
<= sysDate	Synchronous	Future
>= targynap start date	Synchronous	Past

4.6.2.1.6. Order end

Type	timestamp
JSON name	rendelesVege
Size	-
Mandatory	conditional
Value set	-
Comment	Order summary closed at RFC 3339 standard full-time, e.g. 2022-01-07T19:20:30.45+01:00

Error messages

Validation	Type	Error key
is mandatory if rendelesBesorolasa is not SZTORNO	Synchronous	NotNull
Timestamp	Synchronous	MissmatchType
>= targynap start date	Synchronous	Past
order end – order start <= 24 hours	Synchronous	Conflict

4.6.2.1.7. Is it consumed on site?

Type	bool
JSON name	helybenFogyasztott
Size	-
Mandatory	conditional
Value set	-
Comment	In mixed cases, select on-site consumption.

Error messages

Validation	Type	Error key
is mandatory if rendelesBesorolasa is not SZTORNO	Synchronous	NotNull
Bool	Synchronous	MismatchType

4.6.2.1.8. Combined data provision

Type	bool
JSON name	osszesített
Size	-
Mandatory	conditional
Value set	-
Comment	It indicates order summaries that contain the combined sales of a longer period (max. 1 business day). It can only be used in case of a service outage, e.g. power outage or system downtime. It is to be sent with false data in the case of normal data provision

Error messages

Validation	Type	Error key
is mandatory if rendelesBesorolasa is not SZTORNO	Synchronous	NotNull
Bool	Synchronous	MismatchType

4.6.2.1.9. Reason for combined data provision

Type	enum
JSON name	osszesítettIndoklása
Size	-
Mandatory	conditional
Value set	UZEMSZUNET_ARAMSZOLGALTATAS_TERVEZETT_KIMARADASA_MIATT UZEMSZUNET_INTERNETSZOLGALTATAS_TERVEZETT_KIMARADASA_MIATT UZEMZAVAR_ARAMSZOLGALTATAS_ATMENETI_KIMARADASA_MIATT UZEMZAVAR_INTERNETSZOLGALTATAS_ATMENETI_KIMARADASA_MIATT UZEMZAVAR_TERMESZETI_KATASZTROFA_MIATT UZEMZAVAR_LEJART_TANUSITVANY_MIATT UZEMZAVAR_MUSZAKI_HIBA_MIATT
Comment	Mandatory if Osszesített = true

Error messages

Validation	Type	Error key
------------	------	-----------

mandatory if Osszesített = true	Synchronous	NotNull
Enum	Synchronous	ErrorEnum

4.6.2.1.10. Payment Information

Type	object
JSON name	fizetesiInformaciok
Size	-
Mandatory	conditional
Value set	-
Comment	-

4.6.2.1.10.1. Total sum of order in HUF

Type	int
JSON name	rendelesVegosszegeHUF
Size	-
Mandatory	conditional
Value set	-
Comment	-

Error messages

Validation	Type	Error key
is mandatory if rendelesBesorolasa is not SZTORNO	Synchronous	NotNull
integer	Synchronous	MissmatchType
Σ(order items: order summary)	Synchronous	Conflict

4.6.2.1.10.2. Means of payment

Type	array
JSON name	fizetesiModok
Size	1..n
Mandatory	conditional
Value set	-
May be repeated	At least one block, but it may be repeated.
Comment	-

4.6.2.1.10.2.1. Payment method

Type	enum
JSON name	fizetesiMod
Size	-
Mandatory	conditional
Value set	KESZPENZHUF KESZPENZEUR SZEPKARTYA BANKKARTYA ATUTALAS EGYEB VOUCHER

	SZOBAHITEL KEREKITES
Comment	List only the payment method that were used in the given transaction. Calculation method of rounding value: paid sum - payable sum (i.e., if payable sum is HUF 54, the payment methods: KP:55;KER:-1)

Error messages

Validation	Type	Error key
is mandatory if rendelesBesorolasa is not SZTORNO	Synchronous	NotNull
Enum	Synchronous	ErrorEnum

4.6.2.1.10.2.2. Sum paid in HUF

Type	int
JSON name	fizetettOsszegHUF
Size	-
Mandatory	conditional
Value set	-
Comment	Sum paid using the given payment method in HUF.

Error messages

Validation	Type	Error key
is mandatory if rendelesBesorolasa is not SZTORNO	Synchronous	NotNull
integer	Synchronous	MissmatchType
In case of KESZPENZHUF 0 or 5 ending is mandatory, the value of rounding should appear in the KEREKITES payment method	Synchronous	Conflict
$\sum$ fizetettOsszegHUF = rendelesVegosszegeHUF	Synchronous	Conflict

4.6.2.1.11. Order items

Type	array
JSON name	rendelesTetelek
Size	1..n
Mandatory	conditional
Value set	-
May be repeated	At least one block, but it may be repeated.
Comment	is mandatory if rendelesBesorolasa is not SZTORNO

4.6.2.1.11.1. Item name

Type	String
JSON name	megnevezes
Size	255
Mandatory	conditional
Value set	-
Comment	-

Error messages

Validation	Type	Error key
is mandatory if rendelesBesorolasa is not SZTORNO	Synchronous	NotNull
Mandatory	Synchronous	NotEmpty
Length <= 255	Synchronous	Size

4.6.2.1.11.2. Main category

Type	enum
JSON name	fokategoria
Size	-
Mandatory	conditional
Value set	ETEL ALKMENTESITAL_HELYBEN ALKMENTESITAL_NEM_HELYBEN ALKOHOLOSITAL EGYEB
Comment	The field values correspond to the field values of the alkategoria. Detailed list is included in the annex.

Error messages

Validation	Type	Error key
is mandatory if rendelesBesorolasa is not SZTORNO	Synchronous	NotNull
Enum	Synchronous	ErrorEnum

4.6.2.1.11.3. Subcategory

If a discount is sent, the sum of the discount must be sent with a negative value. General discounts for sum totals must be sent as discounts the sums of which are proportionately divided according to the VAT categories in the order summary. The same must be done in the case of the service fee, too.

Type	enum
JSON name	alkategoria
Size	-
Mandatory	conditional
Value set	REGGELI SZENDVICS ELOETEL LEVES FOETEL KORET SAVANYUSAG_SALATA DESSZERT SNACK EGYEB LIMONADE_SZORP_FACSART ALKOHOLMENTES_KOKTEL TEA_FORROCSOKOLADE KAVE VIZ ROSTOS_UDITO

	SZENSAVAS_UDITO KOKTEL LIKOR PARLAT SOR BOR PEZSGO EGYEB SZERVIZDIJ BORRAVALO KISZALLITASI_DIJ KORNYEZETBARAT_CSOMAGOLAS MUANYAG_CSOMAGOLAS KEDVEZMENY
Comment	The field values correspond to the field values of the fokategoria. Detailed list is included in the annex.

Error messages

Validation	Type	Error key
is mandatory if rendelesBesorolasa is not SZTORNO	Synchronous	NotNull
Enum	Synchronous	ErrorEnum
Value corresponding to the enum list filtered by main category	Asynchronous	Conflict

4.6.2.1.11.4. VAT category

Type	enum
JSON name	afaKategoria
Size	-
Mandatory	conditional
Value set	A_5 B_18 C_27 D_AJT E_0
Comment	-

Error messages

Validation	Type	Error key
is mandatory if rendelesBesorolasa is not SZTORNO	Synchronous	NotNull
Enum	Synchronous	ErrorEnum

4.6.2.1.11.5. Gross unit price

Type	double
JSON name	bruttoEgysegar
Size	-
Mandatory	conditional
Value set	-
Comment	-

Error messages

Validation	Type	Error key
is mandatory if rendelesBesorolasa is not SZTORNO	Synchronous	NotNull
Double	Synchronous	MissmatchType

4.6.2.1.11.6. Unit of measurement

Type	enum
JSON name	mennyisegiEgyseg
Size	-
Mandatory	conditional
Value set	DARAB LITER KILOGRAMM EGYSEG
Comment	Select a quantity corresponding to the type of product. E.g. use LITER in the case of a 0.33 litre soft drink can, not DARAB.

Error messages

Validation	Type	Error key
------------	------	-----------

is mandatory if rendelesBesorolasa is not SZTORNO	Synchronous	NotNull
Enum	Synchronous	ErrorEnum

4.6.2.1.11.7. Amount

Type	double
JSON name	mennyiseg
Size	-
Mandatory	conditional
Value set	-
Comment	Quantity value associated with the unit of measurement of the given product. E.g.: "0.33" for a soft drink can.

Error messages

Validation	Type	Error key
is mandatory if rendelesBesorolasa is not SZTORNO	Synchronous	NotNull
Double	Synchronous	MissmatchType
> 0	Synchronous	Negative
<> 0	Synchronous	Zero

4.6.2.1.11.8. Quantity

Type	double
JSON name	tetelszam
Size	-
Mandatory	conditional
Value set	-
Comment	The quantity ordered from the given product. In the case of selling 2 cans of soft drink: "2"

Error messages

Validation	Type	Error key
is mandatory if rendelesBesorolasa is not SZTORNO	Synchronous	NotNull
Double	Synchronous	MissmatchType
> 0	Synchronous	Negative
<> 0	Synchronous	Zero

4.6.2.1.11.9. Date and time of the order

Type	timestamp
JSON name	rendelesIdopontja
Size	-
Mandatory	conditional
Value set	-
Comment	RFC 3339 standard full-time, e.g. 2022-01-07T19:20:30.45+01:00

Error messages

Validation	Type	Error key
is mandatory if rendelesBesorolasa is not SZTORNO	Synchronous	NotNull

Timestamp	Synchronous	MissmatchType
<= sysDate	Synchronous	Future
>= order start	Synchronous	Past
<= order end	Synchronous	Future

4.6.2.1.11.10. Item aggregator

Type	int
JSON name	tetelOsszesito
Size	-
Mandatory	conditional
Value set	-
Comment	the product of amount and gross unit price rounded to an integer (rounding by merchant)

Error messages

Validation	Type	Error key
is mandatory if rendelesBesorolasa is not SZTORNO	Synchronous	NotNull
integer	Synchronous	MissmatchType
= round(bruttoEgysegar * tetelszam)	Synchronous	Conflict

4.6.3. Daily closure message structure

Field name	JSON name
Header (see "Structure of message header")	
Daily closure data	zarasiInformaciok
Transaction date	targynap
Data provision classification of the transaction date	targynapBesorolasa
Date and time of opening	nyitasIdopontja
Date and time of closing	zarasIdopontja
Total tips	osszesBorravalo

4.6.3.1. Daily closure data

Type	object
JSON name	zarasiInformaciok
Size	-
Mandatory	yes
Value set	-
Comment	-

4.6.3.1.1. Transaction date

Type	Date
JSON name	targynap
Size	-
Mandatory	yes
Value set	-

Comment	Which day's closing is included in the message RFC 3339 standard full-time, e.g. 2021-01-07
----------------	--

Error messages

Validation	Type	Error key
Mandatory	Synchronous	NotNull
rfc 3339 full date	Synchronous	MismatchType
<= sysDate	Synchronous	Future

4.6.3.1.2. Data provision classification of the transaction date

Type	enum
JSON name	targynapBesorolasa
Size	-
Mandatory	yes
Value set	ADOTT_NAPON_ZARVA FORGALOM_NELKULI_NAP NORMAL_NAP
Comment	-

Error messages

Validation	Type	Error key
Mandatory	Synchronous	NotNull
Enum	Synchronous	ErrorEnum

4.6.3.1.3. Date and time of opening

Type	timestamp
JSON name	nyitasIdopontja
Size	-
Mandatory	conditional
Value set	-
Comment	is mandatory if the classification of the transaction date is not: ADOTT_NAPON_ZARVA RFC 3339 standard full-time, e.g. 2022-01-07T19:20:30.45+01:00

Error messages

Validation	Type	Error key
is mandatory if the classification of the transaction date is not: ADOTT_NAPON_ZARVA	Synchronous	NotNull
The time of opening cannot be later than the transaction date	Synchronous	Future
Timestamp	Synchronous	MismatchType
<= sysDate	Synchronous	Future
With earlier daily closure cannot be overlapping period -> nyitasIdopontja >= earlier zarasIdopontja	Asynchronous	Past

