**Tax Service Documentation – HDM - Integration of Fiscal Register with External (Commercial) Software**

**Version: 0.7.3**

**Table of Contents**

1. **General Provisions** ...... 4
2. **Definitions** ...... 5
3. **Activating Integration Mode** ...... 6  
   3.1 Description of Integration Window Fields ...... 6
4. **Data Exchange Description** ...... 7  
   4.1 Operating Modes ...... 7  
   4.2 Data Exchange Procedure ...... 7  
   4.3 Functions ...... 7  
   4.4 Data Exchange Format ...... 8  
   4.4.1 Request Header Format ...... 8  
   4.4.2 Response Header Format ...... 9  
   4.4.3 Request and Response Encoding ...... 9  
   4.4.4 Request and Response Format ...... 9  
   4.4.5 Sequence Number and Protection Against Replay Attacks ...... 10  
   4.5 Data Exchange Functions Description ...... 10  
   4.5.1 Get List of HDM Operators and Departments ...... 11  
   4.5.2 HDM Operator Login (Session Start) ...... 12  
   4.5.3 HDM Operator Logout (Session End) ...... 13  
   4.5.4 Print Receipt ...... 13  
   4.5.5 Print Last Receipt Copy ...... 16  
   4.5.6 Print Return Receipt ...... 18  
   4.5.7 Get Receipt Information ...... 19  
   4.5.8 Cash In, Cash Out ...... 23  
   4.6 Get Device Date and Time ...... 23  
   4.6.1 Print Receipt Template ...... 24  
   4.6.2 Print HDM Report ...... 24  
   4.6.3 Set Receipt Header and Footer ...... 25  
   4.6.4 Set Receipt Header Logo ...... 26  
   4.7 HDM Device Synchronization ...... 27  
   4.8 Get List of Payment Systems Available on the HDM Device ...... 27  
   4.9 Check eMark Code ...... 28  
   4.10 Error Codes ...... 29

**1. General Provisions**

The integration of the Fiscal Register (HDM) with Commercial Software (CS) enables the optimization of functions performed by the HDM. As a result of integration, the accounting of sold goods will be done through the CS, and the CS will automatically send fiscal data to the HDM. Integration also allows printing the HDM receipt both through the HDM device itself and through other external printers connected to the CS.

Data exchange between the CS and the HDM is carried out over the network using the TCP protocol. When working in automatic mode, the HDM must be connected to a special internal corporate network via WiFi or Ethernet (using a USB to Ethernet converter).

**2. Definitions**

| **Term** | **Definition** |
| --- | --- |
| HDM | Fiscal Register Machine (ՀԴՄ - Հսկիչ-Դրամարկղային Մեքենա) |
| CS | Commercial Software (ԱԾ - Առևտրային Ծրագիր) |
| JSON | Standard: <http://www.json.org/> |
| TCP | Standard: <http://hy.wikipedia.org/wiki/TCP> |

**3. Activating Integration Mode**

The HDM device can work either in standard mode or in integration mode with the CS simultaneously.

To activate the integration mode, it is necessary to log in to the HDM as an Administrator or Super-Administrator and press the [External Prog. Management] button. The view of the integration screen is shown in Figure 1.

**Figure 1 – Activation of Integration Mode**

[Image Placeholder: Screen showing integration settings with fields described below]

**3.1 Description of Integration Window Fields:**

1. **Activate connection with external program** – By checking this field, the device switches from Standard mode to Integration mode.
2. **Print receipt with HDM printer** – Upon receiving data from the CS, the HDM receipt will be printed via the HDM device's printer. If this field is not checked, when performing a product sale, the CS sends fiscal data to the HDM, the HDM generates a fiscal verification number and sends it back to the CS, enabling the latter to print the HDM receipt using another printer device. If the field is not checked, the CS must print the HDM receipt using another printer. When printing with an external printer, the CS must print the following information received from the HDM program on the HDM receipt: HDM Registration Number (RN), Receipt Fiscal Number, Receipt Sequential Number (RSN), Receipt issue date (date, hour, minute, second), Amount (total).
3. **Automation System IP Address** – The IP address of the CS computer with which the HDM will exchange data.
4. **HDM Password** – The password through which the CS will connect to the HDM. To generate the password, press the [Generate] button.
5. **HDM IP Address** – The IP address of the HDM through which the CS will connect to the HDM.
6. **Port** – The port number of the HDM through which the CS will connect to the HDM. The port number is entered from the HDM keyboard.

