Simple Fiscal Server API (1.0.0)

Download OpenAPI specification: Download

OpenAPI description of the local Fiscal Printer Service HTTP API. The spec aims to describe the current runtime behavior first, including legacy or transitional response shapes where the backend is not yet normalized.

Error handling model:

- Use HTTP status for transport-level outcome and broad error class (`4xx` vs `5xx`).
- Use JSON body field `status` for stable, machine-readable application semantics.
- For `status` code ranges and exact meanings, see `#/components/schemas/StatusCode`.

commercial: sales

Fiscal receipt creation, storno, duplicate, non-fiscal print, and cash operations.

Creates a new sell entry

Creates a sell entry. Items can have a **correctionQuantity** field to indicate correction lines that will be printed accordingly. The endpoint requires VAT configuration to be present; otherwise it returns `400` with status `1003`.

Invoice prerequisite: When `invoiceData` is present, a valid invoice range must be configured first via `POST /invoice-range`. Without a configured range the request fails with `"invalid invoice range"`.

Operator identity: The request may include `operator: { id, name }` or `operatorId`, but not both. When `operator` is supplied, it is stored in the receipt and the operator table is created/renamed as needed. When `operatorId` is supplied, the operator must already exist and is loaded from the local operator table; otherwise the request fails with `400`. If both fields are supplied, the request fails with `400`. If both fields are omitted, legacy behavior is preserved.

Application status codes (JSON `status`): `0` success, `1101` VAT not configured, `1102` sell receipt failed, `1110` invalid `openDrawerAfterPrint`. See `#/components/schemas/StatusCode` for full

mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

- `virtual` — boolean
  Indicates if the receipt is virtual.

REQUEST BODY SCHEMA: application/json
required

- `items` > required — Array of SellItem (object) or ArticleItem (object) or FuelItem (object) or CommentItem (object) or BarcodeItem (object)
  Receipt line items. At least one item is required.
- `footerItems` > — Array of CommentItem (object) or BarcodeItem (object)
  Optional footer lines printed after the main items.
- `operator` > — object (OperatorRef)
  Minimal operator identity stored in receipt-producing requests. If a receipt request sends this object, do not also send `operatorId`.
- `operatorId` — integer >= 1
  Optional existing operator id to load and store in the sale receipt. Do not send together with `operator`.
- `payment` > required — object or object (PaymentOption)
  Payment definition for a sale. Exactly one of `paymentType` or `payments` must be provided.
- `invoiceData` > — object (InvoiceData)
  Invoice data for a sell invoice or for a storno credit note. When used on `/storno`, `NInvoice` is the newly assigned credit note number.
- `usnNumber` — integer
  Unique sales number (USN)
- `receipt_ext_format` — boolean
  Per-request override for extended receipt format (separate rows for quantity, price, total). Ignored for invoices — those always use extended format.
- `openDrawerAfterPrint` — boolean or null
  Default: `false`

`true` opens the cash drawer after the receipt is printed. `false`, `null` or no value opens nothing. Any other value is rejected with status `1110`. Only a successful sale opens the drawer, and a virtual sale (`?virtual=true`) never does. A drawer error after printing is logged, and the sale still returns success. The field is not stored in the journal.

receipt_print_vat

boolean

Per-request override for VAT breakdown printing. Ignored for invoices — those always print VAT.

Responses

- **200** Sell entry created successfully
- **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.
- **404** Not found
- **422** Request body is structurally valid but fails local business validation.
- **500** Unexpected server-side failure.

POST /sell

Request samples

Payload

Content type

application/json

Example

Minimal sell with basic items only

Copy Expand all Collapse all

```
{
  - "items": [
      + { … },
      + { … }
    ],
  - "operator": {
        "id": 1,
        "name": "Иван Иванов"
    },
  - "payment": {
        "paymentType": "CASH"
    }
}
```


Response samples

200 | 400 | 404 | 422 | 500

Content type
application/json

Copy


```
{
    "id": 12,
    "timestamp": "2026-03-10T15:35:39",
    "type": 2,
    "sha1": "0284765de7924ea507ad2bc178f506f5dda55ea7",
    "duplicate_of": 0,
    "send_to_nra_tid": -1,
    "zid": 1,
    "is_virtual": true
}
```


Creates a new storno entry

Creates a storno receipt. The endpoint requires VAT configuration to be present; otherwise it returns `400` with status `1003`.

Credit note prerequisite: When `invoiceData` is present, the storno is issued as a credit note and a valid invoice range must be configured first via `POST /invoice-range`. The service assigns `invoiceData.NInvoice` from the current invoice range value. The original invoice number remains in `stornoData.NInvoice`.

Operator identity: The request may include `operator: { id, name }` or `operatorId`, but not both. When `operator` is supplied, it is stored in the receipt and the operator table is created/renamed as needed. When `operatorId` is supplied, the operator must already exist and is loaded from the local operator table; otherwise the request fails with `400`. If both fields are supplied, the request fails with `400`. If both fields are omitted, legacy behavior is preserved.

Application status codes (JSON `status`): `0` success, `1101` VAT not configured, `1104` storno receipt failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

- `virtual` — boolean
  Indicates if the receipt is virtual.

REQUEST BODY SCHEMA: application/json
required

- `items` > required — Array of SellItem (object) or ArticleItem (object) or FuelItem (object) or CommentItem (object) or BarcodeItem (object)
  Storno line items. At least one item is required.
- `footerItems` > — Array of CommentItem (object) or BarcodeItem (object)
  Optional footer lines printed after the main items.
- `operator` > — object (OperatorRef)
  Minimal operator identity stored in receipt-producing requests. If a receipt request sends this object, do not also send `operatorId`.
- `operatorId` — integer >= 1
  Optional existing operator id to load and store in the storno receipt. Do not send together with `operator`.
- `stornoData` > required — object (StornoData)
  Metadata describing the original document that is being stornoed.
- `invoiceData` > — object (InvoiceData)
  Invoice data for a sell invoice or for a storno credit note. When used on `/storno`, `NInvoice` is the newly assigned credit note number.

usnNumber
integer
Unique sales number (USN)

receipt_ext_format
boolean
Per-request override for extended receipt format (separate rows for quantity, price, total). Ignored for credit notes — those always use extended format.

receipt_print_vat
boolean
Per-request override for VAT breakdown printing. Ignored for credit notes — those always print VAT.

Responses

> **200** Storno entry created successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Not found

> **422** Request body is structurally valid but fails local business validation.

> **500** Unexpected server-side failure.

POST /storno

Request samples

Payload

Content type
application/json

Example
Minimal storno with basic items only

Copy Expand all Collapse all

```
{
  - "items": [
      + { … },
      + { … }
    ],
  - "operator": {
        "id": 1,
        "name": "Иван Иванов"
    },
  - "stornoData": {
        "RStorno": 0,
        "XD": "2026-04-20T10:30:00",
        "ChN": 123,
        "FMIN": "12345678"
    }
}
```


Response samples

200 | 400 | 404 | 422 | 500

Content type
application/json

Copy


```
{
    "id": 12,
    "timestamp": "2026-03-10T15:35:39",
    "type": 2,
    "sha1": "0284765de7924ea507ad2bc178f506f5dda55ea7",
    "duplicate_of": 0,
    "send_to_nra_tid": -1,
    "zid": 1,
    "is_virtual": true
}
```


Creates a duplicate of the last fiscal receipt

Creates a duplicate of the last fiscal receipt. The endpoint requires VAT configuration to be present; otherwise it returns `400` with status `1003`.

Application status codes (JSON `status`): `0` success, `1101` VAT not configured, `1103` duplicate receipt failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

virtual	boolean Indicates if the receipt is virtual.

Responses

> **200** Duplicate entry created successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Not found

> **422** Request body is structurally valid but fails local business validation.

> **500** Unexpected server-side failure.

POST /duplicate

Response samples

200 | 400 | 404 | 422 | 500

Content type
application/json

Copy

```
{
    "id": 12,
    "timestamp": "2026-03-10T15:35:39",
```

```
    "type": 2,
    "sha1": "0284765de7924ea507ad2bc178f506f5dda55ea7",
    "duplicate_of": 0,
    "send_to_nra_tid": -1,
    "zid": 1,
    "is_virtual": true
}
```


Cash in or cash out operation

Accepts a cashIn or cashOut operation with a real value and returns the saved journal entry.

Operator identity: The request may include `operator: { id, name }` or `operatorId`, but not both. When supplied, the resolved operator is stored in the cash-in/out receipt. If both fields are supplied, the request fails with `400`. If both fields are omitted, legacy behavior is preserved.

Application status codes (JSON `status`): `0` success, `1001` invalid request JSON/body, `1201` cash-in/out receipt failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

- `virtual` — boolean
  Indicates if the receipt is virtual.

REQUEST BODY SCHEMA: application/json
required

One of **CashInRequest** | CashOutRequest

- `cashIn` required — number <double>
  Amount of cash to add.
- `operator` > — object (OperatorRef)
  Minimal operator identity stored in receipt-producing requests. If a receipt request sends this object, do not also send `operatorId`.
- `operatorId` — integer >= 1
  Optional existing operator id to load and store in the cash-in receipt. Do not send together with `operator`.

Responses

> **200** Cash in/out operation processed successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Not found

POST `/cashInOut`

Request samples

Payload

Content type
application/json

Example
Cash-in operation example

Copy

```
{
    "cashIn": 100.5
}
```

Response samples

200 | **400** | **404**

Content type
application/json

Copy

```
{
    "id": 12,
    "timestamp": "2026-03-10T15:35:39",
    "type": 2,
```

```
    "sha1": "0284765de7924ea507ad2bc178f506f5dda55ea7",
    "duplicate_of": 0,
    "send_to_nra_tid": -1,
    "zid": 1,
    "is_virtual": true
}
```


Print a non-fiscal (служебен) receipt with comment items

Prints a non-fiscal receipt using comment or barcode items and returns the saved journal entry.

Operator identity: The request may include `operator: { id, name }` or `operatorId`, but not both. When supplied, the resolved operator is stored in the non-fiscal receipt. If both fields are supplied, the request fails with `400`. If both fields are omitted, legacy behavior is preserved.

Application status codes (JSON `status`): `0` success, `1001` invalid request JSON/body, `1105` non-fiscal receipt registration/print failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

- `virtual` — boolean
  Indicates if the receipt is virtual.

REQUEST BODY SCHEMA: application/json
required

- `items` > required — Array of CommentItem (object) or BarcodeItem (object)
- `operator` > — object (OperatorRef)
  Minimal operator identity stored in receipt-producing requests. If a receipt request sends this object, do not also send `operatorId`.
- `operatorId` — integer >= 1
  Optional existing operator id to load and store in the non-fiscal receipt. Do not send together with `operator`.