4.6.3.1.4. Date and time of closing

Type	timestamp
JSON name	zarasIdopontja

Size	-
Mandatory	conditional
Value set	-
Comment	is mandatory if the classification of the transaction date is not: ADOTT_NAPON_ZARVA RFC 3339 standard full-time, e.g. 2022-01-07T19:20:30.45+01:00

Error messages

Validation	Type	Error key
is mandatory if the classification of the transaction date is not: ADOTT_NAPON_ZARVA	Synchronous	NotNull
The time of closing cannot be earlier than the transaction date	Synchronous	Past
The time of closing cannot be earlier than the opening	Synchronous	Past
Timestamp	Synchronous	MismatchType
<= sysDate	Synchronous	Future

4.6.3.1.5. Total tips

Type	int
JSON name	osszesBorravalo
Size	-
Mandatory	conditional
Value set	-
Comment	is mandatory if the classification of the transaction date is NORMAL NAP

Error messages

Validation	Type	Error key
> 0	Synchronous	Negative
Integer	Synchronous	MismatchType

4.6.4. Verification endpoint message structure

Field name	JSON name
Header (see "Structure of message header")	
Processing identifiers	feldolgozasAzonositok
NTDSC system unique processing identifier	feldolgozasAzonosito

4.6.4.1. Processing identifiers

Type	array
JSON name	feldolgozasAzonositok
Size	1..n
Mandatory	yes
Value set	-
May be repeated	At least one block, but it may be repeated.
Comment	-

4.6.4.1.1. NTDS system unique processing identifier

Type	UUID
JSON name	feldolgozasAzonosito
Size	36
Mandatory	yes
Value set	-
Comment	-

Error messages

Validation	Type	Error key
	Synchronous	NotFoundInDb
Mandatory	Synchronous	NotNull
uuid	Synchronous	MismatchType

4.6.5. Response messages

4.6.5.1. In the case of acceptable requests

4.6.5.1.1. In the case of Order summary or Daily closure message

If the data in the submitted Order summary or the Daily closure message are correct based on preliminary checks and the system can accept the message, then a processing identifier is sent in the synchronous response with which the status of the message received can be queried at the verification endpoint.

Field name	JSON name	Type
NTDSC system unique processing identifier	feldolgozasAzonosito	UUID

4.6.5.1.2. In the case of a verification endpoint message

In the case of a successful request message to the verification endpoint, a message is sent with the following structure. The response message contains in every case the Processing identifiers queried and the related Status.

If the message belonging to the Processing identifier has not been processed yet, the Status field value will be „RECEIVED” and the other blocks containing successful/unsuccessful items will not appear.

If the message belonging to the Processing identifier has been processed, its content will be listed by item (order summaries or daily closure) in blocks of successful and unsuccessful items. In the case of successful items the error that occurred will also be detailed. The status field values can be the following: BEFOGADVA, TELJESEN_HIBAS, RESZBEN_SIKERES, TELJESEN_SIKERES, UJRA_KULDENDO,

If the message belonging to the Processing identifier cannot be processed due to an error of the message header, it will be detailed in the Header errors block. In such cases, the blocks containing successful/unsuccessful messages will not appear.

Field name				JSON name
NTDSC system unique processing identifiers				uzenetValaszok
	NTDSC system unique processing identifier			feldolgozasAzonosito
	Header errors			fejlecHibak
		Error		
			Field name	mezoNeve
			Value sent	kuldottErtek
			Error key	hibaKulcs
			Error message	hibaUzenet
	Items processed successfully			
		Successful items		sikerUzenetek
			Type	tipus
			Identifier received from RMS	rmsAzonosito
	Items processed unsuccessfully			
		Unsuccessful item		sikertelenUzenetek
			Type	tipus
			Identifier received from RMS	rmsAzonosito
			List of errors	uzenetHibak
			Error	
			Field name	mezoNeve
			Value sent	kuldottErtek
			Error key	hibaKulcs
			Error message	hibaUzenet
	Status			statusz

4.6.5.2. In the case of unacceptable requests

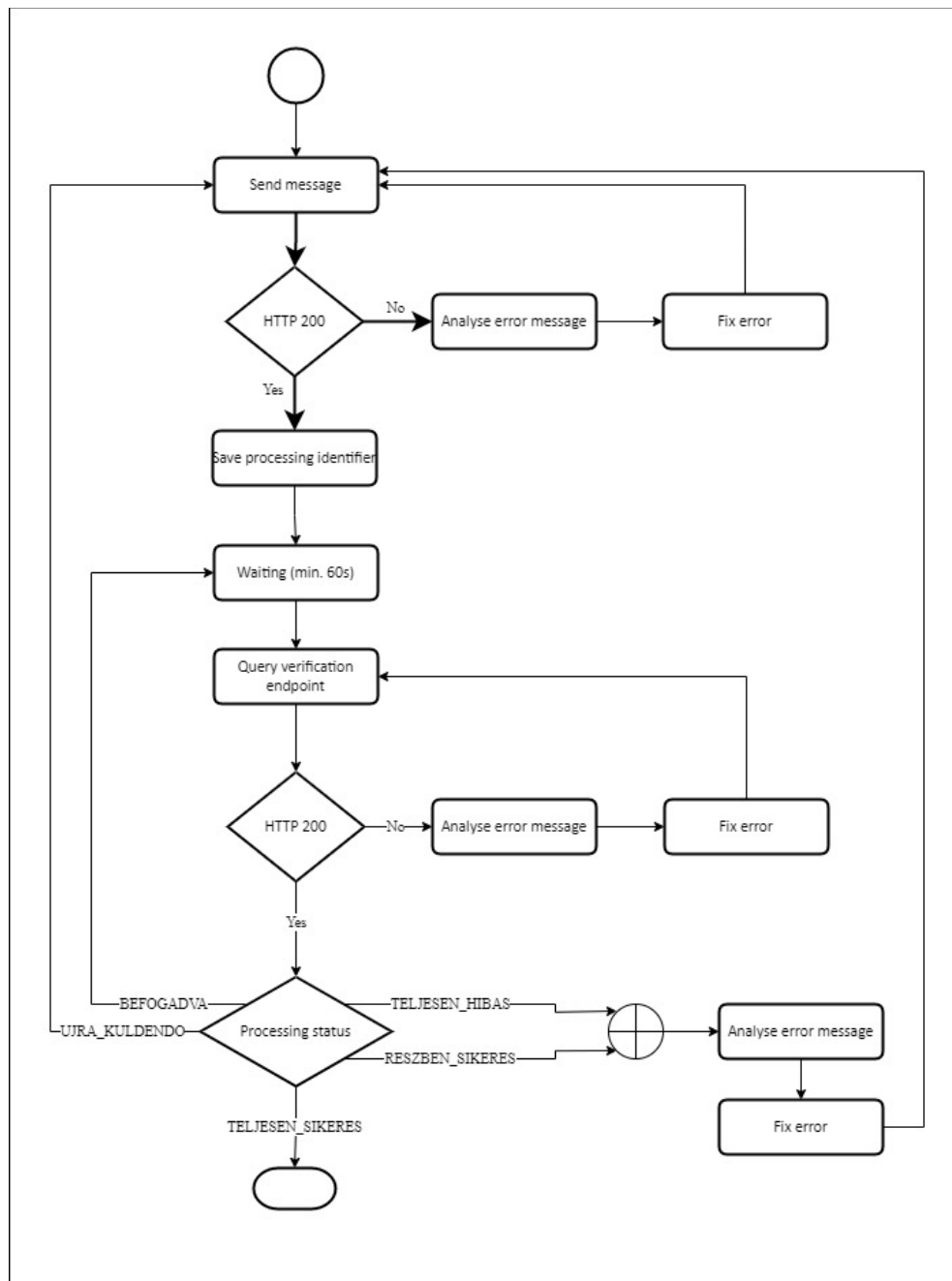
If the message submitted, based on the information it contains, is incorrect and the system cannot accept it, the location and reason of the error will be sent in a synchronous response in the following message structure.

Field name		JSON name	Type
List of errors		uzenetHibak	array
	Field name	mezoNeve	string
	Value sent	kuldottErtek	string
	Error key	hibaKulcs	string
	Error message	hibaUzenet	string

5. Process description, troubleshooting and formal requirements

This chapter details the formal requirements, technical information and error keys related to messaging.

5.1. Data provision process



The order summary, corrective order summary, cancelled order summary and daily closure data must be sent via HTTP REST call endpoints to the endpoint of the data reception interface of the NTDSC VENDEGLÁTÁS system.

To ensure the integrity of data submissions, the submitted data must be signed with a digital signature.

The NTDSC VENDEGLÁTÁS system will first examine the certificate, then the digital signature of the submitted data. If there is any problem with these, it will send a synchronous error response and will not commence processing the submitted data.

If both the certificate and the digital signature are in order, the system will conduct a syntactic test of the data sent in the HTTP REST body. It will examine whether the query contains every mandatory data.

If the submission has successfully passed the syntactic test of the NTDSC VENDEGLÁTÁS system, the system will return a processing identifier to the submitted request. This means that the NTDSC VENDEGLÁTÁS system has accepted the request for processing, in such cases the status of the processing will be BEFOGADVA.

The sending system can use the processing identifier to query the status of the submitted data. The result of the processing of the submitted data can be queried at the verification endpoint (/api/ellenorzes/statusz). The sending systems can use these verification endpoints to query the status of the data successfully processed and / or unsuccessfully processed. The status of more than one submission (processing identifier) can be queried through the verification endpoint at one time.

If the data submission is unsuccessful, the list of errors will contain the values to be corrected. In such cases, the errors indicated in the list of errors must be corrected and the data the content of which has been corrected must be resent to the NTDSC VENDEGLÁTÁS system. Only unsuccessfully processed order summaries must be resent by the sending system.

After being accepted, the queries (the submitted query will receive a BEFOGADVA status at the verification endpoint) will be forwarded to the NTDSC VENDEGLÁTÁS system for further processing. Here, the submitted certificate and the digital signature will be checked again by the system. Both the certificate and the digital signature must be sent as a key-value pair in the HTTP Header of the REST query. The certificate is the x-certificate key, and its value is the BASE64-encoded certificate. The x-jws-signature is the key, and its value will be the generated key.

Example:

x-certificate :

```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSUhEVENDQXZXZ0F3SUJBZ0lDRUFJd0RRWUplb1pjaHJkFRRUxCUUF3Z1k4e
ER6QU5CZ05WQkFNTUJrRIQKURFeU16RU5NQXNHQTFVRUNBd0VVR1Z6ZERFTE1Ba0dBMVVFQmhmNQ1NIVXhJekFoQmdrcW
hraUc5dzBCQ1FFVwpGR2x1Wm05QWFXNTBaWEp0WldScFIYUmxMbWgxTVNFd0h3WURWUUVFLREJoSmJuUmxjbTFsWkdsAG
RHVWdTMkZzCmJRnVJRXRtZEM0eEdEQVdCZ05WQkFzTUQwbFVJR2x1ZEdWeWJXVmtHbV0YwWIRBZU3MHINREExTWpBeE5
EUTIKTVRoYUZ3MHINVEExTWpBeE5EUTJNVGhhTURjeEVUQVBCZ05WQkFNTUNFaElVa3BSTVV0Q01RMHdDd1IEVIFRSQpEQVJ
RWlhOME1STXdFUVIEVFRS0RBCEXZV3h0WVc0Z1QzSm5NSUICSWpBTKJna3Foa2lHOXcwQkFRRU9DCkFROEFNSUICQ2dL
Q0FRRUF5QlZtQIN4akcwWjdCakNHR2wzMjQ5aEV5Nnloa0lxMUdK1214MIQ4Wnh4NmZwbGQKTmt5WVhVR0hsZklKV2p3Ykly
V3NxNnVUvWEpLM2p1SG5ITytcQjFkZThYRIJNSWwyMkJSnVOWE9RRTU4dUNpagovZHhHc3k4WlpNM2hHMWxCSCtva1A0NH
M5Q095R0F6WmVUaXgrNVhUcHc2MjB0U0VJODI2TWwvNINhSnBVcHRUCK5XclNJYVEwSXN4M2Y2MWhBRzI6OVFybkF4dEJla
```