**4. Data Exchange Description**

The program uses UTF-8 encoding for Armenian names. The program is designed to work with Armenian, Russian, and English languages for product data and punctuation symbols. Using symbols with other encodings may cause the program to malfunction.

**4.1 Operating Modes**  
The integration operating modes are presented in the table below.  
**Table 1 – Integration Operating Modes**

| **Code** | **Name** | **Description** |
| --- | --- | --- |
| P | Work with printer | HDM generates and saves a fiscal receipt and prints it on its own printer. |
| R | Work without printer | HDM generates and saves a fiscal receipt and returns the fiscal data of the receipt. |

**4.2 Data Exchange Procedure**  
To perform any function, the following steps must be taken:

1. Establish a physical TCP connection with the HDM according to the pre-set parameters in the HDM (IP, Port).
2. Send a standard 12-byte header for the request.
3. Send the encoded request containing the function.
4. Receive the response message header.
5. Receive the response message.
6. Send another message or break the connection.
7. The maximum request size should not exceed 50 kilobytes.

**4.3 Functions**  
The possible functions for data exchange with the HDM in integration mode are:

* Get list of HDM operators and departments.
* Operator login (session establishment).
* Operator logout (session termination).
* Print receipt (requires current session).
* Print last receipt copy (requires current session).
* Print return receipt (requires current session).
* Set receipt header and footer (requires current session).
* Set receipt header Logo (requires current session).
* Print HDM reports (requires current session).
* Get HDM receipt information (requires current session).
* Cash in, cash out (requires current session).
* Get device date and time (requires current session).
* Print receipt template (requires current session).
* HDM device synchronization (requires current session).
* Get list of payment systems available on the HDM device (requires current session).
* Check eMark code (requires current session).

**4.4 Data Exchange Format**

**4.4.1 Request Header Format**

| **Byte 1-7** | **Byte 8** | **Byte 9** | **Byte 10-11** | **Byte 12...** |
| --- | --- | --- | --- | --- |
| D5 80 D4 B4 D5 84 00 (HDM text as indicator) | 05 (Protocol version) | 01 (Function code) | 00 02 (Request length in bytes, BigEndian) | 05... (3DES encoded request) |

**Function Codes**

1. Get list of HDM operators and departments
2. Operator login (session establishment)
3. Operator logout (session termination)
4. Print receipt (requires current session)
5. Print last receipt copy (requires current session)
6. Print return receipt (requires current session)
7. Set receipt header and footer (requires current session)
8. Set receipt header Logo (requires current session)
9. Print HDM reports (requires current session)
10. Get receipt data (requires current session)
11. Cash in, cash out (requires current session)
12. Get device date and time (requires current session)
13. Print receipt template (requires current session)
14. HDM device synchronization (requires current session)
15. Get list of payment systems available on the HDM device (requires current session)
16. Check eMark code (requires current session)

**4.4.2 Response Header Format**

| **Byte 1** | **Byte 2** | **Byte 3-6** | **Byte 7-8** | **Byte 9-10** | **Byte 11...** |
| --- | --- | --- | --- | --- | --- |
| 00 | 05 (Protocol version) | 02 02 10 02 (Program version?) | 00 00 (Response code) | 00 C8 (Response length in bytes, BigEndian) | 00 00... (Fixed 3DES encoded response) |

**4.4.3 Request and Response Encoding**  
The 3DES standard is used for encoding requests, in ECB mode with PKCS7 padding. There are two keys. The first key is the first 24 bytes of the SHA-256 digest of the HDM password. The second key is generated by the HDM during the "Login" function.

Functions encoded with the first key are:

* Get list of HDM operators and departments,
* Operator login.

Functions encoded with the second key are:

* Operator logout (session termination),
* Print receipt (requires current session),
* Print last receipt copy (requires current session),
* Print return receipt (requires current session),
* Print HDM reports (requires current session),
* Get receipt data (requires current session),
* Set receipt header and footer (requires current session),
* Set receipt header Logo (requires current session),
* Cash in, cash out (requires current session),
* Get device date and time (requires current session),
* Print receipt template (requires current session),
* HDM device synchronization (requires current session),
* Get list of payment systems available on the HDM device (requires current session),
* Check eMark code (requires current session).