Responses

> **200** Non-fiscal receipt printed successfully

> **400** Bad request

POST /print-nf

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
  - "items": [
      + { … }
    ],
  - "operator": {
        "id": 1,
        "name": "Ivan Ivanov"
    },
    "operatorId": 1
}
```


Response samples

200 400

Content type
application/json

Copy


```
{
    "id": 12,
    "timestamp": "2026-03-10T15:35:39",
```

```
    "type": 2,
    "sha1": "0284765de7924ea507ad2bc178f506f5dda55ea7",
    "duplicate_of": 0,
    "send_to_nra_tid": -1,
    "zid": 1,
    "is_virtual": true
}
```


Open the cash drawer connected to the printer

Sends an open pulse to the cash drawer. No request body is needed, and any body sent is ignored. Nothing is printed or written to the journal.

Application status codes (JSON `status`): `0` success, `1109` the printer backend could not open the drawer. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Drawer open pulse sent

> **500** The printer backend could not open the drawer

POST /open-drawer

Response samples

200 | 500

Content type
application/json

Copy

```
{
    "status": 0,
    "message": "OK"
}
```


commercial: reports

X/Z reports, operator/article/contract reports, journal reads, fiscal-memory queries, and receipt rendering.

Query fiscal memory records

Reads fiscal memory blocks by Z-report range or by date/time range. `formatted_text` returns printer-formatted plain text. `json` returns an array of fiscal-memory block records. `csv` is accepted as a query value, but the current backend still returns JSON.

Optional `operatorId` may be supplied to load an existing operator from the local operator table and store it in the printed fiscal-memory report receipt. If omitted, legacy behavior is preserved.

Application status codes (JSON `status`): `0` success, `1001` invalid query/request, `1651` read failed, `1652` text conversion failed, `1653` JSON conversion failed, `1654` print/receipt generation failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

- `virtual` — boolean
  Indicates if the receipt is virtual.
- `operatorId` — integer `>= 1`
  Optional existing operator id to load from the local operator table and store in the printed receipt. If not found, the request fails with `400`.
- `fromId` — integer
- `toId` — integer

fromDateTime	string <date-time> Inclusive start timestamp used when `fromId` and `toId` are not provided.
toDateTime	string <date-time> Inclusive end timestamp used when `fromId` and `toId` are not provided.
limit	integer `>= 1` Default: `100` Maximum number of items to return. There is no enforced upper bound; any positive integer is accepted. Omitting the parameter uses the default of 100.
offset	integer Default: `0` Number of items to skip before returning results.
resultFormat	string Enum: `"json"` `"formatted_text"` `"csv"` Requested output format. `formatted_text` returns plain text. `csv` may still be treated as `json` by legacy handlers.
detailed	boolean Whether to include detailed fiscal memory data

Responses

> **200** Fiscal memory records returned successfully

> **400** Bad request

GET /report/fiscal-memory

Response samples

200 400

Content type

application/json

Copy Expand all Collapse all


```
[
 - {
        "type": "Z Report",
        "record": { }
    }
]
```


Fetch journal entries based on filter criteria.

Fetches journal entries using id, Z-report, datetime, type, pagination, and output-format filters. `resultFormat=formatted_text` returns `text/plain`; `json` returns JSON; `csv` returns `text/csv` with KLEN-style rows only for fiscal receipts, invoices, and storno receipts.

Streaming response: all three non-binary formats are returned as HTTP chunked streaming responses. The service does not load the full matching range into memory before responding; it fetches from the database and streams to the client in chunks of up to 200 entries at a time.

Print chunking: if `limit` exceeds 200 (`JOURNAL_PRINT_CHUNK_SIZE`), the optional journal report receipt is also printed in 200-entry chunks. This allows arbitrarily large journal print requests without OOM risk.

`limit` **range**: `limit` accepts any positive integer with no upper cap (effectively up to `INT_MAX`). The default is 100 when the parameter is omitted. `limit=0` is rejected as invalid.

Optional `operatorId` may be supplied to load an existing operator from the local operator table and store it in the printed journal report receipt. If omitted, legacy behavior is preserved.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

- `virtual` — boolean
  Indicates if the receipt is virtual.

- `operatorId` — integer `>= 1`
  Optional existing operator id to load from the local operator table and store in the printed receipt. If not found, the request fails with `400`.

- `fromId` — integer

- `toId` — integer

- `fromZid` — integer
  First Z report of a Z-report range. Selects every entry booked under Z reports `fromZid`..`toZid` (the entry's `zid`). Mutually exclusive with the id range and the datetime range.

- `toZid` — integer
  Last Z report of a Z-report range. See `fromZid`.

- `fromDateTime` — string

- `toDateTime` — string

- `limit` — integer `>= 1`
  Default: `100`
  Maximum number of items to return. There is no enforced upper bound; any positive integer is accepted. Omitting the parameter uses the default of 100.

- `offset` — integer
  Default: `0`
  Number of items to skip before returning results.

- `type` — string (JournalType)
  Enum: `"JOURNAL_FISCAL_RECEIPT"` `"JOURNAL_INVOICE"` `"JOURNAL_NON_FISCAL_RECEIPT"` `"JOURNAL_DUPLICATE_RECEIPT"` `"JOURNAL_Z_REPORT"` `"JOURNAL_X_REPORT"` `"JOURNAL_OPERATORS_REPORT"` `"JOURNAL_ARTICLES_REPORT"` `"JOURNAL_VAT_CHANGES"` `"JOURNAL_CASH_IN"` `"JOURNAL_CASH_OUT"` `"JOURNAL_STORNO_RECEIPT"` `"JOURNAL_REGISTRATION"` `"JOURNAL_CHANGE_REGISTRATION"` `"JOURNAL_DEACTIVATION"` `"JOURNAL_FM_DETAILED_REPORT"` `"JOURNAL_FM_SHORT_REPORT"` `"JOURNAL_REPORT"` `"JOURNAL_CONTRACTS_REPORT"` `"JOURNAL_ALL"`
  String representation of journal types used in API requests.

- `resultFormat` — string
  Enum: `"json"` `"formatted_text"` `"csv"`
  Requested output format. `formatted_text` returns plain text. `csv` may still be treated as `json` by legacy handlers.

Responses

> **200** Successfully fetched journal entries.

> **400** Bad request

GET /journal

Response samples

200 400

Content type
application/json

Copy Expand all Collapse all

```
[
    { }
]
```

Generate and return X report

Generates an X report receipt and returns the saved report journal entry. VAT must be configured; otherwise the endpoint returns `400`.

Optional `operatorId` may be supplied to load an existing operator from the local operator table and store it in the printed X-report receipt. If omitted, legacy behavior is preserved.

Application status codes (JSON `status`): `0` success, `1202` X-report generation/print failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

⊣ `virtual` boolean
Indicates if the receipt is virtual.

operatorId

integer >= 1

Optional existing operator id to load from the local operator table and store in the printed receipt. If not found, the request fails with `400`.

departments

boolean

Include department totals in the report

articles

boolean

Include article totals in the report

Responses

> **200** X report generated successfully.

> **400** Bad request

GET /report/X_REPORT

Response samples

200 400

Content type
application/json

Copy Expand all Collapse all


```
{
    "id": 12,
    "timestamp": "2026-03-10T15:35:39",
    "type": 2,
    "sha1": "0284765de7924ea507ad2bc178f506f5dda55ea7",
    "duplicate_of": 0,
    "send_to_nra_tid": -1,
    "zid": 1,
    "is_virtual": true,
```

```
- "income_totals": {
      "accumulatedTotals": 1,
      "totalItems": 1,
      "totalAmount": 1000,
      "totalTaxAmount": 167,
      "grossAmountA": 1000,
      "grossAmountB": 0,
      "grossAmountC": 0,
      "grossAmountD": 0,
      "grossAmountE": 0,
      "grossAmountF": 0,
      "grossAmountG": 0,
      "grossAmountH": 0,
      "taxAmountA": 167,
      "taxAmountB": 0,
      "taxAmountC": 0,
      "taxAmountD": 0,
      "taxAmountE": 0,
      "taxAmountF": 0,
      "taxAmountG": 0,
      "taxAmountH": 0,
      "discountedItems": 0,
      "totalDiscountAmount": 0,
      "surchargedItems": 0,
      "totalSurchargeAmount": 0,
      "correctionItems": 0,
      "totalCorrectionAmount": 0
  },
```

```
- "storno_totals": {
      "accumulatedTotals": 1,
      "totalItems": 1,
      "totalAmount": 1000,
      "totalTaxAmount": 167,
      "grossAmountA": 1000,
      "grossAmountB": 0,
      "grossAmountC": 0,
      "grossAmountD": 0,
      "grossAmountE": 0,
      "grossAmountF": 0,
      "grossAmountG": 0,
      "grossAmountH": 0,
      "taxAmountA": 167,
      "taxAmountB": 0,
      "taxAmountC": 0,
      "taxAmountD": 0,
      "taxAmountE": 0,
      "taxAmountF": 0,
      "taxAmountG": 0,
      "taxAmountH": 0,
      "discountedItems": 0,
      "totalDiscountAmount": 0,
      "surchargedItems": 0,
      "totalSurchargeAmount": 0,
      "correctionItems": 0,
      "totalCorrectionAmount": 0
  },
- "payments": {
   + "CASH": { … },
   + "CREDIT_CARD": { … },
   + "CHECK": { … },
   + "DEBIT_CARD": { … },
   + "VOUCHER": { … },
   + "THIRD_PARTY_VOUCHER": { … },
   + "PACKAGING": { … },
   + "INTERNAL_SERVICE": { … },
   + "FAILURE": { … },
   + "BANK": { … },
   + "OTHER": { … },
   + "OTHER2": { … }
  },
```

```
  - "cashIn": {
        "count": 1,
        "amount": 1000
    },
  - "cashOut": {
        "count": 1,
        "amount": 1000
    },
  - "discounts": {
        "count": 1,
        "amount": 1000
    },
  - "corrections": {
        "count": 1,
        "amount": 1000
    },
  - "surcharges": {
        "count": 1,
        "amount": 1000
    },
  - "storno_operator_mistake": {
        "count": 1,
        "amount": 1000
    },
  - "storno_product_return": {
        "count": 1,
        "amount": 1000
    },
  - "storno_tax_change": {
        "count": 1,
        "amount": 1000
    },
    "accumulated_income_turnover": 0,
    "accumulated_income_tax": 0,
    "accumulated_storno_turnover": 0,
    "accumulated_storno_tax": 0,
    "report_id": 0,
    "klen_id": 0,
    "fm_block_index": 0,
    "last_fiscal_receipt_id": 0,
    "last_z_report_date": 0
}
```

Generate and return Z report

Generates a Z report receipt and returns the saved report journal entry. VAT must be configured; otherwise the endpoint returns `400`. For petrol station devices (`FDType=31`) the request body is mandatory and must provide terminal levels and nozzle counters so the backend can update petrol-station state before closing the report. For all other device types the petrol-station counters body must not be sent.

The optional request body may include `operator: { id, name }` or `operatorId`, but not both. When supplied, the resolved operator is stored in the Z-report receipt. If both fields are omitted, legacy behavior is preserved.

Application status codes (JSON `status`): `0` success, `1001` invalid request JSON/body, `1206` petrol-station counters update failed, `1203` Z-report generation/print failed, `1655` no free fiscal-memory slots for Z-report. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

- `virtual` — boolean
  Indicates if the receipt is virtual.
- `operatorId` — integer `>= 1`
  Optional existing operator id to load from the local operator table and store in the printed receipt. If not found, the request fails with `400`.
- `automatic` — boolean
  Internal-use flag. When `true`, the printed Z report includes the extra automatic-generation marker above the standard title.
- `departments` — boolean
  Include department totals in the report
- `articles` — boolean
  Include article totals in the report

REQUEST BODY SCHEMA: application/json
optional

- `terminals` > — Array of objects (TerminalLevels)
  Terminal level readings keyed by terminal number.

nozzles >

Array of objects (NozzleCounterUpdate)

Nozzle counter readings keyed by nozzle number.

operator >

object (OperatorRef)

Minimal operator identity stored in receipt-producing requests. If a receipt request sends this object, do not also send `operatorId`.

operatorId

integer `>= 1`

Optional existing operator id to load and store in the Z-report receipt. Do not send together with `operator`.

Responses

> **200** Z report generated successfully.

> **400** Bad request

POST `/report/Z_REPORT`

Request samples

Payload

Content type

application/json

Example

Non-petrol-station device

Do not send a request body for devices other than `FDType=31`. This empty object is shown only to illustrate that no petrol-station counters are needed.

Copy

```
{ }
```

Response samples

200 **400**

Content type

application/json

Copy Expand all Collapse all


```
{
   "id": 12,
   "timestamp": "2026-03-10T15:35:39",
   "type": 2,
   "sha1": "0284765de7924ea507ad2bc178f506f5dda55ea7",
   "duplicate_of": 0,
   "send_to_nra_tid": -1,
   "zid": 1,
   "is_virtual": true,
 - "income_totals": {
      "accumulatedTotals": 1,
      "totalItems": 1,
      "totalAmount": 1000,
      "totalTaxAmount": 167,
      "grossAmountA": 1000,
      "grossAmountB": 0,
      "grossAmountC": 0,
      "grossAmountD": 0,
      "grossAmountE": 0,
      "grossAmountF": 0,
      "grossAmountG": 0,
      "grossAmountH": 0,
      "taxAmountA": 167,
      "taxAmountB": 0,
      "taxAmountC": 0,
      "taxAmountD": 0,
      "taxAmountE": 0,
      "taxAmountF": 0,
      "taxAmountG": 0,
      "taxAmountH": 0,
      "discountedItems": 0,
      "totalDiscountAmount": 0,
      "surchargedItems": 0,
      "totalSurchargeAmount": 0,
      "correctionItems": 0,
      "totalCorrectionAmount": 0
   },
```