nZrY2FVVW9RL0w0Y0dsaEh5QzF1c0ZvVS8xWUdIMUVUSzQKempNN0Iza013ekhweFQ2ZDZGaitUzV5MIVaQjkhR0ZGTjZMS25
hV2ZjNfN3YwTrUDxSW5yTDhxREt4b1JUVAp0ajc1T2hBSzLzOHY5eUlqMctxMkhBeGdReFF3TXJvMFZQR21od0IEQVFBQm80S
EpNSUHTUF3R0ExVWRfD0VCCI93UUNNQUF3SFFZRFZSME9CQJlFRlBqS3BQc1FhQWM4VUNiam5IRzITeXRPsfDROU1COEdB
MVVksXdRWU1CYUEKRkJFQ3M5RGN4ZVZRFJMBfNuOWFMc3FJMTFheE1Bc0dBMMVVRHdRRUF3SUY0REFUQmdOVkhTVUUV
EREFLQmdncgpCZ0VGQJlFREFUQXICZ05WSFI4RUt6QXBNQ2VnSmFBamhpRm9kSFJ3T2k4dmJHOWpZV3hvYjNOMEwybHVkR1
Z5CmJXVmtHv0YwWIM1amNtd3dJQVIEVlWUkJCa3dGNElWYTJGc2JXRnVhVzUwWlhKdFpXUnBZWFJsTG1oMU1BMEckQ1NxR
1NJYjNEUUVCCQ3dVQUE0SUVBUUFwaVFERmIUZHb5b2QxRDIYNXIHOFFrMWZBSEJvZnBps2RhdxD6d0ltdQpJQ0hzMkpDWFirc
HAvKzYzYWdwazBvSIUvOXcvMk5Nb2hIYjBibnVxcjV6QTbtVEU3d3haNmthTzIzbDBQcStMcNfJSORETHZQVGL6K0xkOE5FMWp1
WFFrQUxFTQnBmdkxNUE15bmV2S3RwaFBQT0x0WGUZuNXFLb1lmakViV9K2kKNFFPTUpaVmNTNkl2eVI2Q1dzWGE2RWdWd
mkxRS9vc0VVRHBRUFluZjIKdE5CMnE5NII1ZFNYYzMA4Y292cklvVgpxVlU1cWFpYk50Q1IveUZWBfFYVzc5YUNwVTRtTTd4VG5KR
0I1UGpJUEhIZUMxMUNsWEXJVTJKR21RVE8waHZICm1OQmtuWHRpT0NLeTFBbnZnQXA2Q05BNXIKK2p5N2d2OEpXN1Z4N1F
2VFB1SG1IV0NQTVBjBss0Q2hKd1JaN20KVFNOM2QyOUI4VDJOMjFpRFJqNHNhZUtMSGdRdjVXT2p3UTZEbndBLzRhBTI1VVEv
U0tDa2p2MHEyZkhCY2IJYgpEMTZydxFHcEZValVINEM2MmRCUzh3dIviS3RTdVJxaVdaUTgzeXJONIFBVTdMSDZ4VE1HN2diTnZ
OTmlzSkR4ClRwbDIGS2BCN1pLaHJsRXcxR1dtRk9LaG5yOXRF50kyREo4Z2V1eWc1WFnWmMlLcINrOWV5OVBFaDjuZDFVUOK
dTU3bW1CdXdBVjJXbXU3U0IQNTVQalZEYzY3WDIRNk5Qb09tZmdoRDJ4UzRhZ3piQ2ZBbTVSZFo4UENuYjIFNgp5SUNkSGNHYjg
1N2FDU3dWK0x4VEdJLzcyQkRtL0ZHRFljU2xnQnpOdGc4aUVnWWVXYXyV3RyWURsblRkV25LCm1oVlUwMTZ5dGJ0NjhNNkh
udUFFS2RPN3M3L09KWGPcB09SK0ppQ2Z6YUtwb1FlaERldW5KdkNaNEhvQy9BY1cKS0s0VldmTHFneURnMGxZd0FYZXNTQU9
nNnl1aVdYbmVmdHZDTUpnbnmVrVJVJTUN1cjM0c1dmWjc1RGJteSs0MQo1T2lyVzYrb1pBenlpZERzSIIMZi85R0dvcFiyWDFYelH
wYVZYFXdYaDJ5WINvV3BuUmR5bnV0L2FxTVJxQnRlCnp1Q3YxNEJJSxcvc3QyWSt6MUy0K05iam5RRnZhaGN0Y1U1WFYrTG9R
c0NGbytvRVNRUDcwmFIKkZgzQ1ZyNzUkBoXkSnBVZiszaVpoNXdFSIzB0tLYVpUS2VGRGxuNFJMBnM1SWVHYvDRVmxzUXZuM
GQcxPlUVAycThqbnMzbwPdmZqcGpRaFNQRVRNBxumOWJYcURSRWZYUHFWS01XRC9Zr2lpUHpiREJJaIM3VUtEaHp5clBsR
TVtYjNZVzNjCm54ZnByS3c4Y3BPRDFkbkxGZUilyUysyGVSeFQ4aGlwelfWOVlyZWJUK2NmNnlyWTZJRzFYNXE5TDROUmXqUEg
KcGRvVzFhcjJdVTZIK3kwQkVqblR5WE9ueXNSTXVXS1pDTTE0WIBSKzV3RudZdnEvL0RKOFk0Q2haNXVdc1JQNApnL2k3RvDee
mo3REhCVzJzSnhVYmZOWms1R2xGVXUxNUpCOE5vRDU4eUyFVpenpIOHFnQUNIMzNtaVhUNGczCnllRGdWRFfjTXBwaHlZT
mcvU3U4bEjJVWWhQzISVkcwR2xDZEI3Z3dadZl5ak5JVklDRXR4NVRtOWIWSTRkMlgKQ3I2aTdpQjRZSjRDRnpzeEJSOEdUdXp
GekRCNTN3NVNsa1A3c3praWlwMVQKLS0tLS1FTkQgQ0VSVEIGSUNBVEUtLS0tLQo=

x-jws-signature :

eyJhbGciOiJIUzI1NiJ9..c0jgbAraamEer3lv6qkgjT8Yrov3snbc_uSOTI7gMI_sbRH_fgil-
4daGKf8j_LKW70xprdg89I4MEwtAZRfBMwWC2TDlabOwBPJM9rCSQTjijy0E7TulscUxjrcnKzqzD-
y6a77i_EfytqYyhHIT2b5x7xfG0p1MS6vg8HO6cLq5GUaYcvbBVZGnYh_ov0zXDQVJXOFqUMxhQGUx3gckh9eIYdQtDM_fmRhaXi
zmR1z9uCrReO5LJJ0x8JZ5cJ9YLBKs0Nk2R0-
0tZjmm2JpuS3PJpGz0ZW_Qc1zYMBL7tyPyeiSbwj5zr8rXC1Stg0hYjwCcoXYt1g00JMBMw

It is very important that the x-jws-signature and the x-certificate sections in the HTTP Header of every REST query directed to NTDSC VENDEGLÁTÁS system are filled out correctly!

The digital signature will need to be the key generated by JOSE JWS. The certificate consists of 3 sections: (the values from the above example in the brackets)

header	(eyJhbGciOiJIUzI1NiJ9),	payload	(empty)	and
signature	(c0jgbAraamEer3lv6qkgjT8Yrov3snbc_uSOTI7gMI_sbRH_fgil- 4daGKf8j_LKW70xprdg89I4MEwtAZRfBMwWC2TDlabOwBPJM9rCSQTjijy0E7TulscUxjrcnKzqzD- y6a77i_EfytqYyhHIT2b5x7xfG0p1MS6vg8HO6cLq5GUaYcvbBVZGnYh_ov0zXDQVJXOFqUMxhQGUx3gckh9eIYdQtDM_fmRhaXi zmR1z9uCrReO5LJJ0x8JZ5cJ9YLBKs0Nk2R0- 0tZjmm2JpuS3PJpGz0ZW_Qc1zYMBL7tyPyeiSbwj5zr8rXC1Stg0hYjwCcoXYt1g00JMBMw) .			

The three sections, separated by a dot, will form the digital signature, the JWS token. The value of the x-jws-signature sent to the NTDSC VENDEGLÁTÁS system will only need to contain the header and signature sections. In the header of the digital signature (JWS header), give the type of the encryption algorithm: **RSA256**. The payload section will be generated by the NTDSC VENDEGLÁTÁS system from the JSON structure sent in the REST body via HTTP, that is why the payload section may be left empty in the submitted signature. After this, it is checked whether the submitted data and the signature sent in the query are valid together or not.

If the signature is not satisfactory, the system will indicate it in the list of errors to the sender.

If everything is in order during the signature check, the NTDSC VENDEGLÁTÁS system will start the semantic test of the data submitted. If the system finds an error, it will indicate them on a list of errors. When the sending system queries the status of the submitted data with the processing identifier, it will see a list of successfully and unsuccessfully processed data. The list of errors of unsuccessfully processed data will indicate the data to be corrected. Unsuccessful messages will need to be corrected and resent by the sending system.

If the request only contains correctly submitted, i.e. processable data, the status of the message will turn to SIKERESEN_FELDOGOZVA. In such cases, the list of unsuccessful messages will be empty.

If the request only contains incorrectly submitted/unprocessable data, the status of the message will turn to TELJESEN_HIBAS. In such cases, the list of successful messages will be empty. In such cases, unsuccessfully processed messages will need to be corrected and resent by the sending system.

If the request contains correctly and incorrectly submitted data, too, the status of the message will turn to RESZBEN_SIKERES. In such cases, neither the list of unsuccessful messages, nor the list of successful messages will be empty. In such cases, all the unsuccessfully processed messages will need to be corrected and resent by the sending system.

If the status query is returned with a UJRA_KULDENDO status, it means that the message could not be processed despite being accepted, and therefore the sending system will need to resend the message without any modification.

5.2. Error keys on the RMS interface

5.2.1. Field value error keys

Error key	Error message (Hungarian)	Error message (English)
NotEmpty	Nem lehet üres értékű a mező!	The field cannot be empty!
NotNull	Nem lehet null értékű a mező!	The field cannot be null!
Size	Mező hossza nem megfelelő!	Field length is incorrect!
Max	Az érték túllépte a maximumot!	Value exceeded maximum!
Negative	Nem lehet negatív a mező értéke!	The value of this field cannot be negative!
Zero	A mező értéke nem lehet nulla!	Field value cannot be zero!
MismatchType	A mező típusa nem megfelelő!	Field type is incorrect!
ErrorEnum	A mező hibás értéket tartalmaz! A mező értékkészlete: <enum>	This field contains an invalid value! The value set for this field is <enum>

UniqueConstraint	A megadott érték már szerepel az adatbázisban!	The specified value is already in the database!
DuplicatedId	A beküldött üzenetben nem lehet egynél többször ugyanaz a rendelés azonosító!	The specified value cannot be present multiple times in a message!
Future	A mező időpontja nem lehet jövőbeli!	Field datetime value cannot be in the future!
Past	A mező időpontja nem lehet múltbeli!	Field datetime value cannot be in the past!
Conflict	A mező értéke ellentmondásban áll a függőségeivel!	The value of the field conflicts with its dependencies!
ReferenceConflict	A hivatkozott rendelésösszesítő nem hivatkozható!	The referenced order summary cannot be referenced!
NotFoundInDb	Nem található az adatbázisban!	Not found in the database!

5.2.2. NTDSC registration error keys

Error key	Error message (Hungarian)	Error message (English)
MissingIdSzolgHelyAzonositoSzamVendeglatasan	Nem található a Szolgáltatási hely azonosító szám az VENDÉGLÁTÁS szakrendszerben!	The identification number of the service location is not found in 'VENDÉGLÁTÁS'!
MismatchSzolgáltatoAdatokAdoszam	Nem található szolgáltató adószám az VENDÉGLÁTÁS szakrendszerben!	The tax number of the service provider is not found in 'VENDÉGLÁTÁS'!
MissingRMSSystemInVendeglatas	Nem található ilyen RMS rendszer azonosító a VENDÉGLÁTÁS-ban!	No such RMS system ID found in 'VENDÉGLÁTÁS'!
ErrorKuldorendszer	Hiba a küldő rendszer adatokban!	Error in sending system data!
ErrorSzolgáltato	Hiba a szolgáltató adatokban!	Error in service provider data!
NotFoundInDbVUzlet	Vendéglátó üzlet nem található az adatbázisban!	Catering establishment not found in the database!
NotFoundInDbSzolgalatasiHely	Szolgáltatási hely nem található az adatbázisban!	Service location not found in the database!