**4.4.4 Request and Response Format**  
The actual request and response are objects described in JSON format. Some requests and responses may not have a body. UTF-8 encoding is used for text data. The program is designed to enter product data and punctuation symbols in Armenian, Russian, and English languages. Using other UTF-8 encoded symbols may cause the program to malfunction.

**4.4.5 Sequence Number and Protection Against Replay Attacks**  
To protect against replay attacks, the following mechanism is used: Each request must contain a unique sequence number.

1. The client (CS) must generate, store, and send the sequence number with each request;
2. The server (HDM) must check the received sequence number against the last received number from this client and reject the request if it is not greater.

This mechanism prevents the HDM from processing a repeated (replayed) request.

**4.5 Data Exchange Functions Description**  
Below is a detailed description of the functions and the format of requests and responses.

**4.5.1 Get List of HDM Operators and Departments**  
**Encoding:** First key.  
**Request:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| password | String | HDM Password |
| **Request Example (JSON):** |  |  |

json

{

"password": "1234ABCD"

}

**Response:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| list | Object | Response object (optional) |
| list.c | Array of Objects | List of operator objects. The fields of each object in list.c are described below: |
| >id | Integer | Operator's unique ID |
| >name | String | Operator's name |
| >deps | Array of integers | List of department IDs available to the operator |
| list.d | Array of objects | List of department objects. The fields of each object in list.d are described below: |
| >id | Integer | Department ID |
| >name | String | Department name |
| >type | Integer | Department tax rate type\* |

*\*Department Type IDs*

| **ID** | **Department Type** |
| --- | --- |
| 1 | VAT 20% |
| 2 | VAT 0% |
| 3 | Without VAT (Roaming?) |
| 4 | Fixed VAT?\* |
| 5 | Patent?\* |
| 6 | Simplified Tax?\* |
| 7 | Excise Tax |

*\*Needs verification from official sources.*

**Response Example:**

json

{

"c": [

{

"id": 2,

"name": "operator",

"deps": [1, 2]

}

],

"d": [

{

"id": 1,

"name": "Food",

"type": 1

}

]

}

**4.5.2 HDM Operator Login (Session Start)**  
**Encoding:** First key.  
**Request:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| password | String | HDM Password |
| cashier | Integer | Operator ID |
| pin | String | Operator PIN code |

**Request Example:**

json

{

"password": "1234ABCD",

"cashier": 3,

"pin": "3233"

}

**Response:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| key | String | 24-byte session key, represented as Base64 |

**Response Example:**

json

{

"key": "46AF00E.." // Truncated example

}

**4.5.3 HDM Operator Logout (Session End)**  
**Encoding:** Session key.  
**Request:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| seq | Integer | Request sequence number |

**Request Example:**

json

{

"seq": 32

}

**4.5.4 HDM Receipt Print**  
**4.5.4.1**  
**Encoding:** Session key.  
When sending a new product to the HDM, the monetary fields must have a maximum of 2 digits after the decimal point, and the quantitative fields must have 3 digits. This is explained by the fact that the smallest monetary unit is the luma, corresponding to 0.01 AMD, and for quantity, gram, milliliter, or similar measurement units are mostly used. That is, 101 grams of rice is sold as 0.101 kilograms of rice. When applying discounts, the result of any calculation must be rounded again to 2-digit accuracy, rounding up: 0.005=0.01; 0.004=0.00.