```
- "storno_totals": {
      "accumulatedTotals": 1,
      "totalItems": 1,
      "totalAmount": 1000,
      "totalTaxAmount": 167,
      "grossAmountA": 1000,
      "grossAmountB": 0,
      "grossAmountC": 0,
      "grossAmountD": 0,
      "grossAmountE": 0,
      "grossAmountF": 0,
      "grossAmountG": 0,
      "grossAmountH": 0,
      "taxAmountA": 167,
      "taxAmountB": 0,
      "taxAmountC": 0,
      "taxAmountD": 0,
      "taxAmountE": 0,
      "taxAmountF": 0,
      "taxAmountG": 0,
      "taxAmountH": 0,
      "discountedItems": 0,
      "totalDiscountAmount": 0,
      "surchargedItems": 0,
      "totalSurchargeAmount": 0,
      "correctionItems": 0,
      "totalCorrectionAmount": 0
  },
- "payments": {
   + "CASH": { … },
   + "CREDIT_CARD": { … },
   + "CHECK": { … },
   + "DEBIT_CARD": { … },
   + "VOUCHER": { … },
   + "THIRD_PARTY_VOUCHER": { … },
   + "PACKAGING": { … },
   + "INTERNAL_SERVICE": { … },
   + "FAILURE": { … },
   + "BANK": { … },
   + "OTHER": { … },
   + "OTHER2": { … }
  },
```

```
  - "cashIn": {
        "count": 1,
        "amount": 1000
    },
  - "cashOut": {
        "count": 1,
        "amount": 1000
    },
  - "discounts": {
        "count": 1,
        "amount": 1000
    },
  - "corrections": {
        "count": 1,
        "amount": 1000
    },
  - "surcharges": {
        "count": 1,
        "amount": 1000
    },
  - "storno_operator_mistake": {
        "count": 1,
        "amount": 1000
    },
  - "storno_product_return": {
        "count": 1,
        "amount": 1000
    },
  - "storno_tax_change": {
        "count": 1,
        "amount": 1000
    },
    "accumulated_income_turnover": 0,
    "accumulated_income_tax": 0,
    "accumulated_storno_turnover": 0,
    "accumulated_storno_tax": 0,
    "report_id": 0,
    "klen_id": 0,
    "fm_block_index": 0,
    "last_fiscal_receipt_id": 0,
    "last_z_report_date": 0
}
```

Generate and return operators report

Generates an operators report receipt filtered by operator ID range and returns the saved journal entry.

Optional `operatorId` may be supplied to load an existing operator from the local operator table and store it in the printed operators report receipt. If omitted, legacy behavior is preserved.

Application status codes (JSON `status`): `0` success, `1204` operators report generation/print failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

- `virtual` — boolean
  Indicates if the receipt is virtual.
- `operatorId` — integer `>= 1`
  Optional existing operator id to load from the local operator table and store in the printed receipt. If not found, the request fails with `400`.
- `fromId` — integer
  Default: `-1`
  Starting operator ID (inclusive). When omitted, the backend uses `-1`.
- `toId` — integer `<= 9999`
  Default: `9999`
  Ending operator ID (inclusive)

Responses

> **200** Operators report generated successfully.

> **400** Bad request

GET /report/operators

Response samples

200 400

Content type
application/json

Copy Expand all Collapse all


```
{
    "id": 12,
    "timestamp": "2026-03-10T15:35:39",
    "type": 2,
    "sha1": "0284765de7924ea507ad2bc178f506f5dda55ea7",
    "duplicate_of": 0,
    "send_to_nra_tid": -1,
    "zid": 1,
    "is_virtual": true,
    "fromId": 1,
    "toId": 9999,
  - "operators": [
      + { … }
    ]
}
```


Generate and return articles report

Generates an articles report receipt filtered by article ID range and returns the saved journal entry.

Optional `operatorId` may be supplied to load an existing operator from the local operator table and store it in the printed articles report receipt. If omitted, legacy behavior is preserved.

Application status codes (JSON `status`): `0` success, `1205` articles report generation/print failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

virtual	boolean Indicates if the receipt is virtual.
operatorId	integer `>= 1` Optional existing operator id to load from the local operator table and store in the printed receipt. If not found, the request fails with `400`.
fromId	integer Default: `-1` Starting article ID (inclusive). When omitted, the backend uses `-1`.
toId	integer `<= 9999` Default: `9999` Ending article ID (inclusive)
soldOnly	boolean Include only sold articles in the report

Responses

> **200** Articles report generated successfully

> **400** Bad request

GET /report/articles

Response samples

200 400

Content type
application/json

Copy Expand all Collapse all


```
{
    "id": 12,
```

```
    "timestamp": "2026-03-10T15:35:39",
    "type": 2,
    "sha1": "0284765de7924ea507ad2bc178f506f5dda55ea7",
    "duplicate_of": 0,
    "send_to_nra_tid": -1,
    "zid": 1,
    "is_virtual": true,
    "fromId": 1,
    "toId": 9999,
    "soldOnly": false,
  - "articles": [
     + { … }
    ]
}
```


Generate and return service contracts report

Generates a service contracts report receipt using the current registration service-contract snapshot and returns the saved journal entry.

Application status codes (JSON `status`): `0` success, `1207` service contracts report generation/print failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

— virtual

boolean

Indicates if the receipt is virtual.

Responses

> **200** Service contracts report generated successfully

> **400** Bad request

GET /report/contracts

Response samples

200 400

Content type
application/json

Copy


```
{
    "id": 12,
    "timestamp": "2026-03-10T15:35:39",
    "type": 2,
    "sha1": "0284765de7924ea507ad2bc178f506f5dda55ea7",
    "duplicate_of": 0,
    "send_to_nra_tid": -1,
    "zid": 1,
    "is_virtual": true,
    "contractNumber": 1,
    "serviceEIK": "0007137391",
    "serviceEIKType": 0,
    "serviceContractExpiration": "2026-11-11"
}
```


Render a single virtual receipt as formatted text

Renders one **virtual** receipt as formatted plain text, producing exactly the same output as `GET /exports/klen?resultFormat=formatted_text&fromId={id}&toId={id}` for that single journal entry.

Unlike `/exports/klen`, the text is returned in a single (non-chunked) `text/plain` response and is **not** a downloadable attachment.

Only virtual receipts can be rendered. If the journal entry with the given ID is not a virtual receipt, the request is rejected with `409 Conflict`. If no journal entry with that ID exists, `404 Not Found` is returned.

This endpoint does not print anything and does not affect KLEN, receipt, printer, or fiscal state.

Application status codes (JSON `status`): `0` success, `1505` invalid ID, `2340` receipt not found, `2341` receipt is not virtual, `2342` render failure. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

id required	integer `>= 0` Journal entry ID of the virtual receipt to render.

Responses

> **200** Successfully rendered the virtual receipt as plain text.

> **400** Invalid receipt ID in the path.

> **404** No journal entry with the requested ID exists.

> **409** The journal entry exists but is not a virtual receipt.

> **500** Failed to render the receipt to text.

GET /virtual/render-receipt/{id}

Response samples

400 | 404 | 409 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "status": 1001,
    "message": "Failed to parse JSON",
```

```
  - "logs": [
        "string"
    ]
}
```

commercial: master-data

Articles, fuels, departments, and operators CRUD used during daily operation.

Get all articles

Returns a list of all simple articles only. Fuel articles are exposed separately via `/fuels`.

Application status codes (JSON `status`): `0` success, `1504` fetch failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** List of all articles

> **500** Unexpected server-side failure.

GET /articles

Response samples

200 500

Content type
application/json

Copy Expand all Collapse all


```
[
 - {
        "FTT": 101,
        "number": 1001,
        "label": "Mineral Water 0.5L",
        "taxId": 1,
        "price": 1.2,
        "stock": 10,
        "soldAmount": 3.5,
        "turnover": 25.8,
        "departmentId": 0
    }
]
```


Create simple article

Creates a simple article entry (`FTT=0`). This change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` success, `1001` invalid request JSON/body, `1502` already exists or open-report conflict, `1503` save failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

FTT	integer Fuel type code. Used for fuel records.
number	integer Simple article number. Used for non-fuel articles.

label required	string Article label/name
taxId required	integer `[1..8]` VAT group ID (1..8)
price required	number <float> Article unit price
quantity	number <float> Initial stock quantity
departmentId	integer Optional. Fuel articles only: routes this fuel to an existing department (e.g. AdBlue) instead of the fuels report. The department must already exist and its `taxId` must match this article's `taxId`, otherwise the request is rejected (`400`). Omit or `0` for a normal fuel. Ignored for simple articles.

Responses

> **200** Simple article created successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

POST /articles

Request samples

Payload

Content type
application/json

Copy


```
{
    "FTT": 101,
    "number": 1001,
    "label": "Mineral Water 0.5L",
    "taxId": 1,
    "price": 1.2,
    "quantity": 10,
    "departmentId": 0
}
```


Response samples

200 | 400 | 409 | 500

Content type
application/json

Copy


```
{
    "FTT": 101,
    "number": 1001,
    "label": "Mineral Water 0.5L",
    "taxId": 1,
    "price": 1.2,
    "stock": 10,
    "soldAmount": 3.5,
    "turnover": 25.8,
    "departmentId": 0
}
```


Get simple article by number

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

number
required

integer

Numeric path identifier for number-based resources.

Responses

> **200** Simple article found

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

GET /articles/{number}

Response samples

200 | 400 | 404

Content type
application/json

Copy


```
{
    "FTT": 101,
    "number": 1001,
    "label": "Mineral Water 0.5L",
    "taxId": 1,
    "price": 1.2,
    "stock": 10,
    "soldAmount": 3.5,
    "turnover": 25.8,
    "departmentId": 0
}
```

Update simple article by number

Updates a simple article entry identified by the path number (`FTT=0`). This change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` success, `1505` invalid ID/number, `1501` not found, `1001` invalid request JSON/body, `1503` update failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

- `number` (required) — integer
  Numeric path identifier for number-based resources.

REQUEST BODY SCHEMA: application/json
required

- `label` (required) — string
  Article label/name
- `taxId` (required) — integer `[1 .. 8]`
  VAT group ID (1..8)
- `price` (required) — number <float>
  Article unit price
- `quantity` — number <float>
  New stock quantity
- `departmentId` — integer
  Optional. Fuel articles only: routes this fuel to an existing department (e.g. AdBlue) instead of the fuels report. The department must already exist and its `taxId` must match this article's `taxId`, otherwise the request is rejected (`400`). If omitted the current value is preserved; `0` clears the routing. Ignored for simple articles.

Responses

> **200** Simple article updated successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **500** Unexpected server-side failure.

PUT /articles/{number}

Request samples

Payload

Content type
application/json

Copy

```
{
    "label": "Mineral Water 0.5L",
    "taxId": 1,
    "price": 1.2,
    "quantity": 10,
    "departmentId": 0
}
```

Response samples

200 | 400 | 404 | 500

Content type
application/json

Copy

```
{
    "FTT": 101,
    "number": 1001,
    "label": "Mineral Water 0.5L",
```

```
    "taxId": 1,
    "price": 1.2,
    "stock": 10,
    "soldAmount": 3.5,
    "turnover": 25.8,
    "departmentId": 0
}
```


Delete simple article by number

Deletes a simple article entry. This change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` success, `1505` invalid ID/number, `1501` not found, `1503` delete failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

number required	integer Numeric path identifier for number-based resources.

Responses

> **200** Simple article deleted successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **500** Unexpected server-side failure.

DELETE /articles/{number}

Response samples

200 | 400 | 404 | 500

Content type
application/json

Copy

```
{
    "status": 0,
    "message": "Hello, world!"
}
```

Get fuel article by FTT

Read-only local view of a fuel article. This GET endpoint is acceptable for inspection/debugging.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

FTT
required
integer

Responses

> **200** Fuel article found

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

GET /fuels/{FTT}

Response samples

200 | 400 | 404

Content type
application/json

Copy


```
{
    "FTT": 101,
    "number": 1001,
    "label": "Mineral Water 0.5L",
    "taxId": 1,
    "price": 1.2,
    "stock": 10,
    "soldAmount": 3.5,
    "turnover": 25.8,
    "departmentId": 0
}
```


Update fuel article by FTT

Updates the `taxId`, `price` and/or `departmentId` of an existing fuel article. Fuel articles are created automatically during `POST /registration` from the `petrol_station.terminals` payload. If `taxId` or `price` were not provided at registration time they will be stored as sentinels (`taxId=0`, `price=-1`) and **must** be set here before a sale involving this fuel can be processed. Changes to fuel structure (adding/removing terminals) must still go through `POST /change-registration` so that NRA stays in sync.