5.2.3. Signature error keys

Error key	Error message (Hungarian)	Error message (English)
NullJwtToken	A kérdésben üres a digitális aláírás!	The digital signature in the request is empty!
JwtTokenError	Digitális aláírás hiba!	Digital signature error!
InvalidAudienceJwtToken	Nem engedélyezett digitális aláírás hallgatóság!	Digital signature audience not allowed!
InvalidJwtToken	Nem érvényes digitális aláírás!	Digital signature is not valid!
ExpiredJwtToken	Lejárt digitális aláírás!	Digital signature expired!

5.2.4. Certificate error keys

Error key	Error message (Hungarian)	Error message (English)
NullCertificate	A kérdésben üres a tanúsítvány!	The certificate is empty in the request!
ErrorReadCertificate	Tanúsítvány beolvasás hiba!	Certificate retrieval error!
ErrorCertificate	Tanúsítvány általános hiba!	Certificate general error!

ErrorCheckCertificateType	Tanúsítvány típusa nem meghatározható!	Certificate type cannot be determined!
ErrorCertificateChain	Tanúsítványlánc ellenőrzés hiba!	Certificate chain check error!
ErrorCertificateChainCheck	Tanúsítványlánc ellenőrzés sikertelen!	Certificate chain check failed!
MissmatchCertificateCnCheck	A tanúsítvány CN és a body-ban küldött érték nem egyezik!	The certificate CN and the value sent in the body do not match!
CheckBeforeDateCertificateError	Tanúsítvány még nem érvényes!	Certificate not yet valid!
ExpiredCertificate	Lejárt tanúsítvány!	Expired certificate!
NotTrustedCertificate	Nem megbízható tanúsítvány!	Untrusted certificate!
JsonSyntaxError	JSON szintaktikai hiba: <hiba>	JSON syntax error: <error>

6. Appendices

6.1. Requesting a certificate

6.1.1. Standards and Recommendations

The following table summarises the list of standards and recommendations referenced in this document.

Keyword	
RFC 5280 X.509 CRL	Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile https://www.ietf.org/rfc/rfc5280.txt
RFC 6818 X.509 CRL	Updates to the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile https://tools.ietf.org/html/rfc6818
RFC 6960 OCSP	X.509 Internet Public Key Infrastructure Online Certificate Status Protocol (OCSP) https://tools.ietf.org/html/rfc6960
RFC 2986 PKCS#10	PKCS #10: Certification Request Syntax Specification Version 1.7 https://tools.ietf.org/html/rfc2986

6.1.2. Creating a certificate signing request in a Windows environment

For testing purposes, in a Windows operating system environment, you can use the command line using **certreq** utility to create a certificate signing request

https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/certreq_1

Creating a test certificate signing request

For the certificate signing request example, a configuration file similar to the one below should be created:

```
;
; SampleAuthentication.ini
; Content of the certificate signing request file used for testing
;
[NewRequest]
Subject = "CN=12345678"
FriendlyName = "Példa Tanúsítvány"
KeySpec = 1
KeyLength = 4096
KeyUsage = "CERT_DIGITAL_SIGNATURE_KEY_USAGE | CERT_NON_REPUDIATION_KEY_USAGE | CERT_KEY_ENCIPHERMENT_KEY_USAGE"
HashAlgorithm = SHA256
KeyAlgorithm = RSA
Exportable = TRUE
MachineKeySet = TRUE
SMIME = FALSE
PrivateKeyArchive = FALSE
UserProtected = FALSE
UseExistingKeySet = FALSE
```

```
ProviderName = "Microsoft Enhanced RSA and AES Cryptographic Provider"
ProviderType = 24
RequestType = PKCS10

[EnhancedKeyUsageExtension]
OID=1.3.6.1.5.5.7.3.2
```

It is important to note that in the above example configuration file, the `Exportable=TRUE` parameter allows the created key pair to be moved to another computer, and the `MachineKeySet=TRUE` parameter allows the key pair to be created in the computer's certificate repository, where the issued certificate must be loaded.

The mandatory data in the issued certificate signing request is the registration number of the service location for which the certificate signing request is issued. This data must be entered in the Subject CN field as shown above, the identification number of the service location in the example, therefore, is "12345678".

To generate a certificate signing request from the command line, use the following command:

```
CertReq -New SampleAuthentication.ini SampleAuthentication.csr
```

6.1.3. Create a certificate signing request in a Linux environment

For testing purposes, in a Linux operating system environment, you can use the command line using **openssl** utility to create a certificate signing request

<https://www.openssl.org/docs/manmaster/man1/openssl.html>

Creating a test certificate signing request

For the certificate signing request example, a configuration file similar to the one below should be created:

```
;
; SampleAuthentication.conf
; Content of the certificate signing request file used for testing
;
[ req ]
default_bits    = 4096
default_md      = sha256
prompt         = no
encrypt_key     = no
```

```
distinguished_name = req_distinguished_name
req_extensions = req_extensions

[ req_distinguished_name ]
CN = 12345678

[ req_extensions ]
keyUsage = digitalSignature, keyEncipherment, nonRepudiation
extendedKeyUsage = clientAuth
```

When using a command line, the certificate signing request needs to be created with the following command:

```
openssl req -newkey rsa:4096 -keyout SampleAuthentication.key -config SampleAuthentication.conf -out SampleAuthentication.csr
```

The mandatory data in the issued certificate signing request is the identification number of the service location for which the certificate signing request is issued. This data must be entered in the Subject CN field as shown above, the identification number of the service location in the example, therefore, is “12345678”.

To check the correctness of the data in the certificate signing request file, use the following command:

```
openssl req -in SampleAuthentication.csr -noout -text
```

6.2. Interface message examples

The following sample messages were submitted in a test system of the interface and therefore the identifiers in them do not correspond to any existing system, service provider or organization. The code sections are formed according to the syntax of cURL command-line tool.

6.2.1. Normal order summary message

```
curl --location --request POST 'https://rms.ntaktst.hu/rms/rendeles-osszesito' --header 'Content-Type: application/json' --header 'x-jws-signature: eyJhbGciOiJIUzU1NiI9.eyJ0W0VIV29DgLY_RkMQcTcplxqBnWUpNsuLb4GOBPwAORVe7eRfa89fqv_4zlmHNXmiCkN4dvsQkhJGpd3jG HtRi9yoSUCZB3f9X6-4KiNHawVtaCuDWpZ55U_m0GaNGcWWshIyJBVUYR-B-DPt0Repwnw4LykpPLeWSeIsk4SfKZCCh4fRXJQKiHhGD2XZDnV0CaEFvuQXE7S2H5vN2Znmjguw3WLeYFd5EhL3hX0vZIX2Nvoudmvs thEVwELD16fJyKXvx0-xTLAB34G4G62wISWpbBY_mHgkYGMHCucI2RGyx7y-E_M1Dw1cXats9NyiheiJUrJCNypR_xt7l- WJOLuTovRpcqclDN4RDXV7dLwyafILy_vnwc2_mGCeGfa_ul0o9CQnWWaEx-SfDUOWMhCQ118qsPRZknxSYmxkyzXWz0om8T4aNZGBwGbGK8w0xI-iNIO7-Qzp2hYHbeh_g15mqGTJlxB7_xW-ozGCT2i8K- 1kTuiZ5rRFhEdciDe8b2ioPLGxJFNFB_IH0mtjJOGsjK8FWanLgMkBCrnNq32rvu5iWCCwOvNlt- LnKl0dEBQk1UgnwqdUwroVLHsIjh6VBH86e8U2LogX3cMRxXydy2k4zyLCpJShMrIAdsDkfiW7c0sLzkiRjE4rj7f_weSeMMzj6aPerYjFhX qQ' --header 'x-certificate: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tDQpNSUllavRDQ0JYR2dD0lCQWdJUKFlIdlVRVVVHVHFRkQ3ZRUIIUMnIwNjIRd0RRWUpL b1pJaHJzJTkFRFRUxUUUF3DQpnWTB4TWpBd0JnTlZCQU1NS1ZSRlUxcFVJRTRVUVVzZdFEwRWdRMnhwWlc1MElFRjFkR2dnVTJsbmJtb HVaeUJEDQpRU0F5TURJeU1Sc3dHUUVIEVFRTERCSIVSVk5hVhNCT1ZFRkxMVU5CSURJd01qSXhHakFZQmdOVKB01FVlJGDQpVMX
```