**Request:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| seq | Long | Request sequence number |
| paidAmount | Double | Amount paid in cash |
| paidAmountCard | Double | Amount paid by card (non-cash) |
| partialAmount | Double | Partial payment amount |
| prePaymentAmount | Double | Amount of prepayment used |
| mode | Integer | Receipt print mode: 1 – Simple receipt, 2 – Itemized receipt, 3 – Prepayment (other values are not accepted) |
| partnerTin | string | Buyer's TIN (Tax Identification Number). 8-digit number or null |
| dep | Integer | Department for simple receipt. Sent only in case of printing a simple receipt |
| useExtPOS | boolean | Use of another payment terminal. Relates to non-cash payment (if the value is false, the HDM activates its internal non-cash payment program) |
| PaymentSystem | Integer | Payment system code, if useExtPOS is false. Can be null if the HDM device has only one payment system or if the payment system needs to be selected on the HDM device. Codes are presented in the "Payment System Codes" table. |
| rrn | string | Transaction identifier (12 digits) |
| terminalId | string | Payment terminal identifier (8 digits) |
| eMarks | array | Array of strings containing the emark codes of products present on the receipt. • Sent only for sale receipts. • The number of characters of emark codes transferred to the HDM must be at least 29 (inclusive) and maximum 110 (inclusive). • emark codes must contain only symbols from the ASCII table, from ASCII code 33 (inclusive) to ASCII code 126 (inclusive) and ASCII code 29. The symbol " in the code must be replaced with \", the symbol \ must be replaced with \\, and ASCII code 29 must be replaced with \u001d. |
| items | Array | Receipt content (list of items). The description of each item is provided below. |
| >dep | Integer | Item department |
| >qty | double | Item quantity. Maximum of 3 digits after the decimal point is allowed. |
| >discount | Double | Discount amount. Must be formatted with maximum 2 decimal places. |
| >discountType | Integer | Discount type: Percentage, amount, etc. Percentage discount type is 1, (total\_price - %) type is 2 (? - price), (initial\_price - discount) \* quantity type (∑ - total) is 4. If discount is not applicable, this field can be null. |
| >additionalDiscount | Double | Additional discount amount. Must be formatted with maximum 2 decimal places. |
| >additionalDiscountType | Integer | Type of additional discount: Percentage or fixed amount. Percentage type is 8, fixed amount type is 16. If not applicable, this field can be null. |
| >price | double | Item unit price. Must be formatted with maximum 2 decimal places. |
| >productCode | string | Product code (Max 50 characters, cannot be empty) |
| >productName | string | Product name (Max 50 characters, cannot be empty) |
| >adgCode | string | Product ADG (Classification) code. If not using the official list from taxservice.am, use codes from the HDM+SPNUSHU SUHUSHHGSHULUP document or UNP UURUH RUUnQ-QRUIJUNGIUSHU UGBEULUUP Euipuupuudhu, selecting codes from the relevant sections. Excel files N 1406-Uu and N 875-U can be used to find ADG codes. |
| >unit | string | Unit of measurement (Max 50 characters, cannot be empty) |

**Request Example 1 (Simple mode):**

json

{

"items": null,

"eMarks": ["\*\*\*\*\*\*\*\*\*\*", "\*\*\*\*\*\*\*\*\*\*"],

"paidAmount": 2200,

"paidAmountCard": 10,

"partialAmount": 0,

"prePaymentAmount": 0,

"useExtPOS": false,

"dep": 1,

"mode": 1,

"partnerTin": null,

"seq": 1

}

**Request Example 2 (Itemized mode):**

json

{

"items": [

{

"adgCode": "0104",

"dep": 1,

"productCode": "001",

"productName": "Pepsi",

"qty": 3.0,

"unit": "litr",

"price": 1000.0,

"additionalDiscount": 500.0,

"additionalDiscountType": 16,

"discount": 10.0,

"discountType": 1

}

],

"rrn": "12345678",

"terminalId": "12345678",

"eMarks": ["\*\*\*\*\*\*\*\*\*\*\*", "\*\*\*\*\*\*\*\*\*\*\*"],

"paidAmount": 2200.0,

"paidAmountCard": 10.0,

"partialAmount": 0.0,

"prePaymentAmount": 0.0,

"useExtPOS": false,

"mode": 2,

"partnerTin": null,

"seq": 4

}

**Request Example 3 (Prepayment):**

json

{

"seq": 1,

"items": null,

"paidAmount": 3000,

"paidAmountCard": 0.0,

"partialAmount": 0.0,

"prePaymentAmount": 0.0,

"mode": 3,

"useExtPOS": false,

"rrn": "12345678",

"terminalId": "12345678",

"partnerTin": null

}

**Response:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| rseq | Long | Receipt sequence number |
| crn | String | HDM Registration Number (RN) |
| sn | String | HDM Serial Number |
| tin | String | Organization's TIN |
| taxpayer | String | Organization name |
| address | String | Organization address |
| time | Long | Receipt registration/print date and time in milliseconds (Unix epoch) |
| fiscal | String | Fiscal number |
| lottery | String | Lottery number |
| prize | Integer | 0 – No win\*, 1 – Win\*\* |
| total | Double | Total amount (precision: 2 decimal places) |
| change | Double | Change given (precision: 2 decimal places) |
| qr | String | QR code text for the receipt |
| emarksCount | string | Number of control marks (eMarks) |
| verificationNumber | string | Receipt verification number. Consists of 13 digits (generated and printed) |