`departmentId` routes a special fuel (e.g. AdBlue, which has a different tax group and must not appear in the fuels report) to an existing department instead. The department must already exist and its `taxId` must match the fuel's `taxId`. This is local-only configuration set **only** here (not in the terminal payload) and is preserved across `POST /change-registration`. Omitting `departmentId` keeps the current value; `0` clears the routing.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

- `FTT` required — integer

REQUEST BODY SCHEMA: application/json required

- `FTT` — integer
  Fuel type code. Used for fuel records.
- `number` — integer
  Simple article number. Used for non-fuel articles.
- `label` required — string
  Article label/name
- `taxId` required — integer `[1 .. 8]`
  VAT group ID (1..8)
- `price` required — number <float>
  Article unit price
- `quantity` — number <float>
  Initial stock quantity
- `departmentId` — integer
  Optional. Fuel articles only: routes this fuel to an existing department (e.g. AdBlue) instead of the fuels report. The department must already exist and its `taxId` must match this article's `taxId`, otherwise the request is rejected (`400`). Omit or `0` for a normal fuel. Ignored for simple articles.

Responses

> **200** Fuel article updated successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

PUT /fuels/{FTT}

Request samples

Payload

Content type
application/json

Copy


```
{
    "FTT": 101,
    "number": 1001,
    "label": "Mineral Water 0.5L",
    "taxId": 1,
    "price": 1.2,
    "quantity": 10,
    "departmentId": 0
}
```


Response samples

200 400 404 409 500

Content type
application/json

Copy


```
{
    "FTT": 101,
    "number": 1001,
    "label": "Mineral Water 0.5L",
```

```
    "taxId": 1,
    "price": 1.2,
    "stock": 10,
    "soldAmount": 3.5,
    "turnover": 25.8,
    "departmentId": 0
}
```


Delete fuel article by FTT

Deletes a fuel article. Should only be used when the Z-report state is empty. Fuel structure changes (adding/removing terminals) should go through `POST /change-registration` so that NRA stays in sync.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

FTT required	integer	

Responses

> **200** Fuel article deleted successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

DELETE /fuels/{FTT}

Response samples

200 | 400 | 404 | 409 | 500

Content type
application/json

Copy

```
{
    "status": 0,
    "message": "Hello, world!"
}
```

Get all fuel articles

Read-only local view of all fuel articles. This GET endpoint is acceptable for inspection/debugging.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** List of all fuel articles

> **500** Unexpected server-side failure.

GET /fuels

Response samples

200 500

Content type
application/json

Copy Expand all Collapse all


```
[
 - {
       "FTT": 101,
       "number": 1001,
       "label": "Mineral Water 0.5L",
       "taxId": 1,
       "price": 1.2,
       "stock": 10,
       "soldAmount": 3.5,
       "turnover": 25.8,
       "departmentId": 0
   }
]
```


Create fuel article

Creates a fuel article locally. Should only be used when the Z-report state is empty. Fuel articles are normally created automatically during `POST /registration` from the `petrol_station.terminals` payload. Direct creation through this endpoint should be reserved for setup/pre-registration flows. Structural changes must still go through `POST /change-registration` so that NRA stays in sync.

An optional `departmentId` routes a special fuel (e.g. AdBlue, with a different tax group) to an existing department instead of the fuels report — see `PUT /fuels/{FTT}` for the routing rules. The department must already exist and its `taxId` must match the fuel's `taxId`.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

label required	string Article label/name
taxId required	integer `[1..8]` VAT group ID (1..8)
price required	number <float> Article unit price
quantity	number <float> New stock quantity
departmentId	integer Optional. Fuel articles only: routes this fuel to an existing department (e.g. AdBlue) instead of the fuels report. The department must already exist and its `taxId` must match this article's `taxId`, otherwise the request is rejected (`400`). If omitted the current value is preserved; `0` clears the routing. Ignored for simple articles.

Responses

> **200** Fuel article created successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

POST `/fuels`

Request samples

Payload

Content type
application/json

Copy

```
{
    "label": "Mineral Water 0.5L",
    "taxId": 1,
    "price": 1.2,
    "quantity": 10,
    "departmentId": 0
}
```

Response samples

200 | 400 | 409 | 500

Content type
application/json

Copy

```
{
    "FTT": 101,
    "number": 1001,
    "label": "Mineral Water 0.5L",
    "taxId": 1,
    "price": 1.2,
    "stock": 10,
    "soldAmount": 3.5,
    "turnover": 25.8,
    "departmentId": 0
}
```

Get all departments

Returns a list of all departments.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

active_only

string

Presence flag. If the parameter is present, only active items are returned.

Responses

> **200** List of all departments

> **500** Unexpected server-side failure.

GET /departments

Response samples

200 500

Content type
application/json

Copy Expand all Collapse all


```
[
 - {
       "id": 1,
       "label": "Food Department",
       "taxId": 1,
       "totalAmount": 1500.5,
       "soldCount": 42
   }
]
```


Create department

Creates a department with initial zeroed statistics. This change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- **id** (required)
  integer
  Department identifier
- **label** (required)
  string
  Department label/name
- **taxId** (required)
  integer [1 .. 8]
  VAT group ID (1..8)

Responses

> **200** Department created successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

POST /departments

Request samples

Payload

Content type
application/json

Copy

```
{
    "id": 1,
    "label": "Food Department",
    "taxId": 1
}
```


Response samples

200 | 400 | 409 | 500

Content type
application/json

Copy


```
{
    "id": 1,
    "label": "Food Department",
    "taxId": 1,
    "totalAmount": 1500.5,
    "soldCount": 42
}
```


Get department by number

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

number
required

integer
Numeric path identifier for number-based resources.

Responses

> **200** Department found

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

GET /departments/{number}

Response samples

200 | 400 | 404

Content type
application/json

Copy

```
{
    "id": 1,
    "label": "Food Department",
    "taxId": 1,
    "totalAmount": 1500.5,
    "soldCount": 42
}
```

Update department by number

Updates department metadata by path id. The path number is injected into the request body by the backend. This change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

number required	integer Numeric path identifier for number-based resources.

REQUEST BODY SCHEMA: application/json

required

label required	string Department label/name
taxId required	integer [1 .. 8] VAT group ID (1..8)

Responses

> **200** Department updated successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **500** Unexpected server-side failure.

PUT /departments/{number}

Request samples

Payload

Content type
application/json

Copy

```
{
    "label": "Food Department",
    "taxId": 1
}
```

Response samples

200 400 404 500

Content type
application/json

Copy


```
{
    "id": 1,
    "label": "Food Department",
    "taxId": 1,
    "totalAmount": 1500.5,
    "soldCount": 42
}
```


Delete department by number

Deletes a department. This change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

number required	integer Numeric path identifier for number-based resources.

Responses

> **200** Department deleted successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **500** Unexpected server-side failure.

DELETE /departments/{number}

Response samples

200 | 400 | 404 | 500

Content type
application/json

Copy

```
{
    "status": 0,
    "message": "Hello, world!"
}
```

Get the current invoice range

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Successfully fetched invoice range.

> **500** Unexpected server-side failure.

GET /invoice-range

Response samples

200 | 500

Content type
application/json

Copy

```
{
    "min": "0000000001",
    "max": "0000009999",
    "current": "0000000005"
}
```

Create a new invoice range

Sets the invoice range. The backend requires only `min` and `max`; `current` is derived automatically. The operation is rejected when report data already exists, so it should only be performed when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- **min** (required)
  string `^\d{10}$`
  Minimum invoice number (10 digits)
- **max** (required)
  string `^\d{10}$`
  Maximum invoice number (10 digits)
- **current**
  string `^\d{10}$`
  Optional current invoice number (10 digits)

Responses

> **200** Invoice range created successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

POST /invoice-range

Request samples

Payload

Content type
application/json

Example
Typical invoice range (current defaults to min)

Copy

```
{
    "min": "0000000001",
    "max": "0000009999"
}
```

Response samples

200 400 409 500

Content type
application/json

Copy

```
{
    "min": "0000000001",
    "max": "0000009999",
    "current": "0000000005"
}
```

Update the invoice range

Replaces the invoice range. The backend requires only `min` and `max`; `current` is derived automatically. The operation is rejected when report data already exists, so it should only be performed when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- **min** *required*
  string `^\d{10}$`
  Minimum invoice number (10 digits)
- **max** *required*
  string `^\d{10}$`
  Maximum invoice number (10 digits)
- **current**
  string `^\d{10}$`
  Optional current invoice number (10 digits)

Responses

> **200** Invoice range updated successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

PUT /invoice-range

Request samples

Payload

Content type
application/json

Example
Typical invoice range (current defaults to min)

Copy

```
{
    "min": "0000000001",
    "max": "0000009999"
}
```

Response samples

200 | 400 | 409 | 500

Content type
application/json

Copy

```
{
    "min": "0000000001",
    "max": "0000009999",
    "current": "0000000005"
}
```

Delete the invoice range

Clears the invoice range configuration. After deletion the device will reject any receipt that requires an invoice number until a new range is configured via `POST /invoice-range`.

This endpoint is called automatically on successful deregistration. It can also be called explicitly to reset the invoice range.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Invoice range deleted successfully.

> **500** Unexpected server-side failure.

DELETE /invoice-range

Response samples

200 500

Content type
application/json

Copy

```
{
    "status": 0,
    "message": "Hello, world!"
}
```

Get the current Unique Sales Number (USN)

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Successfully fetched USN.

> **400** Bad request

GET /usn

Response samples

200 400

Content type
application/json

Copy

```
{
    "value": 12345
}
```

Set the current Unique Sales Number (USN)

Sets the current USN value. The backend accepts only non-negative integer values. This change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

⊣ value required	integer <int64> `>= 0`	New non-negative USN value

Responses

> **200** USN set successfully.

> **400** Bad request

POST /usn

Request samples

Payload

Content type
application/json

Copy

```
{
    "value": 12345
}
```

Response samples

200 400

Content type
application/json

Copy

```
{
    "value": 12345
}
```

Update the current Unique Sales Number (USN)

Replaces the current USN value. The backend accepts only non-negative integer values. This change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

value required	integer <int64> >= 0 New non-negative USN value

Responses

> **200** USN updated successfully.

> **400** Bad request

PUT /usn

Request samples

Payload

Content type
application/json

Copy

```
{
    "value": 12345
}
```

Response samples

200 **400**

Content type
application/json

Copy

```
{
    "value": 12345
}
```

Disable the Unique Sales Number (USN)

Deletes the USN value (sets it as not set). This change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** USN disabled successfully.

> **400** Bad request

DELETE /usn

Response samples

200 400

Content type
application/json

Copy

```
{
    "status": "disabled"
```

```
}
```

Get all operators

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Successfully fetched operators.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **500** Unexpected server-side failure.

GET /operators

Response samples

200 | 400 | 500

Content type
application/json

Copy Expand all Collapse all

```
[
 - {
        "id": 1,
        "name": "Ivan Ivanov",
        "salesCount": 12,
        "customersCount": 5,
        "discountsCount": 2,
        "discountsAmount": 3.5,
        "surchargesCount": 1,
        "surchargesAmount": 1.2,
        "correctionsCount": 0,
        "correctionsAmount": 0,
        "turnover": 152.8
    }
]
```


Create a new operator

Creates or upserts an operator. Only `name` is validated by the handler; if `id` is omitted or zero, the backend will attempt to store that literal value. This change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

⊣ id	integer Operator id used as primary key
⊣ name required	string Operator name

Responses

> **200** Operator created successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **500** Unexpected server-side failure.

POST /operators

Request samples

Payload

Content type
application/json

Copy

```
{
    "id": 1,
    "name": "Ivan Ivanov"
}
```

Response samples

200 400 500

Content type
application/json

Copy

```
{
    "id": 1,
    "name": "Ivan Ivanov",
    "salesCount": 12,
    "customersCount": 5,
    "discountsCount": 2,
    "discountsAmount": 3.5,
    "surchargesCount": 1,
    "surchargesAmount": 1.2,
```

```
    "correctionsCount": 0,
    "correctionsAmount": 0,
    "turnover": 152.8
}
```


Get operator by id

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

id
required

integer
Numeric path identifier for id-based resources.

Responses

> **200** Successfully fetched operator.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

GET /operators/{id}

Response samples

200 | 400 | 404

Content type
application/json

Copy


```
{
    "id": 1,
```

```
    "name": "Ivan Ivanov",
    "salesCount": 12,
    "customersCount": 5,
    "discountsCount": 2,
    "discountsAmount": 3.5,
    "surchargesCount": 1,
    "surchargesAmount": 1.2,
    "correctionsCount": 0,
    "correctionsAmount": 0,
    "turnover": 152.8
}
```


Update operator by id

Updates or upserts an operator using the path id. The request body uses only the `name` field. This change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

id required	integer Numeric path identifier for id-based resources.

REQUEST BODY SCHEMA: application/json
required

name required	string New operator name

Responses

> **200** Operator updated successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **500** Unexpected server-side failure.