BVSUVSQIZGskJTeUJMWm5RdU1SRXdEd1IEVIFRSErBaENkv1JoY0dWemRERUxNQWtHQTFVRUJoTUNTRIV3DQpIaGNOTWpJe1q
QXINVGN3TVRNNVdoY05NaIf4TWpBeE1UY3dNVE01V2pDQnFURUxNQWtHQTFVRUJoTUNTRIV4DQpGekFWQmdOVkYJBY01Ea0ox
WkdGd1pYtJBjQzBnV0VrdU1RMHdDd1IEVIFRUKRBUXhNVEV4TVJFd0R3WURWUUVJoDQpEQWd4TWpNME5UWXpNakVuTUNVR0E
xVUVDZ3dIVWsxVEIGSkRJQWxMUVRJeU1EQTBnak0xSURFeU16UTFOak15DQpNaIf6TVJFd0R3WURWUUVFEREFneU1qQXDOREl6TI
RFUk1BOEdBMVVFQ3d3SVJFTlPwRTRFRVfVneEVEQU9CZ05WDQpCQVUUVQJFabGJtUmxaekV3Z2dJaU1BMEdDU3FHU0liM0RRRUJB
UUVBQTRJQOR3QXdnZ0ILQW9JQ0FRQzN2TU9mDQpuK2FtSvp2YmxPbFNXS0pvRDkxVjJvSFVackJ0QVVLQ01NUUtBNFILZTBmSFIOO
Uh2RjNRbS9aUk1uV0JIMjhUDQoyZU9sL0xGbGN0RU5KeWk2cmZvOC9iWWpDTWRnTXpEU09qYXArT3RUNTFGaGc0VGZVRm9hSD
NHdnB6aHc1b0xIDQpTVk03aXiOWxMWJlZy2dyQzdqTURzdVJFeHp5NDZlekhiZmxuVDNmWVNpc0N1OFQ0bUs5N1E2cVNpZ3hwdF
NGDQpBWGICUlljT2lJb0hHdmNXRStnand4K1A2bjZuaUR6bGRuVTUxSGE2TFF2UHNYNmU3Q1dqUjZvNW00YXRidG5ODQo2a1J0dG
10eHBWNnIzSXY4SGRtbnliVC8zdHhCSURMR0haNEdTbIM2SEIKnk1LZFlveGFISEZHn2N4VIMNjNIDQpHUzZrNGRrUURRUnl1MDQx
MIZOdFpTUvNsMXNLUThLui95Wk9CbnhtZU95L3pGUFA5YTV0OHnyVWdpVzhLS0R3DQpuOUJDRGw3SHFJRGnkRFgybJJYVvK4dHR
oaDBYv3FEc1M1OVFNvWljZGVnaVVEL1hSRGlS3VsWXBUUEdaVklIDQpmN2FGcmg2VHhsZDdlcytrY3kyYThBQjMr1dFXbHIXWnpH
OC9aWfNYZmh4VDYxbk5hd3d3WXdySxozMGJHdGZuDQpNaXQ4V0VwTWtIC0ZTcXZCaVZWMDRScDltVnczRzI3RlMRNXpYbUJFeW1
4dXNTRktVNXFFZkl1SG9ET2J2OVYxZDQpYay9hNW9RMktLSDU3ZXNjZWJtNVRMQ2dhUXRuQXdsR2RvGtzZ0ZDSk5zZTZlWDYyZktB
dlhXUVBxcBJS3U3DQpYT0wOeS9wVCtuWfY1NHQ1RnVUVhpyY3JDZEVNSFlIS3dwSmZ6d0IEQVFBQm80SUJ4RENDQWNBD0NRWU
RWUjBUDQpCQUi3QURCckJnTIZiUjhFWkRCaU1HQ2dYcUjJaGxbw2RIUndPaTh2ZEdWemVuUXVibJJoYtJOaExtadFMMjUwDQpZV3R
qWVdOc2FXVnVkr0YxZEdoemFXZHvHvZvWuTJFeU1ESXIMMk55YkM5dWRHRnJZMkZqYkdsbGJuUmhKwFJvDQpJmMxuYm1sdVoy
TmhNakF5TWk1amNtd3dnY1FHQ0NzR0FRVUZCd0VCQkIHMO1JRzBNR3NHQ0NzR0FRVUZCekFDDQpobDlvZEhSd09pOHZkR1Z6ZW
5RdWJuUmhHkMk5oTG1oMUWYNTBZV3RqWVdOc2FXVnVkr0YxZEdoemFXZHvHvZvWuTJFeU1ESXIMBUs5DQpjakJGQmdnckJnRUZCUWN3QVIZNWFIUjBjRG92TDN
SbGMzcdBMBtUwWVd0aillTNW9kUzl1ZEdGclkyRmpIR2xsDQpibljZfHsb2MybG5ibWx1WjJOaE1qQXlNaTl2WTNOD01BNEdBMVV
kRHdFQI93UUVBd0IGNERBYUjNtIZiU0FFDQpFekFSTUE4R0RTc0dBUVFCZzhNYUFRTUjBd0I3RXdZRFZSMGxCQXdx3Q2dZSUt3WUJCU
VVIQXjd0h3WURWUjBqDQpCQmd3Rm9BVVRPvXfMvWQ5dTNTdWdPa2R1engrUFQwSVRMRXdiUVIEVllwT0JCWUUVGSjZvTWxuY
kwyd2lLcm9pDQpvZGt4OXNsBUXNN1ZNQTBHQ1Nxr1NJYjNEUUVQC3dVQUE0SUNBUUNkeXE1cTuvbW5UeEVXQU9uQzlwWEs5Y
Wt2DQpWYWo2bk1QeVN3d1JvSFJxWVBIUFU1aXhdyINjdNMrM1gvZvdHVjdzQ3FYMXVsYVhvTndjaDlqUVMrcTkwaEFYDQpKZlpVXV
BEMTRsdjlnWmFpCVBUKzFKU2F3bER1ZkFPMTFyUm5OSlhHQIZNbTdNNGJNd1BubTBnRmFlVfFozUlpiDQpKUNExai9xWHRBNnp0Szd
mUC8xMTNvWVJpMUxQUkNoQJRTYtc5Y0ZOZWpra1cvcnhZKzdIN1NXUGtHeWpnm2JhDQpBTmdSZlk3TOxdn2J2bFp4V0oyWkJnZG
FCSUVreDR0VWRYakxBdU9SK3dKSnE3T1kyZ3JVRi9ueWg4ZUN0RGw4DQpQbFh5aC9LT1ZxSm4xWncwN09QNY90a3pNVFlNyKJoW
HZuZzNTYXZtWTRKSGtpMWIUemU2eHdyVUV0S1kzZ1dPDQpNaNkdVYS9qa0RONTNKGOGxSN1lhVjI5akFNOUImN2I5VHdoemJrNU9K
VThVvkn2bGdyaWRLZFJlQ0wXZfhyUmV0DQpDY2doNnc3Nzg1a0NHVkpneE1aMVf0UGdWSnhVUVZHNDhJcEt2NVU2YjVlcEtBaGd
MNWUyWmMxYUVkaE8yQ0wzDQp6bURDdjgyaFZkR1JlV1F3NkxjQ0JscWd4UzBRSGFjR2szVmJEd3FmVjdmWDJtRWZTQmRFMVZD
L2JXYVdjMFA4DQpqbVMONWjxUTBCVUdHMXILYmg0U2Fzak10TFpaWnVBnHBuRzBMWWhjYt1hRaTJNdTUzVVBXSkI2NFQzUTVvVaH
VBDQo0K1BQQUovbhmjNHIXY1IzeWdRbE0vTGMraTdTSmFrd1hCL3pSSzJZTjFzWkprY3J6ZDFYcklwOTVzckl5TUZkdQoyTmtONUF6Z
1E1WnVQSDNvRhC9PQOKLS0tLS1FTkQgQ0VSVEIGSUNBVEUtLS0tLQ0K' --data-raw '{

```
"szolgaltatoAdatok": {
  "adoszam": "12345632243",
  "vendeglatoUzletRegSzam": "KA22004235"
},
"uzenetAdatok": {
  "uzenetKuldesIdeje": "2022-12-02T18:06:17.961+01:00"
},
"kuldoRendszerAdatok": {
  "rmsRendszerNTAKAzonosito": "Vendeg1",
  "rmsRendszerVerzioszam": "1"
},
"rendelesOsszesitok": [
  {
    "rendelesBesorolasa": "NORMAL",
    "rmsRendelesAzonosito": "3f2f30af-fe09-4109-9ec8-a868b146849f",
    "targynap": "2022-04-19",
    "rendelesKezdetek": "2022-04-19T11:16:26.09+02:00",
    "rendelesVege": "2022-04-19T11:30:26.09+02:00",
    "helybenFogyasztott": true,
    "osszesített": false,
    "fizetesiInformaciok": {
      "rendelesVegosszegeHUF": 1163,
      "fizetesiModok": [
        {
          "fizetesiMod": "KESZPENZHUF",
          "fizetettOsszegHUF": 1165
        }
      ],
      "fizetesiMod": "KEREKITES",
    }
  }
]
```

```

        "fizetettOsszegHUF": -2
    }
}
},
"rendelesTetelek": [
{
    "megnevezes": "Tétel 1",
    "fokategoria": "ETEL",
    "alkategoria": "DESSZERT",
    "afaKategoria": "A_5",
    "bruttoEgysegar": 1001,
    "mennyisegiEgyseg": "DARAB",
    "mennyiseg": 1,
    "tetelszam": 1,
    "rendelesidopontja": "2022-04-19T11:30:26.09+02:00",
    "tetelOsszesito": 1001
},
{
    "megnevezes": "Tétel 2",
    "fokategoria": "EGYEB",
    "alkategoria": "SZERVIZDIJ",
    "afaKategoria": "A_5",
    "bruttoEgysegar": 1001,
    "mennyisegiEgyseg": "DARAB",
    "mennyiseg": 1,
    "tetelszam": 0.01,
    "rendelesidopontja": "2022-04-19T11:30:26.09+02:00",
    "tetelOsszesito": 10
},
{
    "megnevezes": "Tétel 2",
    "fokategoria": "EGYEB",
    "alkategoria": "SZERVIZDIJ",
    "afaKategoria": "C_27",
    "bruttoEgysegar": 200,
    "mennyisegiEgyseg": "DARAB",
    "mennyiseg": 1,
    "tetelszam": 0.01,
    "rendelesidopontja": "2022-04-19T11:30:26.09+02:00",
    "tetelOsszesito": 2
},
{
    "megnevezes": "Tétel 3",
    "fokategoria": "EGYEB",
    "alkategoria": "KEDVEZMENY",
    "afaKategoria": "A_5",
    "bruttoEgysegar": -200,
    "mennyisegiEgyseg": "DARAB",
    "mennyiseg": 1,
    "tetelszam": 1,
    "rendelesidopontja": "2022-04-19T11:30:26.09+02:00",
    "tetelOsszesito": -200
},
{
    "megnevezes": "Tétel 4",
    "fokategoria": "EGYEB",
    "alkategoria": "BORRAVALO",
    "afaKategoria": "E_0",
    "bruttoEgysegar": 150,
    "mennyisegiEgyseg": "DARAB",
    "mennyiseg": 1,
    "tetelszam": 1,

```



```

BEMTRsdjInWmFpCvBUKzFKU2F3bER1ZkFPMTfyUm5OSihHQIZNbTdNNGJNd1BubTBnRmFIVFozUlpIDQpKUnExai9xWHRBNnp0Szd
mUC8xMTNvWVJpMuxQUkNoQIRTYTc5Y0ZOZWpra1cvcnhZKzdIN1NXUGtHeWpnM2JhDQpBTmdSZlk3T0xnd2J2bFp4V0oyWkJnZG
FCSUVreDR0VWRYakxBdU9SK3dKSnE3T1kyZ3JVri9ueWg4ZUN0RGw4DQpQbFh5aC9LT1ZxSm4xWncwN09QNY9Oa3pNVFlNykJoW
HZuZzNTYXZtWTRKSGtpMWIUemU2eHdyVUV0S1kzZ1dPDQpaNkdVYS9qa0RONTNKGxSN1lhVjI5akFNOUImN2I5VHdoemJrNU9K
VThVvkn2bGdyaWRLZfJlQ0wxZFhyUmV0DQpDY2doNnc3Nzg1a0NHVkpneE1aMVf0UGdWSnhVUVZHNDhJcEt2NVU2YjVlcEtBaGd
MNWUyWmMxYUvkaE8yQ0wzDQp6bURDdjgyaFZkR1JlV1F3NkxjQ0JscWd4UzBRSGFjR2szVmJEd3FmVjdmWDJtRWZTQmRfMVZD
L2JXYVdjMFA4DQpqbVMONWJxUTBCVudHMXILYmgOU2Fzak10TFpaWnVBnHBuRzBMWHJyT1hRaTJNdTUzVVBXSkI2NFQzUTVvaH
VBDQo0K1BQQUovbmhjNHlXY1lZeWdRbEOvTGMraTdTSmFrd1hCL3pSSzJZTjFzWkprY3J6ZDFYcklwOTVzckI5TUZkDQoyTmtONUF6Z
1E1WnVQSDNvRHc9PQOKLS0tLS1FTkQgQ0VSVEIGSUNBVEUtLS0tLQOK' --data-raw '{
  "szolgaltatoAdatok": {
    "adoszam": "12345632243",
    "vendeglatoUzletRegSzam": "KA22004235"
  },
  "uzenetAdatok": {
    "uzenetKuldesIdeje": "2022-12-02T18:06:48.032+01:00"
  },
  "kuldoRendszerAdatok": {
    "rmsRendszerNTAKAazonosito": "Vendeg1",
    "rmsRendszerVerzioszam": "1"
  },
  "rendelesOsszesitok": [
    {
      "hivatkozottRendelesOsszesito": "3f2f30af-fe09-4109-9ec8-a868b146849f",
      "rendelesBesorolasa": "HELYESBITO",
      "rmsRendelesAazonosito": "52ae3efb-c2bb-4f17-b102-9503a2a1541e",
      "targynap": "2022-04-19",
      "rendelesKezdetek": "2022-04-19T11:16:26.09+02:00",
      "rendelesVege": "2022-04-19T11:30:26.09+02:00",
      "helybenFogyasztott": true,
      "osszesített": false,
      "fizetesiInformaciok": {
        "rendelesVegosszegeHUF": 1000,
        "fizetesiModok": [
          {
            "fizetesiMod": "KESZPENZHUF",
            "fizetettOsszegHUF": 1000
          }
        ]
      }
    },
    {
      "rendelesTetelek": [
        {
          "megnevezes": "Tétel 1",
          "fokategoria": "ETEL",
          "alkategoria": "DESSZERT",
          "afaKategoria": "C_27",
          "bruttoEgysegar": 1000,
          "mennyisegiEgyseg": "DARAB",
          "mennyiseg": 1,
          "tetelszam": 1,
          "rendelesIdopontja": "2022-04-19T11:30:26.09+02:00",
          "tetelOsszesito": 1000
        }
      ]
    }
  ]
}'

```

6.2.3. Cancelled order summary message

```

curl --location --request POST 'https://rms.ntaktst.hu/rms/rendeles-osszesito' --header 'Content-Type: application/json' --header 'x-jws-signature:

```

```

eyJhbGciOiJSUzI1NiJ9..H8zY0tdVR61JxbVvIr6jM1BLzrlqZDqRaNNszcVkobFF0htlGhqxBaxSQlpP0Vj8VsBlJxod4x4BSXcHSSaCglwFf
Z9kpjN-rSZDpkrGlmKzgyWoPHiHoCg5Wiml4xj1ruJhFvkVfjezhp8RIWLB-_AbZ8SveW-
_BxVtz1wwVRQcMDKOnQFziArygP5NqiBxYbTe3qu53rbWDhkpTpyOBnQ1r4Wx5x_uvrtugKlmnKneCEoSUEWug84iPasb2vgWt1icPP
5AcyVpL_eiPachRIl6OSfQ2cVBxBP2Hz4GZFRVolfVqN4xW4ziKHdN7S1fHb5rEzSZ4PaZDm76483Mr0yH0WLPpyvJBXnAggYlwy02uFF
3Y0d7l7J80YGLCHN72xbAj0bHe5R_L5vhG6cFoZwN7mZeQEc9lphbOkBwDbgEihEfa1Jb8u0xUPsLEaYpbjyZ4olvm8WQXzRjHwZ4Wt
WvTzzk_yYJXESfdG1QKJQRUDXqnjdHPtZT49_QDdlzrcQy4N_SU9-
rDdb3QeEeg_BWOZjRzqGDhSXsOXPsUIE6c3pB8eQ79nCZ0ArtKASuhkymba2x9_ZYyvVBlwbvUSIG0GbJ-
D0Lyh5l9CbW_fcXBg3wWtA8aJv_fid9NUE5IDPUdwcDGTsWGO9DHployGZc1Wt6xjUy4OumiY' --header 'x-certificate:
LS0tLS1CRUdJTiBDRVJUSUZQOFURS0tLS0tDQpNSUllaVRDQ0JR2dBd0lCQWdJUKFldlVRVWVHVFRkQ3ZRUllUMnIwNjRlRd0RRWUpL
b1pJaHJZTkFRURUxUUUFDQ3pnbWtB4TWpBd0JnTlZCQU1NS1ZSRlUxcFVJRTRVUUVVzdFwRWdRMnhwWlcl1MElFRJfKfR2dnVTJsbmJtb
HVaeUJEDQpRU0F5TURJeU1Sc3dHUVIEVFRTERCSIVSVk5hVhNCT1ZFRkxMVU5CSURJd01qSXhHakFZQmdOVkBJb01FVlJGdQpVMX
BVSUVSQlZGSKJTeUJMWm5RdU1SRXEd1IEVIFRSEBaNkV1JoY0dWemRERUxNQWtHQTFVRUJoTUNTRIV3DQplaGNOTWpJeE1q
QXlINVGN3TVRNNVdoY05NaIf4TWpBeE1UY3dNVE01V2pDQnFURUxNQWtHQTFVRUJoTUNTRIV4DQpGekFWQmdOVkBJb01Ea0ox
WkdGd1pYtJlBjQzBnV0VrdU1RMHdDd1IEVIFRUKRBUxhNVEV4TVJfD0R3WURWUUVJoDQpEQWd4TWpNME5UWXPnAKvUUNVR0E
xVUVDZ3dlVWsxVElGSgRjQWxMUVRJeU1EQTBNAk0xSURFeU16UTFOak15DQpNaIf6TVJfD0R3WURWUUVFEREFneU1qQXhORElGfI
RFUk1BOEdBMVVFQ3d3SVJFTlplWRTFRVfVneEVEQU9CZ05WDQpCQVUUVQjFabGJtUmxaekV3Z2dJaU1BMEduDU3FHU0liM0RRRUJb
UUVBQTRJQ0R3QXdnZ0lLQW9JQ0FRQzN2TU9mDQpuK2FtSVp2YmxPbFNXSOvpRDkxVjJvSFVackJ0QVVLQ01NUUtbNfILZTBmSFIOO
Uh2RjNRbS9aUk1uV0JIMjhUDQoyZU9sL0xGbGN0RU5KeWk2cmZvOC9iWWpDTWRnTXpEU09qYXArT3RUNTFGaGc0VGZVRm9hSD
NHdnB6aHc1b0xIDQpTVk03aXlJ0WxMWJlZy2dyQzdQURzdVJFeH5NDZlekhiZmxuVDNmWVNpc0N1OFQ0bUs5N1E2cVNpZ3hwdF
NGDQpBWBGICUlljTlJlB0hHdmNXRStnand4K1A2bjZuaUR6bGRuVTUxSGE2TF2UHNYNmU3Q1dqUjZvNW00YXRidG5ODQo2a1J0dG
10eHBWNnIzSY4SGRtbniVC8zdHhCSURMR0haNEdTbIM2SEIKNk1LZFlveGFSEZHN2N4VIMNjNIDQpHUzZrNGRrUURRUnl1MDQx
MlZOdFpTUUVNsMXNLUthLUI95Wk9CbnhtZU95L3pGUFA5YTV0OHNYVWdpVzhLS0R3DQpuOUJDRGw3SHFJRGNKRfgybJlYVVK4dHR
oaDBYv3FEc1M1OVFNvWlJZGVnaVVEL1hSRGlsS3VsWXBUUEdaVklIDQpmN2FGcmg2VHhsZDdlcytrY3kyYThBQjMrdd1FXbHIXWnpH
OC9aWfNYZmh4VDYxbk5hd3d3WXdySXozMGJHdGZuDQpNaXQ4V0VwTWtIc0ZTcXZCaVZWMDRScDltVnczRzI3RlMRnXpYbUJFeW1
4dXNTRktVNXFFZk1lSG9ET2J2OVYxZDQpYay9hNW9RMktLSDU3XZNjZjVjNVNRMQ2dhUXRuQXdsR2RpVGtzZ0ZDSk5zZTZlWDYyZktB
dlhXUVBxcjBjS3U3DQpYT0w0eS9wVCtuWfY1NHQ1RnVUVHpyY3JDZEVNSFlIS3dwSmZ6d0IEQVFBQm80SUJ4RENDQWNbD0NRWU
RWUjBUDQpCQUl3QURCcKJnTlZlUjFwKRCaU1HQ2dYcUjJaGxwb2RIUndPaTh2ZEdWemVuUXViblj0YtJOaExtadFMMjUwDQpZV3R
qWVdOc2FXVnVkr0YxZEdoemFXZHvHvZVuWTJFeU1ESXIMMk55YkM5dWRHRnJZMkZqYkdsbGJuUmhKWFJvDQpJmMxuYm1sdVoy
TmhNakF5TWk1amNtd3dnY1FHQ0NzR0FRVUZCd0VCQkIHMO1JRzBNR3NHQ0NzR0FRVUZCekFDDQpobDlvZEhSd09pOHZKR1Z6ZW
5RdWJUUmhHmK5oTG1oMUWYNTBZV3RqWVdOc2FXVnVkr0YxZEdoemFXZHvHvZVuWTJFeU1ESXIMMk55YkM5dWRHRnJZMkZqYkdsbGJuUmhKWFJvDQpJmMxuYm1sdVoy
NTBZV3RqWVdOc2FXVnVkr0YxZEdoemFXZHvHvZVuWTJFeU1ESXIMMk55YkM5dWRHRnJZMkZqYkdsbGJuUmhKWFJvDQpJmMxuYm1sdVoy
SbGMzcdBMBmtUwWVd0allTNW9kUz1ZEdeGclkyRmpir2xsDQpibJ0ZfHsb2MybG5ibWx1WjJOaE1qQXlNA1Tl2WTN0d01BNEdBMVV
kRHdFqj93UUVBd0lGNERBYUJnTlZlUjFwKRCaU1HQ2dYcUjJaGxwb2RIUndPaTh2ZEdWemVuUXViblj0YtJOaExtadFMMjUwDQpZV3R
VVIQXjd0h3WURWUjBjDQpCQmd3Rm9BVRPvXfMvWQ5dTNTdWdPa2R1engrUFQwSVRMXRdIUUVIEVlIwT0JCWUUVGSjZvTWxuY
kwyd2lLcm9pDQpvZGt4OXNsbUxNN1ZNQTBHQ1Nxr1N1JYjNEUUVQC3dVQEU0SUNBUUNkeXE1cUvbw5UeEVXQU9uQzlwWES5Y
Wt2DQpWYWo2bk1QeVN3d1JvSFJxWVBIUFU1aXhydlNjdnNrM1gvZvdHVjdZ3FYMXVsYVhvTndjaDIqUVMrcTkwaEFYDQpKZlpXVX
BEMTRsdjlnWmFpCvBUKzFKU2F3bER1ZkFPMFTfYUm5OSlHhQJZNbTdNNGJNd1BubTbnRmFIVFozUlpiDQpKUNExai9xWHRBNnp0Szd
mUC8xMTNvWVJpMUxQUkNoQJRTYTc5Y0ZOZWpra1cvcnhZKzdln1NXUGtHeWpnmM2JhDQpBTmdSZlk3T0xnd2J2bFp4V0oyWkNjZG
FCSUVreDR0VWRyYakxBdU9SK3dKSnE3T1kyZ3JVRI9ueWg4ZUN0RGw4DQpQbFh5aC9LT1ZxSm4xWncwN09QNY90a3pNVFlNykJoW
HZuZzNTYXZtWTRKSGtpMWIUemU2eHdyVUV0S1kzZ1dPDQpNaNkdVYS9qa0RONTNKGXSN1lhVjl5akFNOUImN2I5VHdoemJrNU9K
VThVVKn2bGdyaWRLZFIJQ0wXZFhyUmV0DQpDY2doNnc3Nzg1a0NHVkpneE1aMVf0UGdWSnhVUVZHNDhJcEt2NVU2YjVlcEtBaGd
MNWUyWmMxYUUVkaE8yQ0wzDQp6bURDdjgyaFZkr1JlV1F3NkxjQ0JscWd4UzBRSGFjR2szVmJEd3FmVjdmWDJtRWZTQmRFMVZD
L2JXYVdjMFA4DQpqbVMOmNWJxUTBCVUdHMxILYmg0T5Fzak10TFpaWnVBNHBuRzBMWHJyT1hRaTJNdTUzVVBXSKI2NFQzUTVVaH
VBDQo0K1BQKUovbmhJNHIXY1ZeWdRbE0vTGMraTdTSMFrd1hCL3pSSzJZFzWkprY3J6ZDFYcklwOTVzckl5TUZkdQqoyTmtONUF6Z
1E1WnVQSDNvRhC9PQOKLS0tLS1FTkQgQ0VSVEIGSUNBVEUtLS0tLQOK' --data-raw '{
  "szolgaltatoAdatok": {
    "adoszam": "12345632243",
    "vendeglatoUzletRegSzam": "KA22004235"
  },
  "uzenetAdatok": {
    "uzenetKuldesIdeje": "2022-12-02T18:07:05.359+01:00"
  },
  "kuldoRendszerAdatok": {
    "rmsRendszerNTAKazonosito": "Vendeg1",
    "rmsRendszerVerzioszam": "1"
  },
  "rendelesOsszesitok": [
    {
      "hivatkozottRendelesOsszesito": "52ae3efb-c2bb-4f17-b102-9503a2a1541e",
      "rendelesBesorolasa": "SZTORNO",
      "rmsRendelesAzonosito": "b62a5195-b70b-46fa-a13c-a680606ff397",
      "targynap": "2022-04-19"
    }
  ]
}