*\*This message is printed on the receipt as "No win"*  
*\*\*This message is printed on the receipt as "You have won"*  
*\*\*\*Receipt lottery participation is optional*

**Response Example 4 ("Simple receipt"):**

json

{

"rseq": 179,

"crn": "31008940",

"sn": "080414503833",

"tin": "00000019",

"taxpayer": "ExampleOrg",

"address": "Yerevan, Myasnikyan 2 str.",

"time": 1490190340000.0,

"fiscal": "68287355",

"lottery": "00000002",

"prize": 0,

"total": 3000.0,

"change": 0.0,

"emarksCount": "1",

"verificationNumber": "128503",

"qr": "TIN:00000019, CRN:31008940, SERIAL:, Receipt\_ID: 365, Receipt Time:02/03/2023 2:49:44 PM, FISCAL:92543463,TOTAL\_CASH:0.0,TOTAL\_NONCASH:0.0,PREP\_USAGE:0.0,PARTIAL:0.0,TOTAL:0.0"

}

**Response Example 5 (Itemized receipt):**

json

{

"rseq": 166,

"crn": "31008940",

"sn": "080414503833",

"tin": "00000019",

"taxpayer": "ExampleOrg",

"address": "Yerevan, Myasnikyan 2 str.",

"time": 149017838000.0,

"fiscal": "92543463",

"lottery": "00000002",

"prize": 0,

"total": 2200.0,

"change": 0.00,

"emarksCount": "1",

"verificationNumber": "128503",

"qr": "TIN:00000019, CRN:31008940, SERIAL:, Receipt\_ID: 365, Receipt Time:02/03/2023 2:49:44 PM, FISCAL:92543463,TOTAL\_CASH:0.0,TOTAL\_NONCASH:0.0,PREP\_USAGE:0.0,PARTIAL:0.0,TOTAL:0.0"

}

**Response Example 6 (Prepayment):**

json

{

"rseq": 175,

"crn": "31008940",

"sn": "080414503833",

"tin": "00000019",

"taxpayer": "ExampleOrg",

"address": "Yerevan, Myasnikyan 2 str.",

"time": 1490190340000.0,

"fiscal": "68287355",

"lottery": "00000002",

"prize": 0,

"total": 3000.0,

"change": 0.0,

"qr": "TIN:00000019, CRN:31008940, SERIAL:, Receipt\_ID: 365, Receipt Time:02/03/2023 2:49:44 PM, FISCAL:92543463,TOTAL\_CASH:0.0,TOTAL\_NONCASH:0.0,PREP\_USAGE:0.0,PARTIAL:0.0,TOTAL:0.0"

}

**4.5.5 Print Last Receipt Copy**  
**Encoding:** Session key.  
**Request:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| seq | Integer | Request sequence number |

**Request Example:**

json

{

"seq": 32

}

*(Response format is the same as for 4.5.4 Print Receipt)*

**4.5.6 Print Return Receipt**  
**Encoding:** Session key.  
**Request:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| seq | Integer | Request sequence number |
| receiptID | String | ID of the receipt to return |
| crn | String | Registration Number (RN) of the HDM that printed the original receipt |

**Request Example:**

json

{

"crn": "31005178",

"receiptId": "1",

"seq": 3

}

**Response:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| rseq | Long | Return receipt sequence number |
| cid | String | Client ID? |
| *(Other fields likely similar to a standard receipt response)* |  |  |

**4.5.7 Get Receipt Information**  
**Encoding:** Session key.  
**Request:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| seq | Integer | Request sequence number |
| crn | String | Device registration number |
| returnTicketId | Integer | ID of the receipt to get info for/return? |
| cashAmountForReturn | BigDecimal | Amount to return in cash. Sent if a partial return was made. |
| cardAmountForReturn | BigDecimal | Amount to return by card. Sent if a partial return was made. |
| prePaymentAmountForReturn | BigDecimal | Amount of prepayment to return. Sent if a partial return was made. |
| rrn | string | Transaction identifier (12 digits) |
| terminalId | string | Payment terminal identifier (8 digits) |
| eMarks | Array | Array of strings containing emark codes (same rules as print receipt). Sent only for sale receipts. |
| returnItemList | array | Data of items to be returned. Sent only if a partial return of an itemized receipt occurred. Description of each returnItemList element below: |
| >rpid | Long | Item line sequential number |
| >quantity | Double | Quantity of the item in this line to return |