PUT /operators/{id}

Request samples

Payload

Content type
application/json

Copy

```
{
    "name": "Ivan Ivanov"
}
```

Response samples

200 | 400 | 404 | 500

Content type
application/json

Copy

```
{
    "id": 1,
    "name": "Ivan Ivanov",
    "salesCount": 12,
    "customersCount": 5,
    "discountsCount": 2,
    "discountsAmount": 3.5,
    "surchargesCount": 1,
    "surchargesAmount": 1.2,
    "correctionsCount": 0,
```

```
    "correctionsAmount": 0,
    "turnover": 152.8
}
```


Delete operator by id

Deletes an operator. This change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` success, `1505` invalid ID/number, `1501` not found, `1503` delete failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

id required	integer Numeric path identifier for id-based resources.

Responses

> **200** Operator deleted successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **500** Unexpected server-side failure.

DELETE /operators/{id}

Response samples

200 | 400 | 404 | 500

Content type
application/json

Copy

```
{
    "status": 0,
    "message": "Hello, world!"
}
```

Clear all operators' statistics

Resets all counters (sales, customers, discounts, surcharges, turnover) to zero for all operators. Operator records and names are preserved. This change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` success, `1503` clear/reset failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** All operator statistics cleared successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **500** Unexpected server-side failure.

POST /operators/clear

Response samples

200 400 500

Content type
application/json

Copy

```
{
    "status": 0,
    "message": "Hello, world!"
}
```

commercial: petrol-ops

Runtime petrol-station operations - terminal levels, nozzle counters, deliveries, and gauge state (with NRA sync).

Update only runtime level fields for a terminal **Deprecated**

Internal, unofficial local write endpoint. It should not be used by external integrations. In real operation, `db` entities are treated as read-only inspection data, and petrol-station changes should flow through the supported registration/change-registration process instead of direct local writes. Any local change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

number required	integer Numeric path identifier for number-based resources.

REQUEST BODY SCHEMA: application/json
required

TNo required	integer Unique terminal number

VolAct required	number <float>	Fuel volume at normal temperature
Vol15C required	number <float>	Fuel volume at 15°C
FL required	number <float>	Fuel level
Tmpr required	number <float>	Current temperature

Responses

> **200** Terminal levels updated successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **500** Unexpected server-side failure.

POST /terminals/levels/{number}

Request samples

Payload

Content type
application/json

Copy

```
{
    "TNo": 1,
    "VolAct": 0.01,
    "Vol15C": 0.22,
    "FL": 45,
```

```
    "Tmpr": 0
}
```


Response samples

200 | 400 | 404 | 500

Content type
application/json

Copy


```
{
    "TNo": 1,
    "VolAct": 0.01,
    "Vol15C": 0.22,
    "FL": 45,
    "Tmpr": 0,
    "FTNom": "27101141",
    "FTT": 101,
    "TVol": 5000,
    "IsCalib": 1,
    "FuelName": "Дизел 1",
    "taxId": 3,
    "price": 2.35,
    "CalibDate": "2012-11-01",
    "lastUpdated": 1719936000,
    "lastLevelUpdated": 1719936000
}
```


Update all runtime level fields for all terminals in one transaction **Deprecated**

Internal, unofficial local write endpoint. It should not be used by external integrations. In real operation, `db` entities are treated as read-only inspection data, and petrol-station changes should flow through the supported registration/change-registration process instead of direct local writes. Any local change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

Array [

- TNo
  required
  integer
  Unique terminal number
- VolAct
  required
  number <float>
  Fuel volume at normal temperature
- Vol15C
  required
  number <float>
  Fuel volume at 15°C
- FL
  required
  number <float>
  Fuel level
- Tmpr
  required
  number <float>
  Current temperature

]

Responses

> **200** All terminal levels updated successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **500** Unexpected server-side failure.

POST /terminals/levels

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
[
 - {
        "TNo": 1,
        "VolAct": 0.01,
        "Vol15C": 0.22,
        "FL": 45,
        "Tmpr": 0
    }
]
```


Response samples

200 400 500

Content type
application/json

Copy Expand all Collapse all


```
[
 - {
        "TNo": 1,
        "VolAct": 0.01,
        "Vol15C": 0.22,
        "FL": 45,
        "Tmpr": 0,
        "FTNom": "27101141",
        "FTT": 101,
        "TVol": 5000,
        "IsCalib": 1,
        "FuelName": "Дизел 1",
        "taxId": 3,
        "price": 2.35,
        "CalibDate": "2012-11-01",
        "lastUpdated": 1719936000,
        "lastLevelUpdated": 1719936000
    }
]
```

Update counters for all nozzles in one transaction **Deprecated**

Internal, unofficial local write endpoint. It should not be used by external integrations. In real operation, `db` entities are treated as read-only inspection data, and petrol-station changes should flow through the supported registration/change-registration process instead of direct local writes. Any local change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

```
Array [
```

⊣ NNo required	integer Nozzle number to update
⊣ CVal required	number <float> New counter value

```
]
```

Responses

> **200** All nozzle counters updated successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

POST /nozzles/counters

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
[
 - {
       "NNo": 1,
       "CVal": 456.78
   }
]
```


Response samples

200 400

Content type
application/json

Copy Expand all Collapse all


```
[
 - {
       "NNo": 1,
       "TNo": 1,
       "DNo": 1,
       "CVal": 123.45,
       "updateCounterTimestamp": 1719936000,
       "NType": 1,
       "FTT": 123
   }
]
```


Create delivery paper for oil

Creates a local oil-delivery record. The backend currently accepts this only when registration is active and the registered fiscal device type is `31` (petrol station with a level-gauge system). Inactive registration and any other `FDType` — including the non-gauge petrol station `FDType` 3 — both return `409`.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- `deliveryId` — integer
  External delivery identifier sent to NRA
- `DelNo` — integer <int32> (DelNoField) [1 .. 2147483647]
  Unique fuel delivery number, supplied by the client and forwarded to NRA as-is. Per NRA it is optional (`minOccurs=0`); when supplied it starts at 1 and increases by 1 for each delivery, with document (paper) and gauge deliveries numbered independently. The service does not validate it. Use the matching `.../last-delno` endpoint to obtain the last successfully-registered value.
- `terminal` > required — object (OilDeliveryTerminal)

Responses

> **200** Delivery paper for oil created successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

POST `/petrol-station/delivery-paper-oil`

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

```
{
    "deliveryId": 1001,
    "DelNo": 1,
  - "terminal": {
        "TNo": 1,
        "FTT": 123,
        "AddNo": "01001000000000000001",
        "AddDate": "2025-08-13",
        "AddEIK": "131063188",
        "Vol15C": 1220.7
    }
}
```


Response samples

200 | 400 | 409 | 500

Content type
application/json

Copy


```
{
    "deliveryId": 1001,
    "DelNo": 1,
    "requestJson": "{\"deliveryId\":1001,\"terminal\":{\"TNo\":1}}",
    "nraResponse": "{\"status\":0}",
    "status": -1,
    "type": 1
}
```


Get last successful DelNo for oil delivery papers

Returns the highest DelNo among successfully-registered (status 0) oil delivery papers, or 0 if none. Clients use this to compute the next DelNo to submit.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Last successful DelNo for oil delivery papers

> **500** Unexpected server-side failure.

GET /petrol-station/delivery-paper-oil/last-delno

Response samples

200 500

Content type
application/json

Copy


```
{
    "type": 1,
    "DelNo": 7
}
```


Get delivery paper for oil by id

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

id required	integer	

Responses

> **200** Delivery paper for oil found

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

GET /petrol-station/delivery-paper-oil/{id}

Response samples

200 | 400 | 404

Content type
application/json

Copy


```
{
    "deliveryId": 1001,
    "DelNo": 1,
    "requestJson": "{\"deliveryId\":1001,\"terminal\":{\"TNo\":1}}",
    "nraResponse": "{\"status\":0}",
    "status": -1,
    "type": 1
}
```


Delete delivery paper for oil by id

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

id required	integer

Responses

> **200** Delivery paper for oil deleted successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **500** Unexpected server-side failure.

DELETE /petrol-station/delivery-paper-oil/{id}

Response samples

200 | 400 | 404 | 500

Content type
application/json

Copy

```
{
    "status": 0,
    "message": "Hello, world!"
}
```

Set NRA response for delivery paper oil by id

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

id required	integer	

REQUEST BODY SCHEMA: application/json
required

property name* additional property	any	

Responses

> **200** NRA response set successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **500** Unexpected server-side failure.

PUT /petrol-station/delivery-paper-oil/nra-response/{id}

Request samples

Payload

Content type
application/json

Copy

```
{ }
```

Response samples

200 | 400 | 404 | 500

Content type
application/json

Copy


```
{
    "deliveryId": 1001,
    "DelNo": 1,
    "requestJson": "{\"deliveryId\":1001,\"terminal\":{\"TNo\":1}}",
    "nraResponse": "{\"status\":0}",
    "status": -1,
```

```
    "type": 1
}
```

Create delivery paper for gas

Creates a local gas-delivery record. The backend currently accepts this only when registration is active and the registered fiscal device type is `31` (petrol station with a level-gauge system). Inactive registration and any other `FDType` — including the non-gauge petrol station `FDType` 3 — both return `409`.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- `deliveryId` — integer
  External delivery identifier sent to NRA
- `DelNo` — integer <int32> (DelNoField) [1 .. 2147483647]
  Unique fuel delivery number, supplied by the client and forwarded to NRA as-is. Per NRA it is optional (`minOccurs=0`); when supplied it starts at 1 and increases by 1 for each delivery, with document (paper) and gauge deliveries numbered independently. The service does not validate it. Use the matching `.../last-delno` endpoint to obtain the last successfully-registered value.
- `terminal` > required — object (GasDeliveryTerminal)

Responses

> **200** Delivery paper for gas created successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

POST /petrol-station/delivery-paper-gas

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
    "deliveryId": 1002,
    "DelNo": 1,
  - "terminal": {
        "TNo": 1,
        "FTT": 123,
        "AddNo": "01001000000000000002",
        "AddDate": "2025-08-13",
        "AddEIK": "131063188",
        "D": 0.55,
        "W": 500.3
    }
}
```


Response samples

200 400 409 500

Content type
application/json

Copy


```
{
    "deliveryId": 1001,
    "DelNo": 1,
    "requestJson": "{\"deliveryId\":1001,\"terminal\":{\"TNo\":1}}",
    "nraResponse": "{\"status\":0}",
    "status": -1,
```

```
    "type": 1
}
```

Get last successful DelNo for gas delivery papers

Returns the highest DelNo among successfully-registered (status 0) gas delivery papers, or 0 if none. Clients use this to compute the next DelNo to submit.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Last successful DelNo for gas delivery papers

> **500** Unexpected server-side failure.

GET /petrol-station/delivery-paper-gas/last-delno

Response samples

200 | 500

Content type
application/json

Copy

```
{
    "type": 1,
    "DelNo": 7
}
```

Get delivery paper for gas by id

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

id required	integer	

Responses

> **200** Delivery paper for gas found

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

GET /petrol-station/delivery-paper-gas/{id}

Response samples

200 | 400 | 404

Content type
application/json

Copy


```
{
    "deliveryId": 1001,
    "DelNo": 1,
    "requestJson": "{\"deliveryId\":1001,\"terminal\":{\"TNo\":1}}",
    "nraResponse": "{\"status\":0}",
    "status": -1,
    "type": 1
}
```

Delete delivery paper for gas by id

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

id required	integer

Responses

> **200** Delivery paper for gas deleted successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **500** Unexpected server-side failure.

DELETE /petrol-station/delivery-paper-gas/{id}

Response samples

200 | 400 | 404 | 500

Content type
application/json

Copy

```
{
    "status": 0,
    "message": "Hello, world!"
}
```

Set NRA response for delivery paper gas by id

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

id required	integer

REQUEST BODY SCHEMA: application/json
required

property name* additional property	any

Responses

> **200** NRA response set successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **500** Unexpected server-side failure.

PUT /petrol-station/delivery-paper-gas/nra-response/{id}

Request samples

Payload

Content type
application/json

Copy

```
{ }
```

Response samples

200 | 400 | 404 | 500

Content type
application/json

Copy


```
{
    "deliveryId": 1001,
    "DelNo": 1,
    "requestJson": "{\"deliveryId\":1001,\"terminal\":{\"TNo\":1}}",
    "nraResponse": "{\"status\":0}",
    "status": -1,
    "type": 1
}
```


Create delivery gauge

Creates a local gauge-delivery record. The backend currently accepts this only when registration is active and the registered fiscal device type is `31` (petrol station with a level-gauge system). Inactive registration and any other `FDType` — including the non-gauge petrol station `FDType` 3 — both return `409`.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- `deliveryId` — integer
  External delivery identifier sent to NRA
- `DelNo` — integer <int32> (DelNoField) [1 .. 2147483647]
  Unique fuel delivery number, supplied by the client and forwarded to NRA as-is. Per NRA it is optional (`minOccurs=0`); when supplied it starts at 1 and increases by 1 for each delivery, with document (paper) and gauge deliveries

numbered independently. The service does not validate it. Use the matching `.../last-delno` endpoint to obtain the last successfully-registered value.

terminal > required

object (GaugeDeliveryTerminal)

Responses

> **200** Delivery gauge created successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

POST /petrol-station/delivery-gauge

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
    "deliveryId": 1003,
    "DelNo": 1,
  - "terminal": {
        "TNo": 1,
        "FTT": 123,
        "VolAct": 1500,
        "Vol15C": 1480.5
    }
}
```

Response samples

200 | 400 | 409 | 500

Content type
application/json

Copy


```
{
    "deliveryId": 1001,
    "DelNo": 1,
    "requestJson": "{\"deliveryId\":1001,\"terminal\":{\"TNo\":1}}",
    "nraResponse": "{\"status\":0}",
    "status": -1,
    "type": 1
}
```


Get last successful DelNo for gauge deliveries

Returns the highest DelNo among successfully-registered (status 0) gauge deliveries, or 0 if none. Gauge deliveries are numbered independently from paper deliveries. Clients use this to compute the next DelNo to submit.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Last successful DelNo for gauge deliveries

> **500** Unexpected server-side failure.