```

```
}  
}  
}
```

6.2.4. Daily closure message

```
curl --location --request POST 'https://rms.ntaktst.hu/rms/napi-zaras' --header 'Content-Type: application/json' --header 'x-jws-  
signature: eyJhbGciOiJIUzI1NiJ9..GfzEh3WZFDI3AFISpVPD9-  
wOPNrkDDMYwRF4Yd3BD_Boi5EIRfL39q2JWsgLL7olohu4AiT0SdzOPPRJ61VK9dAlwOJ6pvg1H82RiQVEZPON2XO2B3DalfToKJ1HT  
9CVJaho68tS6uMUf5dY9kcc3dXkuE-iL0q4scwzW8Dau-nN7bZ-  
lg3OnECKOJGgDgpLe6vwWBZNMOT9eqsPP1WhulfOhsFtrXJKt3ej9JZaB7w2AxW43fBTI-  
3Uk9Uc53gdvY6kcrq6yY654xtfvEV2GFQaPCOCqfJGopaqDD4VDQdLfm8cRS6dEubFOrLIISj2FJRE6PmhaBrfkFenK2rO9p1P5dr02hpT  
IQ5F0pRnEm0D-Gey9G3B1Tta1f67Rfhr_-  
J7Y7xGg5bL2TSR9lyhlaF2JsH2T6iQAXkDYOPXrLEyYosmtASjYvC6MMnCbh20kpZzojs4JhInM4Y00ttqPY-  
pKDwlG0K9975Njyzcs3KmdvMWHjfoZ0qYN1AF0ON8uf7ZCUDL1O80VYlpgU5-azgADDnMvMwslS-  
O14Z3u9vtNet9Gkm1V8ShP1C7QMj8QFlpJsYt7qnaiw5U5UXS0jMdia1i08HiUPLROhv5vwidDaMZYPffOMUSwFD_jA6DreoxnUFizVi  
qo6EcQIIdJkWRKO0aTVbEe8p4U' --header 'x-certificate:  
LS0tLS1CRUdJTiBDRVJUSUZlQ0FUR50tLS0tDQpNSUlllaVRDQ0JYR2dBD0lCQWdJUKFlIdIVRVVHVHFRKQ3ZRUIIUmlwNjIRd0RRWUpl  
b1pJaHJZTkFRRUxUcUUF3DQpnWTB4TWpBd0JnTlZCQU1NS1ZSRlUxcFVJRTRVUUVVzdFwRWdRMnhwWlcl1MElFRJfK2dnVTJsbmJtb  
HVaeUJEDQpRU0F5TURJeU1Sc3dHUVEIFRTERCSIVSVk5hVhNCT1ZFRkxMVU5CSURJd01qSXhHakFZQmdOVkBJb01FVlJGDDQpVMX  
BVSUVSQIZGSKJTeUJMWm5RdU1SRXdEd1IEVIFRSEBENkv1JoY0dWemRERUxNQWtHQTFVRUJoTUNTRIV3DQplaGNOTWpJeE1q  
QXINVGN3TVRNNVdoY05NalF4TWpBeE1UY3dNVE01V2pDQnFURUxNQWtHQTFVRUJoTUNTRIV4DQpGekFWQmdOVkBJb01Ea0ox  
WkdGd1pYtjBJQzBnV0VrdU1RMHdDd1IEVIFRUKRBUXhNVEV4TVJFd0R3WURWUWVJoDQpEQWd4TWpNME5UWXPnAKVuTUNVR0E  
xVUVDZ3dlVWsxVEIGSKRJQWxMUVRJeU1EQTBNAk0xSURFeU16UTFOak15DQpNalF6TVJFd0R3WURWUWVFEREFneU1qQXhOREl6TI  
RFUk1BOEdBMVVFQ3d3SVJFTlplWRTFRVFnEVEQU9CZ05WDQpCQVUWUQJFabGJtUmxaekV3Z2dJaU1BMEdDU3FHU0liM0RRURJJB  
UUVBQTRJQ0R3QXdnZ0lLQW9JQ0FRQzN2TU9mDQpuK2FtSVp2YmxPbFNXS0pvdKxVjJvSFVackJ0QVVLQ01NUUtBNFILZTBmSFIOO  
Uh2RjNRbS9aUk1uV0JIMjhUDQoyZU9sL0xGbGN0RU5KeWk2cmZvOC9iWWpDTWRnTXpEU09qYXArT3RUNTfGaGc0VGZVRm9hSD  
NHdnB6aHc1b0xIDQpTVk03aXlJOWxMWJlZy2dyQzdQURzdVJFeH5NDZlekhiZmxuVDNmWVNpc0N1OFQ0bUs5N1E2cVNpZ3hwdF  
NGDQpBwGICUlljT2JlJb0hHdmNXRStnand4K1A2bjZuaUR6bGRuVtUXSGE2TF2UHNYNmU3Q1dqUjZvNW00YXRidG5ODQo2a1J0dG  
10eHBWNNlZSXY4SGRtbniVC8zdHhCSURMR0haNEdTbIM2SEIKNk1LZFlveGFISEZH2N4VVMNjNIDQpHuzZrNGRUURRUnl1MDQx  
MlZOdFpTUUVNsMXNLUThLUI95Wk9CbnhtZU95L3pGUFA5YTV0OHNYvWdpVzhLS0R3DQpuOUJDRGw3SHFJRGNKRfgyblJYVvK4dHR  
oaDBYV3Fec1M1OVFNvWljZGVnaVVEL1hSRGlsS3VsWXBUUEdaVklIDQpmN2FGcmg2VHhsZDdlcytrY3kyYThBQjMrdd1FXbHIXWnpH  
OC9aWfNYZmh4VDYxbk5hd3d3WXdySXozMGJHdGZuDQpNaXQ4V0VwTWtIC0ZTcXZCaVZWMDRScDltVnczRzI3RlMRNXpYbUJFeW1  
4dXNTRktVNXFFZk1lSG9ET2J2OVYxZDQpYay9hNW9RMktLSDU3Z3NjZVJtNVRMQ2dhUXRuQXdsR2RpVGtzZ0ZDSk5zZTZlWDYyZktB  
dlhXUVBxclBJs3U3DQpYT0w0eS9wVctuWfY1NHQ1RnVUVHpyY3JDZEVNSFlIS3dwSmZ6d0lEQVFBQm80SUJARENDQWNBD0NRWU  
RWUjBUDQpCQUl3QURCckJnTlZlUjhfWkRCaU1HQ2dYcUJJaGxwb2RlUndPaTh2ZEdWemVuUXViblJoYtJOaExtadFMMjUwDQpZV3R  
qWVdOc2FXVnVkr0YxZEdoemFXZHvHvZvUuTJFeU1ESXIMMk55YkM5dWRHRnJZMkZqYkdsbGJuUmhKWfJvDQpJmMxuYm1sdVoy  
TmhNakF5TWk1amNtd3dnY1FHQ0NzR0FRVUZCd0VCQkIHMO1JRzBNR3NHQ0NzR0FRVUZCekFDDQpobDlvZEhSd09pOHZkR1Z6ZW  
5RdWJuUmhMhMk5oTG1oMUWYNTBZV3RqWVdOc2FXVnVkr0YxZEdoemFXZHvHvZvUuTJFeU1ESXIMMk55YkM5dWRHRnJZMkZqYkdsbGJuUmhKWfJvDQpJmMxuYm1sdVoy  
NTBZV3RqWVdOc2FXVnVkr0YxZEdoemFXZHvHvZvUuTJFeU1ESXIMMk55YkM5dWRHRnJZMkZqYkdsbGJuUmhKWfJvDQpJmMxuYm1sdVoy  
SbGMzcBDMbTUwVWVd0aillTNW9kUz1ZEdGclkyRmpir2xsDQpibljOZfHsb2MybG5ibWx1WjJOaE1qQXlNaTl2WTNOD01BNEdBMVV  
kRHdFQ9i3UUVBd0lGNERBYUJnTlZlUjhfWkRCaU1HQ2dYcUJJaGxwb2RlUndPaTh2ZEdWemVuUXViblJoYtJOaExtadFMMjUwDQpZV3R  
VVIQXjd0h3WURWUjBqDQpCQmd3Rm9BVVRPvXfMvWQ5dTNTdWdPa2R1engrUFQwSVRMXRdIUVEIiwT0JCWUUVGSjZvTWxuY  
kwyd2lLcm9pDQpvZGt4OXNSbUxNN1ZNQTBHQ1Nxr1NJYjNEUUVQC3dVQUE0SUNBUUNkeXE1cUvbw5UeEVXQU9uQzlwWES5Y  
Wt2DQpWYWo2bk1QeVN3d1JvSFJxWVBIUFU1aXhYdlnJdnNrM1gvZvdHVjdZQ3FYMXVsYVhvTndjaDlqUVMrcTkwaEFYDQpKZlpXVX  
BEMTRsdjlnWmFPcVBUKzFKU2F3bER1ZkFPMTfyUm5OSlhHQIZNbTdNNGJNd1BubTbnRmFlVfFozUlpiDQpKUnExai9xWHRBNnp0Szd  
mUC8xMTNvWVJpMUxQUkNoQJRTYtC5Y0ZOZWpra1cvcnhZKzdln1NXUGtHeWpnM2JhDQpBTmdSZlk3T0xnd2J2bFp4V0oyWkNjZG  
FCSUVreDR0VWRyAkxBdU9SK3dKSnE3T1kyZ3JVRI9ueWg4ZUN0RGw4DQpQbFh5aC9LT1ZxSm4xWncwN09QNY9a03pNVFlNjYoW  
HZuZzNTYXZtWTRKSGtpMWIUemU2eHdyVUv0S1kzZ1dPDQpANkdVYS9qa0RONTNKOgXSN1lhVjl5akFNOUImN2I5VHdoemJrNU9K  
VThVvkn2bGdyaWRLZFJlQ0wxZFhyUmV0DQpDY2doNnc3Nzg1a0NHVkpneE1aMVFOUGdWSnhVUVZHNDhJcEt2NVU2YjVlcEtBaGd  
MNWUyWmMxYUvkaE8yQ0wZDQp6bURDdjgyaFZKR1JlV1F3NkxjQ0JscWd4UzBRSGFJR2szVmJEd3FmVjdmWDJtRWZTQmRFMVZD  
L2JXYVdjMFA4DQpqbVMONWJxUTBCVUdHMXlYmg0U2Fzak10TFpaWnVBNHBuRzBMWHjYt1hRaTJNdTUzVVBXSKI2NFQzUTVvAh  
VBDQ0K1BQQUovbmhjNHIXY1IzeWdRbE0vTGMraTdTSmFrd1hCL3pSSZJtZfZwKprY3J6ZDFYcklwOTVzckl5TUzkdQoyTmtONUF6Z  
1E1WnVQSDNvRhC9PQOKLS0tLS1FTkQgQ0VSVEIGSUNBVEUtLS0tLQOK' --data-raw '{  
  "szolgaltatoAdatok": {  
    "adoszam": "12345632243",  
    "vendeglatoUzletRegSzam": "KA22004235"  
  },  
  "uzenetAdatok": {  
    "uzenetKuldesideje": "2022-12-02T18:09:02.846+01:00"  
  },  
}
```

```

"kuldoRendszerAdatok": {
  "rmsRendszerNTAKAzonosito": "Vendeg1",
  "rmsRendszerVerzioszam": "1"
},
"zarasiInformaciok": {
  "targynap": "2022-12-02",
  "targynapBesorolasa": "NORMAL_NAP",
  "nyitasIdopontja": "2022-12-02T00:00:00.001+02:00",
  "zarasIdopontja": "2022-12-02T16:59:59.999+02:00",
  "osszesBorravalo": "0"
}
}