**Request Example 8:**

json

{

"seq": 2,

"crn": "31008940",

"returnTicketId": "205",

"eMarks": ["\*\*\*\*\*\*\*\*\*\*\*", "\*\*\*\*\*\*\*\*\*\*\*"],

"cashAmountForReturn": 1000.0,

"cardAmountForReturn": 0.0,

"rrn": "12345678",

"terminalId": "12345678",

"prePaymentAmountForReturn": 0.0,

"returnItemList": [

{

"rpid": 0,

"quantity": "1.0"

}

]

}

**Response:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| rseq | long | Receipt sequence number |
| crn | String | HDM Registration Number (RN) |
| sn | String | HDM Serial Number |
| tin | String | Organization's TIN |
| taxpayer | String | Organization name |
| address | String | Organization address |
| time | long | Original receipt registration/print date & time (Greenwich time) |
| rtime | long | Return receipt registration/print date & time (Greenwich time) |
| fiscal | String | Fiscal number |
| lottery | String | Lottery number |
| prize | int\*\*\* | 0 – No win\*, 1 – Win\*\* |
| total | double | Total amount (precision: 2 decimal places) |
| change | double | Change given (precision: 2 decimal places) |
| emarksCount | string | Number of control marks (eMarks) |
| verificationNumber | string | Receipt verification number (13 digits, generated and printed) |
| *\*This message is printed on the receipt as "No win"* |  |  |
| *\*\*This message is printed on the receipt as "You have won"* |  |  |
| *\*\*\*Receipt lottery participation is optional* |  |  |

**4.5.8 Cash In, Cash Out**  
**Encoding:** Session key.  
**Request:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| seq | int | Request sequence number |
| amount | double | Cash amount |
| isCashin | bool | True for Cash In, False for Cash Out |
| cashierid | Integer | Cashier ID |
| description | String | Description of the operation |

**Request Example:**

json

{

"seq": 1,

"amount": 5000.0,

"isCashin": true,

"description": "Cash in example"

}

**4.6 Get Device Date and Time**  
**Encoding:** Session key.  
**Request:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| seq | Integer | Request sequence number |

**Request Example:**

json

{

"seq": 2

}

**Response:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| dt | String | Date and time |

**4.6.1 Print Receipt Template**  
**Encoding:** Session key.  
**Request:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| seq | Integer | Request sequence number |

**Request Example:**

json

{

"seq": 2

}

**4.6.2 Print HDM Report**  
**Encoding:** Session key.  
**Request:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| seq | Integer | Request sequence number |
| reportType | Integer | Report type selection\* |
| deptId | Integer | Department selection. If the search is performed without selecting a specific type, this field is absent. Only one filter can be used. |
| cashierId | Integer | Cashier selection. If the search is performed without selecting a specific type, this field is absent. Only one filter can be used. |
| transactionTypeId | Integer | Payment method: cash, non-cash. If the search is performed without selecting a specific type, this field is absent. Only one filter can be used. |
| startDate | Integer | Start of the time period (Unix timestamp) |
| endDate | Integer | End of the time period (Unix timestamp) |

*\*"1" – X Report, "2" – Z Report presumably*  
**Request Examples:**

json

{

"seq": 55,

"reportType": 1,

"deptId": 18,

"startDate": 123231324,

"endDate": 123271324

}

{

"seq": 55,

"reportType": 1,

"transactionTypeId": 1,

"startDate": 123231324,

"endDate": 123271324

}

{

"seq": 55,

"reportType": 1,

"cashierId": 3,

"startDate": 123231324,

"endDate": 123271324

}

Report types and their codes are defined in the HDM software, refer to its documentation.

**4.6.3 Set Receipt Header and Footer**  
**Encoding:** Session key.  
**Request:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| seq | Integer | Request sequence number |
| headers | Array | Array of header objects |
| headers[0].align | Integer | Alignment: Values: 1-left, 2-centered, 3-right |
| headers[0].bold | Boolean | Bold |
| headers[0].fsize | Integer | Font size (1,2,3,4,5) |
| headers[0].text | String | Header text |
| ... | ... | ... (for subsequent header objects) |
| footers | Array | Array of footer objects |
| footers[0].align | Integer | Alignment: Values: 1-left, 2-centered, 3-right |
| footers[0].bold | Boolean | Bold |
| footers[0].fsize | Integer | Font size (1,2,3,4,5) |
| footers[0].text | String | Footer text |
| ... | ... | ... (for subsequent footer objects) |

**Request Example:**

json

{

"seq": 435,

"headers": [

{

"align": 1,

"bold": true,

"fsize": 2,

"text": "Welcome"

}

],

"footers": [

{

"align": 1,

"bold": true,

"fsize": 2,

"text": "..."