GET /petrol-station/delivery-gauge/last-delno

Response samples

200 | 500

Content type
application/json

Copy


```
{
    "type": 1,
    "DelNo": 7
}
```


Get delivery gauge by id

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

id required	integer

Responses

> **200** Delivery gauge found

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

GET /petrol-station/delivery-gauge/{id}

Response samples

200 400 404

Content type
application/json

Copy


```
{
    "deliveryId": 1001,
    "DelNo": 1,
    "requestJson": "{\"deliveryId\":1001,\"terminal\":{\"TNo\":1}}",
    "nraResponse": "{\"status\":0}",
    "status": -1,
    "type": 1
}
```


Delete delivery gauge by id

AUTHORIZATIONS: *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

id required	integer

Responses

200 Delivery gauge deleted successfully

400 Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

404 Requested resource was not found.

500 Unexpected server-side failure.

DELETE /petrol-station/delivery-gauge/{id}

Response samples

200 | 400 | 404 | 500

Content type
application/json

Copy

```
{
    "status": 0,
    "message": "Hello, world!"
}
```

Set NRA response for delivery gauge by id

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

id required	integer

REQUEST BODY SCHEMA: application/json
required

property name* additional property	any

Responses

> **200** NRA response set successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **500** Unexpected server-side failure.

PUT /petrol-station/delivery-gauge/nra-response/{id}

Request samples

Payload

Content type
application/json

Copy

```
{ }
```

Response samples

200 | 400 | 404 | 500

Content type
application/json

Copy


```
{
    "deliveryId": 1001,
    "DelNo": 1,
    "requestJson": "{\"deliveryId\":1001,\"terminal\":{\"TNo\":1}}",
    "nraResponse": "{\"status\":0}",
    "status": -1,
    "type": 1
}
```


Get all delivery papers that are not processed

Returns a list of all delivery papers (oil, gas, or gauge) that have not been processed yet.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** List of not processed delivery papers and registration info

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **500** Unexpected server-side failure.

GET /petrol-station/delivery-pending

Response samples

200 | 400 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
 - "deliveries": [
    + { … }
   ],
```

```
- "registration": {
    "FDType": 3,
    "EIK": "175192260",
    "EIKType": 0,
    "FDIN": "OS800150",
    "FMIN": "58800150",
    "FDRID": "161407",
    "FDCert": "864",
    "IMSI": "284050052821582",
    "MSISDN": "359892698419",
    "OPID": "1",
    "OrgName": "ЕНСИС ООД",
    "PSNum": "12421",
    "PSType": 1,
    "SEKATTE": "68134",
    "Settl": "София",
    "AEkatte": "68134-17",
    "Area": "Витоша",
    "StreetCode": "00059",
    "Street": "АЛЕЯ ФЛОЙД БЛЯК",
    "StrNo": "1",
    "Block": "1",
    "En": "1",
    "Fl": "1",
    "Ap": "1",
    "PSName": "тест-нсис",
    "SOD": "2025-08-13T14:50:17",
    "RegDate": "2025-08-13T14:50:17",
    "ServiceEIK": "175192260",
    "ServiceEIKType": 0,
    "ServiceContractExpiration": "2027-07-23T23:59:59",
  + "petrol_station": { … },
  + "operator": { … },
    "operatorId": 1,
    "CT": "2025-08-13T14:50:17",
    "Status": 0,
  + "ZRD": { … },
    "state": 1,
    "AD": "2026-03-10T12:25:30.282Z",
    "request_xml": "string",
    "raw_response_xml": "string"
  }
```

```
}
```

Create gauge state for a gauge

Creates a local gauge-state record. The backend currently accepts this only when registration is active and the registered fiscal device type is `31` (petrol station with a level-gauge system). Inactive registration and any other `FDType` — including the non-gauge petrol station `FDType` 3 — both return `409`.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- **EType** (integer, required)
  Enum: `0` `1` `2`
  Gauge state (0=fail, 1=restore, 2=info)

- **BRC** (integer, required)
  Enum: `1` `2`
  Gauge communication state. Must be 1 or 2.

Responses

> **200** Gauge state created successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

POST /petrol-station/gauge-state

Request samples

Payload

Content type
application/json

Copy

```
{
    "EType": 2,
    "BRC": 1
}
```

Response samples

200 | 400 | 409 | 500

Content type
application/json

Copy

```
{
    "id": 1,
    "timestamp": "2026-03-10 11:22:33",
    "EType": 0,
    "gaugeType": 1,
    "requestJson": "{\"EType\":0,\"BRC\":1,\"gaugeType\":1}",
    "nraResponse": "{\"status\":0}",
    "status": -1
}
```

Create gauge state for terminals

Creates a gauge-state record for the provided terminals. The request must supply each terminal's current level readings (`VolAct`, `Vol15C`, `FL`, `Tmpr`); these are persisted to the local terminal records and embedded into the generated gauge-state request reported to NRA. Every referenced terminal must already exist in the local database. The backend currently accepts this only when registration is active and the registered fiscal device type is `31` (petrol station with a level-gauge system). Inactive registration and any other `FDType` — including the non-gauge petrol station `FDType` 3 — both return `409`.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- **EType** *required*
  integer
  Enum: `0` `1` `2`
  Gauge state (0=fail, 1=restore, 2=info)

- **tnos** > *required*
  Array of objects non-empty
  Terminals whose state is being set, each carrying its current level readings

Responses

> **200** Gauge states for terminals created successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

POST /petrol-station/gauge-state/terminals

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
    "EType": 2,
  - "tnos": [
      + { … }
    ]
}
```


Response samples

200 400 409 500

Content type
application/json

Copy


```
{
    "id": 1,
    "timestamp": "2026-03-10 11:22:33",
    "EType": 0,
    "gaugeType": 1,
    "requestJson": "{\"EType\":0,\"BRC\":1,\"gaugeType\":1}",
    "nraResponse": "{\"status\":0}",
    "status": -1
}
```


Create gauge state for dispensers

Creates a gauge-state record for the provided dispensers. The request must supply the current counter (`CVal`) of every nozzle belonging to each dispenser; these are persisted to the local nozzle records and embedded into the generated gauge-state request reported to NRA. Every referenced dispenser must already have local nozzles in the database. The backend currently accepts this only when registration is active and the registered fiscal device type is `31` (petrol station with a level-gauge system). Inactive registration and any other `FDType` — including the non-gauge petrol station `FDType` 3 — both return `409`.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

⊣ EType required	integer Enum: `0` `1` `2` Gauge state (0=fail, 1=restore, 2=info)
⊣ dnos > required	Array of objects non-empty Dispensers whose state is being set, each carrying the current counters of its nozzles

Responses

> **200** Gauge states for dispensers created successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

POST /petrol-station/gauge-state/dispensers

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

```
{
   "EType": 2,
```

```
  - "dnos": [
      + { … }
    ]
}
```


Response samples

200 | 400 | 409 | 500

Content type
application/json

Copy


```
{
    "id": 1,
    "timestamp": "2026-03-10 11:22:33",
    "EType": 0,
    "gaugeType": 1,
    "requestJson": "{\"EType\":0,\"BRC\":1,\"gaugeType\":1}",
    "nraResponse": "{\"status\":0}",
    "status": -1
}
```


Get gauge state by id

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

id required	integer	

Responses

> **200** Gauge state found

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

GET /petrol-station/gauge-state/{id}

Response samples

200 | 400 | 404

Content type
application/json

Copy


```
{
    "id": 1,
    "timestamp": "2026-03-10 11:22:33",
    "EType": 0,
    "gaugeType": 1,
    "requestJson": "{\"EType\":0,\"BRC\":1,\"gaugeType\":1}",
    "nraResponse": "{\"status\":0}",
    "status": -1
}
```


Delete gauge state by id

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

id required	integer

Responses

> **200** Gauge state deleted successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **500** Unexpected server-side failure.

DELETE /petrol-station/gauge-state/{id}

Response samples

200 | 400 | 404 | 500

Content type
application/json

Copy

```
{
    "status": 0,
    "message": "Hello, world!"
}
```

Set NRA response for gauge state by id

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

id required	integer

REQUEST BODY SCHEMA: application/json
required

property name* additional property	any

Responses

> **200** NRA response set successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **500** Unexpected server-side failure.

PUT /petrol-station/gauge-state/nra-response/{id}

Request samples

Payload

Content type
application/json

Copy

```
{ }
```

Response samples

200 | 400 | 404 | 500

Content type
application/json

Copy


```
{
    "id": 1,
    "timestamp": "2026-03-10 11:22:33",
    "EType": 0,
    "gaugeType": 1,
    "requestJson": "{\"EType\":0,\"BRC\":1,\"gaugeType\":1}",
```

```
    "nraResponse": "{\"status\":0}",
    "status": -1
}
```


Get all gauge states that are not processed

Returns a list of all gauge states that have not been processed yet.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** List of not processed gauge states and registration info

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **500** Unexpected server-side failure.

GET /petrol-station/gauge-state-pending

Response samples

200 | 400 | 500

Content type
application/json

Copy Expand all Collapse all

```
{
  - "gauge_states": [
      + { … }
    ],
```

```
- "registration": {
    "FDType": 3,
    "EIK": "175192260",
    "EIKType": 0,
    "FDIN": "OS800150",
    "FMIN": "58800150",
    "FDRID": "161407",
    "FDCert": "864",
    "IMSI": "284050052821582",
    "MSISDN": "359892698419",
    "OPID": "1",
    "OrgName": "ЕНСИС ООД",
    "PSNum": "12421",
    "PSType": 1,
    "SEKATTE": "68134",
    "Settl": "София",
    "AEkatte": "68134-17",
    "Area": "Витоша",
    "StreetCode": "00059",
    "Street": "АЛЕЯ ФЛОЙД БЛЯК",
    "StrNo": "1",
    "Block": "1",
    "En": "1",
    "Fl": "1",
    "Ap": "1",
    "PSName": "тест-нсис",
    "SOD": "2025-08-13T14:50:17",
    "RegDate": "2025-08-13T14:50:17",
    "ServiceEIK": "175192260",
    "ServiceEIKType": 0,
    "ServiceContractExpiration": "2027-07-23T23:59:59",
  + "petrol_station": { … },
  + "operator": { … },
    "operatorId": 1,
    "CT": "2025-08-13T14:50:17",
    "Status": 0,
  + "ZRD": { … },
    "state": 1,
    "AD": "2026-03-10T12:25:30.282Z",
    "request_xml": "string",
    "raw_response_xml": "string"
  }
```

```
}
```

Get all pending entities (deliveries and gauge states)

Returns a list of all pending deliveries and gauge states.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** List of all pending entities, or a status wrapper when the device is not a petrol station

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **500** Unexpected server-side failure.