```

6.2.5. Verification message

```

curl --location --request POST 'https://rms.ntaktst.hu/rms/ellenorzes' --header 'x-jws-signature:
eyJhbGciOiJIUzI1Ni9iLCJmZGZlZ3ZFdjI3AFiSpvPD9-
wOPNrkDMYwFRF4Yd3BD_Boi5EIRfL39q2JWsgLL7olohu4AiT0SdzkOPPRJ61VK9dAlwOJ6pvg1H82RiQVEZPON2XO2B3DalfToKJ1HT
9CVJaho68tS6uMuF5Y9kcq3dXkuE-iLQ4scwzW8Dau-nN7bZ-
lg3OnECKOJGgDgpLe6vwWBZNMOT9eqsPP1WhulfOhsFtrXJKt3ej9JJaB7w2AxW43fBTI-
3Uk9Uc53gdy6kcrq6yY654xtfvEV2GfQaPCOCqfJGopaqDD4VDqDLfM8cRS6dEubFOrLIISj2FJRE6PmhaBrfkFenK2rO9p1P5dr02hpT
IQ5F0prNEm0D-Gey9G3B1Tta1f67Rfhr_-
J7Y7xGg5bL2TSR9lyhlaF2JsH2T6iQAXkDYOPxRLEyYosmtASjYvC6MMnCbh20kpZzojs4JlInM4Y00ttqPY-
pKDwLg0K9975Njyzcs3KmdvMWHjfOzOqYN1AF0ON8uf7ZCUDL1O80VYlpgU5-azgADDnMvMwslS-
O14Z3u9vtNet9Gkm1V8ShP1C7QMj8QFlpJsYt7qnaiw5U5UXS0jMdIA1i08HiUPLROhv5vwidDaMZYPffOMUSwFD_jA6DreoxnUFizVi
qo6EcQIIdJkwrKOa0aTVbEe8p4U' --header 'x-certificate:
LS0tLS1CRUdJTiBDRVJUSUZQOFURS0tLS0tDQpNSUllaVRDQ0JR2dBD0lCQWdJukFIdlVRVVVHWFkRkQ3ZRUIIUMnIwNjIRdORRWUpL
b1pJaH2JtKfRRUxUUUf3DQpnWTB4TWpBd0JnTIZCCQ1NS1ZSRlUxcFVJRtTVUUVVzdFEwRWdRMnhwWlc1MElFRjFkR2dnVTJsbmJtb
HVaeUJEDQpRU0F5TURJeU1Sc3dHUUVIEVFRTERCSlVSvk5hVKNCT1ZFRkxMVU5CSURJd01qSXhHakFZQmdOVKB01FVlJGDQpVMX
BVSUUVSQlZGSKJTeUJMWMw5RdU1SRXdEd1IEVlFRSERBaENkV1JoY0dWemRERUxNQWtHQTFVRUJoTUNTRIV3DQplaGNOTWpJe1q
QXlINVGN3TVRNNVdoY05NalF4TWpBeE1UY3dNVE01V2pDQnFURUxNQWtHQTFVRUJoTUNTRIV4DQpGekFWQmdOVKB01Ea0ox
WkdGd1pYtJBJQzBnV0VrdU1RMHdDd1IEVlFRUkRBUxhNVEV4TVJFd0R3WURWUWJoDQpEQWd4TWpNME5UWXPnNakVuTUNVR0E
xVUVDZ3dlVWsxVElGSKRJQWxMUVRJeU1EQTBnak0xSURFeU16UTFOak15DQpNalF6TVJFd0R3WURWUWFEREfneU1qQXDOREl6TI
RFUk1BOEDBMVVFQ3d3SVJFTltpWRTFRVfVneEVEQU9CZ05WDQpCQVUWQjFabGJtUmxaekV3Z2dJaU1BMEduDU3FHU0liM0RRRUJB
UUVBQTRJQ0R3QXdnZ0lLQW9JQ0FRQzN2TU9mdDQpK2FtSvp2YmxPbFNXSOvpRDkxVjJvSFVackJ0QVVLQ01NUUtbNfILZTBmSFIOO
Uh2RjNRbS9aUk1uV0JIMjhUDQoyZU9sL0xGbGN0RU5KeWk2cmZvOC9iWWpDTWRnTXpEU09qYXArT3RUNTFGaGc0VGZVRm9hSD
NHdnB6aHc1b0xIDQpTVk03aXJiOWxMWjIzY2dyQzdQURzdVJFeH5NDZlekhiZmxuVDNmWVNpc0N1OFQ0bUs5N1E2cVnpZ3hwdF
NGDQpBWGICUllJ2IjB0hHdmNXRStnand4K1A2bjZuaUR6bGRuVtUxSGE2TFF2UHNYNmU3Q1dqUjZvNW00YXRidG5ODQo2a1J0dG
10eHBWnNlZSYX45GRtbnliVC8zdHhCSURMR0haNEdBtBmI2SEIKNk1LZFlveGFISEZH2N2N4VVMNjNIDQpHuzZrNGRfUURRUnl1MDQx
MlZ0dFpTUUVNsMXNLUThLUi95Wk9CbnhtZU9SL3pGUFA5YTV0OHNYVWdpVzhLSOR3DQpuOUJDRGw3SHFJRGNGKRFgyblUYVVK4dHR
oaDBYV3FEc1M1OVFNvWljZGVnaVVEL1hSRGlsS3VsWXBUUEdaVklIDQpmN2FGcmg2VHhsZDdlcyY3kyYThBQjMrdd1FXbHIXWnpH
OC9aWfNYZmh4VDYxbk5hd3d3WXdySXozMGJHdGZuDQpNaXQ4V0VwTWtIc0ZTcXZCaVZWMDRScDltVnczRzI3RlMRNXPYbUJFeW1
4dXNTRktVNXFFZki1SG9ET2J2OVYxYDQpYay9hNW9RMktLSDU3ZXRjZjVjNVmRmQ2dhUXRUQXdsR2RpGtzZ0ZDsk5zZTZWdYyZktB
dlhXUVBXclBJS3U3DQpYT0w0eS9wVctuWfY1NHQ1RnVUVHpyY3JDZEVNSFIIS3dwSmZ6d0IEQVFBQm80SUJ4RENDQWNBd0NRWU
RWUjBUDQpCQlU3QURCckJnTIZiUjhFWkRCaU1HQ2dYcUjJaGxwb2RIUndPaTh2ZEdWemVuUXVibJJoYtTJOaExtadFMMjUwDQpZV3R
qWVdOc2FXVnVkr0YxZEdoemFXZHvVhVzVuWTJFeU1ESXIMMk55YkM5dWRHRnJZMkZqYkdsbGJuUmhkWFJvDQpJmMxuYm1sdVoy
TmhNakF5TWk1amNtd3dnY1FHQ0NzR0FRVUZCd0VCQklHM01JRzBNR3NHQ0NzR0FRVUZCekFDDQpobDlvZEhSd09pOHZkR1Z6ZW
5RdWJUUmhMk5oTG1oMUWynTBZV3RqWVdOc2FXVnVkr0YxZEdoemFXZHvVhVzVuWTJFeU1ESXIMbU5sDQpjakJGQmdnckJnRUZCUWN3QVIZNWFIUjBjRG92TDN
SbGmZcDBMBtUwWVd0allTNW9kUzl1ZEdGclkyRmpir2xsDQpibJJoZfHsb2MybG5ibWx1WjJOaE1qQXlNaTI2WTN0d01BNEdBMMV
kRHdFQj93UUVBd0lGNERBYUJnTIZiU0FFDQpFekfSTUE4R0RTc0dBuVFCZzhNYUFRTUjBd0l3RXdZRFZSMGxQXdx3Q2dZSut3WUJCU
VVIQXdd0h3WURWUjBqDQpCQmd3Rm9BVVRPVXFmVWQ5dTNTdWdPa2R1engrUFQwSVRMRXdiUVIEVllwT0JCWUUVGSjZvTWxuY
kwyd2lLcm9pDQpvZGt4OXNsBxNN1ZNQTBHQ1Nxr1NjYjNEUUVQC3dVQEU0SUNBUUNkeXE1cTUvbW5UeEVXQU9uQzlwWES5Y
Wt2DQpWYWo2bk1QeVN3d1JvSFJxWVBIUFU1aXhydlNjdNrm1gvZdHVjdzQ3FYMXVsYVhvTndjaDlqUVMrcTkwaEFYDQpKZlpXVX
BEMTRsdjlnWmFPcVBUKzFKU2F3bER1ZkFPMTfyUm5OSihHQlZNBtdNNGJNd1BubTBnRmFIVFozUlpiDQpKUExai9xWHRBnnp0Szd
mUC8xMTNvWVJpMUxQUkNoQIRTYTc5Y0ZOZWpra1cvcnhZKzdIN1NXUGtHeWpnM2JhDQpBTmdSZlk3T0xnd2J2bFp4V0oyWkNjZG
FCSUVreDR0VWRYakxBdU9SK3dKSnE3T1kyZ3JVRi9ueWg4ZUNORGw4DQpQbFh5aC9L1ZxSm4xWncwN09QNY90a3pNVFlNykJoW
HZuZzNTYXZtWTRKSGtpMWIUemU2eHdyVUV0S1kz1dPDQpNkdVYS9qa0RONTNKOgXSN1lhVjI5akFNOUImN2I5VHdoemJrNU9K
VThVVKn2bGdyaWRLZFIJQ0wXZFhyUmV0DQpDY2doNnc3Nzg1a0NHVkpneE1aMVf0UGdWSnhVUVZHNDHJcEt2NVU2YjVlcEtBaGd
MNWUyWmMxYUvkaE8yQ0wzDQp6bURDdjgyaFZkR1JlV1F3NkxjQ0JscWd4UzBRSGFJR2szVmJEd3FmVjdmWDJtRWZTQmRfMVZD

```

```

L2JXYVdjMFA4DQpqbVM0NWJxUTBCVudHMXILYmg0U2FZak10TFpaWnVBNHBuRzBMWHJyT1hRaTJNdTUzVVBXSki2NFQzUTVvaH
VBDQo0K1BQQUovbmhjNHIXY1lZeWdRbE0vTGMraTdTsmFrd1hCL3pSSzJZTjFzWkprY3J6ZDFYcklwOTVzckl5TUZkdQoyTmtONUF6Z
1E1WnVQSDNvRHc9PQOKLS0tLS1FTkQgQ0VSVEIGSUNBVEUtLS0tLQOK' --header 'Content-Type: application/json' --data-raw '{
  "szolgaltatoAdatok": {
    "adoszam": "12345632243",
    "vendeglatoUzletRegSzam": "KA22004235"
  },
  "uzenetAdatok": {
    "uzenetKuldesIdeje": "2022-12-02T18:09:02.846+01:00"
  },
  "kuldoRendszerAdatok": {
    "rmsRendszerNTAKAazonosito": "Vendeg1",
    "rmsRendszerVerzioszam": "1"
  },
  "feldolgozasAzonositok": [
    {
      "feldolgozasAzonosito": "06afd0bf-976b-4717-bd24-f5c10140b312"
    }
  ]
}'

```

6.3. Categories

Main category	Subcategory	JSON field value of main category	JSON field value of subcategory
Food	breakfast	ETEL	REGGELI
Food	sandwich	ETEL	SZENDVICS
Food	starter	ETEL	ELOETEL
Food	soup	ETEL	LEVES
Food	main course	ETEL	FOETEL
Food	side dish	ETEL	KORET
Food	pickles/salad	ETEL	SAVANYUSAG_SALATA
Food	dessert	ETEL	DESSZERT
Food	snack	ETEL	SNACK
Food	main course with side dish	ETEL	FOETEL_KORETTEL
Food	food package	ETEL	ETELCSOMAG
Food	other	ETEL	EGYEB
Alcohol-free beverages produced locally	lemonade / syrup / freshly squeezed beverages	ALKMENTESITAL_HELYBEN	LIMONADE_SZORP_FAC SART
Alcohol-free beverages produced locally	alcohol-free cocktail	ALKMENTESITAL_HELYBEN	ALKOHOLMENTES_KOK TEL
Alcohol-free beverages produced locally	beverage package	ALKMENTESITAL_HELYBEN	ITALCSOMAG
Alcohol-free beverages produced locally	tea/hot chocolate	ALKMENTESITAL_HELYBEN	TEA_FORROCSOKOLADE
Alcohol-free beverages produced locally	coffee	ALKMENTESITAL_HELYBEN	KAVE
Alcohol-free beverages not produced locally	water	ALKMENTESITAL_NEM_HELYBEN	VIZ
Alcohol-free beverages not produced locally	juices	ALKMENTESITAL_NEM_HELYBEN	ROSTOS_UDITO
Alcohol-free beverages not produced locally	carbonated beverages	ALKMENTESITAL_NEM_HELYBEN	SZENSAVAS_UDITO

Alcohol-free beverages not produced locally	noncarbonated beverages	ALKMENTESITAL_NEM_HELYBEN	SZENSÁVMENTES_UDITO
Alcohol-free beverages not produced locally	beverage package	ALKMENTESITAL_NEM_HELYBEN	ITALCSOMAG
Alcoholic beverages	cocktail	ALKOHOLOSITAL	KOKTEL
Alcoholic beverages	liquor	ALKOHOLOSITAL	LIKOR
Alcoholic beverages	distillate	ALKOHOLOSITAL	PARLAT
Alcoholic beverages	beer	ALKOHOLOSITAL	SOR
Alcoholic beverages	wine	ALKOHOLOSITAL	BOR
Alcoholic beverages	champagne	ALKOHOLOSITAL	PEZSGO
Alcoholic beverages	beverage package	ALKOHOLOSITAL	ITALCSOMAG
Other	other	EGYEB	EGYEB
Other	service fee	EGYEB	SZERVIZDIJ
Other	tip	EGYEB	BORRAVALO
Other	delivery fee	EGYEB	KISZÁLLÍTÁSI_DIJ
Other	non-catering	EGYEB	NEM_VENDEGLÁTÁS
Other	environmentally friendly packaging	EGYEB	KÖRNYEZETBARÁT_CSOMAGOLÁS
Other	plastic packaging	EGYEB	MŰANYAG_CSOMAGOLÁS
Other	discount	EGYEB	KEDVEZMÉNY