}

]

}

**4.6.4 Set Receipt Header (Logo)**  
**Encoding:** Session key.  
**Request:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| seq | Integer | Request sequence number |
| headerLogo | Base64String | The bytes of the header logo image encoded as a Base64String. The image must be in Bitmap format and must not contain more than 4-bit colors. |

**Request Example:**

json

{

"seq": 435,

"headerLogo": "IMAGE\_ENCODED\_AS\_BASE64STRING"

}

**4.7 Fiscal Device Synchronization**

**Encoding: Using the session key.**

**Request:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| **seq** | **Integer** | **Request sequence number** |

**Example:**

**json**

**{**

**"seq": 82**

**}**

**4.8 Obtaining the list of payment systems available in the Fiscal Device (F.D.)**

**Encoding: Using the session key.**

**Request:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| **seq** | **Integer** | **Request sequence number** |

**Example:**

**json**

**{**

**"seq": 83**

**}**

**Response:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| **PaymentSystems** | **array** | **Array of payment systems:** |
| **>code** | **Integer** | **Payment system code** |
| **>name** | **String** | **Payment system name** |

**Payment System Codes:**

| **Code** | **Name** |
| --- | --- |
| **1** | **Cash** |
| **10** | **Ameria** |
| **11** | **ArCa** |
| **12** | **Visa** |
| **13** | **MasterCard** |
| **14** | **IDram** |
| **15** | **TelCell** |
| **16** | **Ala** |
| **17** | **Easypay** |
| **18** | **MobiDram** |

**Example:**

**json**

**{**

**"PaymentSystems": [**

**{**

**"code": 1,**

**"name": "Cash"**

**},**

**{**

**"code": 13,**

**"name": "MasterCard"**

**}**

**]**

**}**

**4.9 eMark Code Verification**

**Encoding: Using the session key.**

**Request:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| **seq** | **Integer** | **Request sequence number** |
| **eMark** | **String** | **eMark code** |
|  |  | **• The number of characters in the eMark code transmitted to the F.D. must be at least 29 (inclusive) and at most 110 (inclusive)** |
|  |  | **• eMark codes must contain only symbols from the ASCII table, from ASCII code 33 (inclusive) to ASCII code 126 (inclusive) and ASCII code 29. The " symbol in the code must be replaced with \", the \ symbol must be replaced with \\, and ASCII code 29 must be replaced with \u001d.** |

**Example (JSON Request):**

**json**

**{**

**"seq": 55,**

**"eMark": "\*\*\*\*\*\*\*\*\*\*"**

**}**

**4.10 Error Codes**

| **Code** | **Name** | **Description** | **Device disconnects** |
| --- | --- | --- | --- |
| **200** | **Successful operation completion** |  | **X** |
| **500** | **F.D. internal error** | **An error occurred while processing the request** | **X** |
| **400** | **Request error** | **Invalid format, or missing required field** | **X** |
| **402** | **Unsupported protocol version** |  | **X** |
| **403** | **Access denied** | **Invalid format, or the F.D.'s IP address does not match the IP address specified in the settings** | **X** |
| **404** | **Invalid function code** | **Invalid format, or the requested function code does not exist** | **X** |
| **101** | **Login [password] error** |  | **X** |
| **102** | **Session key encoding error** |  |  |
| **103** | **Header format error** | **X** |  |
| **104** | **Request sequence number error** |  | **X** |
| **105** | **JSON formatting error** |  | **X** |
| **141** | **Last receipt record is missing** |  |  |
| **142** | **Last receipt belongs to another user** |  |  |
| **143** | **Printer general error** |  |  |
| **144** | **Printer initialization error** |  |  |
| **145** | **Printer is out of paper** |  |  |
| **Login Function Errors** |  |  |  |
| **111** | **Operator password error** | **X** |  |
| **112** | **No such operator exists** | **Three possible cases: 1. This user's role is not an operator 2. This user is not active 3. Such a user is not registered** | **X** |
| **113** | **Operator is not active** | **X** |  |
| **121** | **Incorrect user** | **X** |  |
| **Receipt Printing Function Errors** |  |  |  |
| **151** | **No such department exists** | **This error will also be returned if the operator is not assigned to the specified department** |  |
| **152** | **Amount entered is less than the total amount** |  |  |
| **153** | **Receipt amount exceeds the set threshold** |  |  |
| **154** | **Receipt amount must be a positive number** |  |  |
| **...** | **...** | **...** | **...** |
| **183** | **Invalid ADG (Commodity Code) code** |  |  |
| **184** | **Invalid prepayment refund request** |  |  |
| **185** | **Cannot refund this receipt. F.D. program synchronization is required** |  |  |
| **186** | **Incorrect amount for prepayment case** |  |  |
| **187** | **Incorrect list for prepayment case** |  |  |
| **188** | **Incorrect amounts** |  |  |
| **189** | **Incorrect rounding** |  |  |
| **190** | **Payment method is not available** |  |  |
| **191** | **For cash in/out, the amount must be greater than 0** |  |  |
| **192** | **ADG code is mandatory** |  |  |
| **193** | **Buyer's TIN format is incorrect** |  |  |
| **194** | **eMark codes should not be sent for prepayment cases** |  |  |
| **195** | **eMark code format error** |  |  |
| **196** | **Foreign country code** |  |  |