GET /petrol-station/all-pending

Response samples

200 | 400 | 500

Content type
application/json

Example
AllPendingResponse

Copy Expand all Collapse all

```
{
  - "deliveries": [
      + { … }
    ],
  - "gauge_states": [
      + { … }
    ],
```

```
- "registration": {
    "FDType": 3,
    "EIK": "175192260",
    "EIKType": 0,
    "FDIN": "OS800150",
    "FMIN": "58800150",
    "FDRID": "161407",
    "FDCert": "864",
    "IMSI": "284050052821582",
    "MSISDN": "359892698419",
    "OPID": "1",
    "OrgName": "ЕНСИС ООД",
    "PSNum": "12421",
    "PSType": 1,
    "SEKATTE": "68134",
    "Settl": "София",
    "AEkatte": "68134-17",
    "Area": "Витоша",
    "StreetCode": "00059",
    "Street": "АЛЕЯ ФЛОЙД БЛЯК",
    "StrNo": "1",
    "Block": "1",
    "En": "1",
    "Fl": "1",
    "Ap": "1",
    "PSName": "тест-нсис",
    "SOD": "2025-08-13T14:50:17",
    "RegDate": "2025-08-13T14:50:17",
    "ServiceEIK": "175192260",
    "ServiceEIKType": 0,
    "ServiceContractExpiration": "2027-07-23T23:59:59",
  + "petrol_station": { … },
  + "operator": { … },
    "operatorId": 1,
    "CT": "2025-08-13T14:50:17",
    "Status": 0,
  + "ZRD": { … },
    "state": 1,
    "AD": "2026-03-10T12:25:30.282Z",
    "request_xml": "string",
    "raw_response_xml": "string"
  }
```

```
}
```

commercial: config-read

Read-only access to configuration and structure needed to render the app (config, header, VAT, cliche, localization, payment labels, petrol structure, invoice range, USN).

Retrieves the configuration

Returns the effective device configuration, including read-only fields injected from startup settings. The response always includes `global_settings_update_interval_sec`, `max_fm_blocks_warning_threshold`, `unsent_receipts_max_age_hours`, `max_klen_db_size_mb`, and `klen_db_size_warning_threshold_mb`, even if they were not persisted explicitly.

Application status codes (JSON `status`): `0` success, `1504` config fetch failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Configuration retrieval was successful

> **404** Requested resource was not found.

GET /config

Response samples

200 404

Content type
application/json

Copy Expand all Collapse all


```
{
   "port": 8888,
   "bind_address": "127.0.0.1",
   "db_path": "nsys-fiscal.db",
 - "printer_config": {
      "device_path": "/dev/ttyp0",
      "baud_rate": 115200,
      "data_bits": 8,
      "parity": 0,
      "stop_bits": 1,
      "is_rtscts": 0
   },
 - "receipt_config": {
      "cut_type": "FULL",
      "delimiter_char": "-",
      "before_print_feed": 3,
      "before_cut_feed": 5,
      "encoding": "CP1251",
      "receipt_ext_format": true,
      "receipt_print_vat": true,
      "print_dark": 30,
      "print_speed": 2,
      "header_logo_enabled": true,
      "header_logo_max_height": 300,
    + "header_style": { … },
    + "body_style": { … },
    + "footer_style": { … },
    + "total_style": { … }
   },
   "printer_type": "aclas",
   "fm_backend_type": "aclas",
   "modem_status_script": "./system-scripts/get_device_info.sh --json --port=/d
   "modem_device_file": "/dev/ttyUSB0",
   "sms_save_path": "./sms-messages",
   "nra_access_url": "http://10.199.1.11:7000/fdmon3/FDMonitorPort",
```

```
  - "ip_whitelist": [
        "127.0.0.1",
        "192.168.1.50"
    ],
    "operator_label": "----",
    "footer_line1": "",
    "footer_line2": "",
    "localization_language": "bg",
    "resource_path": "/tmp/sd/fiscal-nsys/resources",
    "auto_z_report_enabled": false,
    "allow_empty_z_report": false,
    "auto_z_report_virtual": true,
    "auto_backup_enabled": true,
    "config_migration_level": 1,
    "global_settings_update_interval_sec": 1800,
    "max_fm_blocks_warning_threshold": 100,
    "max_klen_db_size_mb": 2048,
    "klen_db_size_warning_threshold_mb": 100,
  - "update_config": {
        "update_check_interval_sec": 3600,
        "update_check_retry_interval_sec": 300,
        "update_apn_check_interval_sec": 86400,
        "command_max_per_run": 100,
        "update_via_apn": "on_failure",
        "update_apn": "",
        "update_apn_route_mode": "default",
        "update_apn_server_url": "",
        "reboot_after_z_for_update": "never"
    },
    "gsm_max_z_failures": 3,
    "unsent_receipts_max_age_hours": 72
}
```


Retrieves the company information

Returns the company/header data used on receipts.

Application status codes (JSON `status`): `0` success, `1504` company/header fetch failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Company retrieval was successful

> **404** Requested resource was not found.

GET /header

Response samples

200 | 404

Content type
application/json

Copy


```
{
    "companyName": "ЕНСИС ООД",
    "companyAddress": "гр.София",
    "storeName": "ЕНСИС ООД",
    "storeAddress": "ИЛИНДЕН",
    "companyVat": "BG112412412",
    "companyNumber": "123456789"
}
```


Get the company logo as it will be printed

Returns the stored company logo: the 1-bit BMP produced by PUT /printer/logo, 576 dots wide with the picture centered. This is exactly what prints at the top of fiscal receipts and invoices. Application status codes (JSON `status`): `2370` no logo configured.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** The stored logo

> **404** Requested resource was not found.

GET /printer/logo

Response samples

404

Content type
application/json

Copy Expand all Collapse all


```
{
    "status": 1001,
    "message": "Failed to parse JSON",
  - "logs": [
        "string"
    ]
}
```


Get the company logo metadata

The stored logo's metadata without the image. Application status codes (JSON `status`): `2370` no logo configured.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Logo metadata

> **404** Requested resource was not found.

GET /printer/logo/info

Response samples

200 | 404

Content type
application/json

Copy


```
{
    "width": 576,
    "height": 120,
    "imageWidth": 320,
    "size": 8702,
    "sha1": "3F786850E387550FDAB836ED7E6DC881DE23001B",
    "updatedAt": 1791532800
}
```


Get the current cliche array

Returns the legacy cliche representation of the same data exposed by `/header`. The response contains 6 strings in this order: company name, company address, EIK, store name, store address, company VAT. Application status codes (JSON `status`): `0` success, `1504` cliche/company fetch failed, `1503` cliche conversion failure. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Successfully fetched cliche array

> **404** Requested resource was not found.

> **500** Unexpected server-side failure.

GET /cliche

Response samples

200 | 404 | 500

Content type
application/json

Copy

```
[
    "string",
    "string",
    "string",
    "string",
    "string",
    "string"
]
```

Retrieves the VAT information

Returns the currently stored VAT configuration.

Application status codes (JSON `status`): `0` success, `1504` VAT fetch failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** VAT retrieval was successful

> **404** Requested resource was not found.

GET /vat

Response samples

200 | 404

Content type
application/json

Copy Expand all Collapse all


```
{
   "taxA": 20,
   "taxB": 9,
   "taxC": 9,
   "taxD": -1,
   "taxE": -1,
   "taxF": -1,
   "taxG": -1,
   "taxH": -1,
 - "operator": {
      "id": 1,
      "name": "Ivan Ivanov"
   },
   "operatorId": 1
}
```

Retrieves the localization settings

Returns the full localization dictionary used for receipt and report labels. Internal read-only endpoint intended for inspection/debugging.

Application status codes (JSON `status`): `0` success, `1504` localization fetch failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Localization retrieval was successful

> **404** Requested resource was not found.

GET /localization

Response samples

200 | 404

Content type
application/json

Copy


```
{
    "label_total_amount": "Обща сума",
    "label_payment_cash": "В брой",
    "label_fiscal_receipt": "Фискален бон",
    "label_x_report": "X отчет",
    "label_report_turnover": "Оборот"
}
```

Retrieves custom payment labels for OTHER and OTHER2

Returns the stored labels for the alternative payment types `OTHER` and `OTHER2`.

Application status codes (JSON `status`): `0` success, `1504` payment-labels fetch failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Payment labels retrieval was successful

> **404** Requested resource was not found.

GET /payment/labels

Response samples

200 | 404

Content type
application/json

Copy

```
{
    "label_payment_other": "с други средства",
    "label_payment_other2": "с други средства 2"
}
```

Get all terminals

Read-only local view of the currently stored terminal configuration. This GET endpoint is acceptable for inspection/debugging.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Successfully fetched terminals.

> **500** Unexpected server-side failure.

GET /terminals

Response samples

200 500

Content type
application/json

Copy Expand all Collapse all

```
[
  - {
        "TNo": 1,
        "VolAct": 0.01,
        "Vol15C": 0.22,
        "FL": 45,
        "Tmpr": 0,
        "FTNom": "27101141",
        "FTT": 101,
        "TVol": 5000,
        "IsCalib": 1,
        "FuelName": "Дизел 1",
        "taxId": 3,
        "price": 2.35,
        "CalibDate": "2012-11-01",
        "lastUpdated": 1719936000,
        "lastLevelUpdated": 1719936000
    }
]
```


Get terminal by number

Read-only local view of a single terminal. This GET endpoint is acceptable for inspection/debugging.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

number required	integer Numeric path identifier for number-based resources.

Responses

> **200** Successfully fetched terminal.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

GET /terminals/{number}

Response samples

200 | 400 | 404

Content type
application/json

Copy


```
{
    "TNo": 1,
    "VolAct": 0.01,
    "Vol15C": 0.22,
    "FL": 45,
    "Tmpr": 0,
    "FTNom": "27101141",
    "FTT": 101,
    "TVol": 5000,
    "IsCalib": 1,
    "FuelName": "Дизел 1",
    "taxId": 3,
    "price": 2.35,
    "CalibDate": "2012-11-01",
    "lastUpdated": 1719936000,
    "lastLevelUpdated": 1719936000
}
```


Get all dispensers

Read-only local view of the currently stored dispenser configuration. This GET endpoint is acceptable for inspection/debugging.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Successfully fetched dispensers.

> **500** Unexpected server-side failure.

GET /dispensers

Response samples

200 500

Content type
application/json

Copy Expand all Collapse all


```
[
 - {
      "DNo": 1,
      "DNType": 1,
      "Doos": 2,
      "DModel": "Model X",
      "DType": 1
   }
]
```

Get dispenser by number

Read-only local view of a single dispenser. This GET endpoint is acceptable for inspection/debugging.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

number required	integer Numeric path identifier for number-based resources.

Responses

> **200** Successfully fetched dispenser.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

GET /dispensers/{number}

Response samples

200 | 400 | 404

Content type
application/json

Copy


```
{
    "DNo": 1,
    "DNType": 1,
```

```
    "Doos": 2,
    "DModel": "Model X",
    "DType": 1
}
```


Get all nozzles

Read-only local view of the currently stored nozzle configuration. This GET endpoint is acceptable for inspection/debugging.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

TNo
integer
Filter nozzles by terminal number.

DNo
integer
Filter nozzles by dispenser number.

Responses

> **200** Successfully fetched nozzles.

> **500** Unexpected server-side failure.

GET /nozzles

Response samples

200 500

Content type
application/json

Copy Expand all Collapse all


```
[
 - {
        "NNo": 1,
        "TNo": 1,
        "DNo": 1,
        "CVal": 123.45,
        "updateCounterTimestamp": 1719936000,
        "NType": 1,
        "FTT": 123
    }
]
```


Get nozzle by number

Read-only local view of a single nozzle. This GET endpoint is acceptable for inspection/debugging.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

number required	integer Numeric path identifier for number-based resources.

Responses

> **200** Successfully fetched nozzle.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

GET /nozzles/{number}

Response samples

200 | 400 | 404

Content type
application/json

Copy


```
{
    "NNo": 1,
    "TNo": 1,
    "DNo": 1,
    "CVal": 123.45,
    "updateCounterTimestamp": 1719936000,
    "NType": 1,
    "FTT": 123
}
```


Get the current petrol station configuration

Read-only local view of the currently stored petrol-station configuration. This GET endpoint is acceptable for inspection/debugging.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Successfully fetched petrol station configuration.

> **500** Unexpected server-side failure.