**Receipt Printing Examples**

**// Prepayment receipt print (cash)**

**json**

**{**

**"seq": 1,**

**"paidAmount": 3000,**

**"paidAmountCard": 0,**

**"partialAmount": 0,**

**"prePaymentAmount": 0,**

**"mode": 3,**

**"partnerTin": null,**

**"useExtPOS": true,**

**"items": []**

**}**

**// Prepayment receipt print (card)**

**json**

**{**

**"seq": 1,**

**"paidAmount": 0,**

**"paidAmountCard": 3000,**

**"partialAmount": 0,**

**"prePaymentAmount": 0,**

**"mode": 3,**

**"partnerTin": null,**

**"useExtPOS": true,**

**"items": []**

**}**

**// Prepayment receipt print (card + cash)**

**json**

**{**

**"seq": 1,**

**"paidAmount": 3000,**

**"paidAmountCard": 3000,**

**"partialAmount": 0,**

**"prePaymentAmount": 0,**

**"mode": 3,**

**"partnerTin": null,**

**"useExtPOS": true,**

**"items": []**

**}**

**// Products case**

**json**

**{**

**"seq": 1,**

**"paidAmount": 4000,**

**"paidAmountCard": 1000,**

**"partialAmount": 0,**

**"prePaymentAmount": 1000,**

**"mode": 2,**

**"partnerTin": null,**

**"useExtPOS": true,**

**"items": [**

**{**

**"dep": 1,**

**"qty": 3,**

**"price": 1000,**

**"productCode": "001",**

**"productName": "Coca cola",**

**"adgCode": "VM01",**

**"unit": "liter"**

**},**

**{**

**"dep": 1,**

**"qty": 3,**

**"price": 1000,**

**"productCode": "002",**

**"productName": "Fanta",**

**"adgCode": "VM02",**

**"unit": "liter"**

**}**

**]**

**}**

**// Simple case**

**json**

**{**

**"seq": 1,**

**"paidAmount": 4000,**

**"paidAmountCard": 1000,**

**"partialAmount": 0,**

**"prePaymentAmount": 1000,**

**"mode": 1,**

**"useExtPOS": true,**

**"items": null,**

**"dep": 1,**

**"partnerTin": null**

**}**

**// Products with discounts**

**json**

**{**

**"seq": 1,**

**"paidAmount": 52000,**

**"paidAmountCard": 0,**

**"partialAmount": 0,**

**"prePaymentAmount": 0,**

**"mode": 2,**

**"useExtPOS": true,**

**"items": [**

**{**

**"dep": 1,**

**"qty": 1,**

**"price": 65000,**

**"productCode": "001",**

**"discount": 60,**

**"discountType": 1,**

**"productName": "Coca cola",**

**"adgCode": "VM01",**

**"unit": "liter"**

**},**

**{**

**"dep": 1,**

**"qty": 1,**

**"price": 65000,**

**"productCode": "002",**

**"discount": 60,**

**"discountType": 1,**

**"productName": "Fanta",**

**"adgCode": "VM02",**

**"unit": "liter"**

**}**

**]**

**}**