GET /petrol-station

Response samples

200 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "GNo": "P-4914",
  - "terminals": [
      + { … }
    ],
  - "dispensers": [
      + { … }
    ],
  - "nozzles": [
      + { … }
    ]
}
```


commercial: device-control

Emergency-mode toggling and firmware update-server registration key.

Enable emergency mode (override the NRA-connection blocks)

Authorizes an **emergency override** that lets the device keep issuing receipts despite the NRA-connection blocks: the overdue-unsent-receipts lock (`unsent_receipts_max_age_hours`, default 72h) and the GSM `NO_CONNECTION` lock. It deliberately does **not** override the integrity/registration locks — a foreign/uninitialized KLEN (`klenMode`), a swapped SIM (`simMode`) or a deregistered device stay blocked.

Authorization: the request `deviceKey` must equal the update-server registration key stored at registration with the update server (the `registrationKey`). If no key is stored, the request is rejected.

On a real state change the device persists the new state (so it survives a restart) and prints an "emergency mode enabled" status bon. When the device is already in emergency mode the call is a no-op: HTTP `409` with status `2333` and **no** bon is printed.

Application status codes (JSON `status`): `0` success, `2331` `deviceKey` missing, `2332` invalid `deviceKey`, `2333` already in emergency mode, `2336` invalid operator (both `operator` and `operatorId` supplied, or `operatorId` not found), `2337` invalid `comments`.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- `deviceKey` required
  string
  Must equal the update-server registration key stored at registration with the update server (the `registrationKey`).
- `operator` >
  object
  Optional operator that authorized the toggle, printed in the receipt header exactly like on every other bon. Supply either `operator` (requires `id` and `name`) or `operatorId`, but not both. Handled the same way as sell/storno/print-nf receipts.
- `operatorId`
  integer
  Optional id of an existing operator that authorized the toggle. The operator must already exist in the local operator table; its name is loaded and printed in the receipt header. Mutually exclusive with `operator`.
- `comments` >
  Array of objects
  Optional comment lines printed after the status header on the bon (e.g. the reason for toggling emergency mode). Each item uses the same `{left,`

`center, right}` COMMENT shape as sell/storno/print-nf receipts; every item must carry at least one non-empty field.

Responses

> **200** Emergency mode enabled; current device state returned

> **403** Invalid `deviceKey` (HTTP 403, status 2332)

> **409** Already in emergency mode (HTTP 409, status 2333)

> **422**
`deviceKey` is required (HTTP 422, status 2331), invalid operator (status 2336), or invalid comments (status 2337)

POST `/system/enable-emergency-mode`

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

```
{
    "deviceKey": "update-server-registration-key-1234",
  - "operator": {
        "id": 1,
        "name": "Иван"
    },
    "operatorId": 1,
  - "comments": [
      + { … },
      + { … }
    ]
}
```


Response samples

200 | 403 | 409 | 422

Content type
application/json

Copy


```
{
    "klenMode": "NORMAL",
    "simMode": "OK",
    "gsmMode": "OK",
    "emergencyMode": "OFF",
    "capacityExhausted": false,
    "deregistered": false,
    "writesBlocked": false,
    "blockReason": "NONE"
}
```


Disable emergency mode (restore normal NRA-connection blocking)

Turns off the emergency override and returns to normal NRA-connection blocking. If the device still has no NRA connection (the GSM lock or an overdue-unsent backlog is still present), receipt issuance is blocked again from this point on.

Authorization is identical to `POST /system/enable-emergency-mode`: the `deviceKey` must match the stored update-server registration key.

On a real state change the device persists the new state and prints an "emergency mode disabled" status bon. When the device is already not in emergency mode the call is a no-op: HTTP `409` with status `2334` and **no** bon is printed.

Application status codes (JSON `status`): `0` success, `2331` `deviceKey` missing, `2332` invalid `deviceKey`, `2334` already not in emergency mode, `2336` invalid operator (both `operator` and `operatorId` supplied, or `operatorId` not found), `2337` invalid `comments`.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- `deviceKey` required
  string
  Must equal the update-server registration key stored at registration with the update server (the `registrationKey`).
- `operator` >
  object
  Optional operator that authorized the toggle, printed in the receipt header exactly like on every other bon. Supply either `operator` (requires `id` and `name`) or `operatorId`, but not both. Handled the same way as sell/storno/print-nf receipts.
- `operatorId`
  integer
  Optional id of an existing operator that authorized the toggle. The operator must already exist in the local operator table; its name is loaded and printed in the receipt header. Mutually exclusive with `operator`.
- `comments` >
  Array of objects
  Optional comment lines printed after the status header on the bon (e.g. the reason for toggling emergency mode). Each item uses the same `{left, center, right}` COMMENT shape as sell/storno/print-nf receipts; every item must carry at least one non-empty field.

Responses

> **200** Emergency mode disabled; current device state returned

> **403** Invalid `deviceKey` (HTTP 403, status 2332)

> **409** Already not in emergency mode (HTTP 409, status 2334)

> **422**
`deviceKey` is required (HTTP 422, status 2331), invalid operator (status 2336), or invalid comments (status 2337)

POST /system/disable-emergency-mode

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
    "deviceKey": "update-server-registration-key-1234",
  - "operator": {
        "id": 1,
        "name": "Иван"
    },
    "operatorId": 1,
  - "comments": [
     + { … },
     + { … }
    ]
}
```


Response samples

200 403 409 422

Content type
application/json

Copy


```
{
    "klenMode": "NORMAL",
```

```
    "simMode": "OK",
    "gsmMode": "OK",
    "emergencyMode": "OFF",
    "capacityExhausted": false,
    "deregistered": false,
    "writesBlocked": false,
    "blockReason": "NONE"
}
```


Read the current emergency-mode state

Returns whether emergency mode is currently active, plus the full device-state snapshot (all lock modes, the consolidated `writesBlocked` flag and the active `blockReason`). No `deviceKey` is required — this is a read-only status endpoint.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Current emergency-mode state returned

GET /system/emergency-mode

Response samples

200

Content type
application/json

Copy Expand all Collapse all


```
{
    "emergencyMode": false,
```

```
  - "deviceState": {
        "klenMode": "NORMAL",
        "simMode": "OK",
        "gsmMode": "OK",
        "emergencyMode": "OFF",
        "capacityExhausted": false,
        "deregistered": false,
        "writesBlocked": false,
        "blockReason": "NONE"
    }
}
```


commercial: status

Health, greeting, device info, printer status, datetime read, and provisioning read.

Returns a greeting status payload

Responses

> **200** A successful response

GET /hello

Response samples

200

Content type
application/json

Copy


```
{
    "status": 0,
    "message": "Hello, world!"
}
```


Health check

Returns the same printer status payload as `/system/printer-status`. Intended as a lightweight liveness/readiness check.

Application status codes (JSON `status`): `0` success, `1907` printer status retrieval failed. See `#/components/schemas/StatusCode` for full mapping.

Responses

> **200** Health information returned successfully

> **500** Unexpected server-side failure.

GET /health

Response samples

200 500

Content type
application/json

Copy

```
{
    "can_print": true,
    "status_initizalized": true,
    "printer_drawer_connector_pin_3": true,
    "printer_is_offline": true,
    "offline_cover_open": true,
    "offline_paper_end": true,
    "has_unfinished_print_job": true,
    "offline_error": true,
    "error_recoverable_occurred": true,
    "error_autocutter_occurred": true,
    "error_unrecoverable_occurred": true,
    "error_autorecoverable_occurred": true,
    "paper_near_end": true,
    "paper_not_present": true,
    "temperature_out_of_range": true,
    "power_fail_occurred": true
}
```


What the device still owes the NRA, and how long it has got

A single read that answers whether this device is keeping up with the NRA, meant to be polled by monitoring software so it can alarm an operator before the device stops selling. It is read-only: nothing is sent and no task is touched.

It answers with:

- `sells.unsent_count` / `z_reports.unsent_count` — how many documents are still waiting right now. Sales, stornos and invoices are counted together, exactly as the overdue-unsent lock counts them.
- `sells.block` — the countdown to that lock: `deadline`, `seconds_left`, `hours_left` and whether it is already `blocked`. The deadline comes from the OLDEST waiting receipt plus `unsent_receipts_max_age_hours` (default 72), so a receipt printed at 10:00 and still unsent at 17:00 leaves 65 hours.
- `oldest_unsent` — which document is at the head of each queue: its number (`id`), shift (`zid`) and time. Informative, but it is the one that has to go through first.
- `schedule.next_send_at_str` — when each queue goes out next, Z reports included. `null` means no NRA task of that kind is active, i.e. nothing is scheduled at all.
- `restrictions` — every lock currently on the device (foreign KLEN, swapped SIM, GSM, deregistration, capacity, emergency mode), in the same shape `/system/info` uses.

Application status codes (JSON `status`): `0` success, `2105` fetch pending-info failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Successfully fetched NRA journal pending info

> **400** Bad request

GET /nra-update-tasks/nra-send-pending-info

Response samples

200 400

Content type
application/json

Copy Expand all Collapse all


```
{
    "now": 1756713600,
    "now_str": "2026-09-01T12:00:00",
  - "sells": {
        "unsent_count": 12,
        "sent_count": 3480,
      + "oldest_unsent": { … },
      + "last_sent": { … },
      + "schedule": { … },
      + "block": { … }
    },
```

```
  - "z_reports": {
        "unsent_count": 12,
        "sent_count": 3480,
      + "oldest_unsent": { … },
      + "last_sent": { … },
      + "schedule": { … },
      + "block": { … }
    },
  - "restrictions": {
        "klenMode": "NORMAL",
        "simMode": "OK",
        "gsmMode": "OK",
        "modemConnected": true,
        "emergencyMode": "ON",
        "capacityExhausted": true,
        "deregistered": true,
        "unsentSalesBlocked": true,
        "writesBlocked": true,
        "blockReason": "NONE"
    }
}
```

Get device info including modem status and registration

Returns current device, modem, fiscal-memory, and registration-related information. When registration is active, the response is enriched with stored registration fields.

Application status codes (JSON `status`): `0` success, `1904` required system/modem context unavailable. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

⊣ `extended` boolean

When true, return the extended variant of the payload.

Responses

> **200** Device info returned successfully

> **503** Service dependency is temporarily unavailable.

GET /system/info

Response samples

200 503

Content type
application/json

Copy


```
{
    "applicationState": "0: No errors, this is a dummy test info state",
    "deviceId": "unknown",
    "gsmSignalAsu": -1,
    "gsmOperator": "unknown",
    "imei": "unknown",
    "imsi": "unknown",
    "env": "INFO",
    "fiscalDocumentNumber": 12,
    "FMIN": "FM000001",
    "isRegistered": true,
    "klenNumber": "0001",
    "klenMode": "NORMAL",
    "simMode": "OK",
    "gsmMode": "OK",
    "emergencyMode": "OFF",
    "capacityExhausted": false,
    "writesBlocked": false,
    "blockReason": "NONE",
    "FDIN": "DT000001",
    "zReportNumber": 1,
    "zReportTid": "-------------",
    "version": "1.0.0",
    "pendingFirmwareUpdate": "1.0.253"
```

```
}
```

Get current provisioning data

Returns current provisioning identifiers or an informational status when they are not set yet.

Application status codes (JSON `status`): `0` success, `1990` not provisioned yet. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** A successful response

GET /provisioning

Response samples

200

Content type
application/json

Example
Provisioning

Copy

```
{
   "FDIN": "OS800150",
   "FMIN": "58800150"
}
```

Get the current printer status

Retrieves the current status of the printer.

Application status codes (JSON `status`): `0` success, `1907` printer status could not be retrieved. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Printer status retrieved successfully

> **400** Bad request

GET /system/printer-status

Response samples

200 400

Content type
application/json

Copy


```
{
    "can_print": true,
    "status_initialized": true,
    "printer_drawer_connector_pin_3": true,
    "printer_is_offline": true,
    "offline_cover_open": true,
    "offline_paper_end": true,
    "has_unfinished_print_job": true,
    "offline_error": true,
    "error_recoverable_occurred": true,
```

```
    "error_autocutter_occurred": true,
    "error_unrecoverable_occurred": true,
    "error_autorecoverable_occurred": true,
    "paper_near_end": true,
    "paper_not_present": true,
    "temperature_out_of_range": true,
    "power_fail_occurred": true
}
```


Get current system date and time

Returns the current system time as both UNIX timestamp and ISO 8601 string.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Current date and time

GET `/system/datetime`

Response samples

200

Content type
application/json

Copy

```
{
    "timestamp": 1724695200,
    "datetime": "2024-08-26T12:00:00+03:00"
}
```


service: config

Writing device configuration documents - config, header, VAT, cliche, localization, payment labels, and datetime set.

Updates the configuration

Partially updates the persisted configuration. Read-only startup fields such as `port`, `db_path`, `printer_config`, `printer_type`, `fm_backend_type`, `modem_status_script`, `modem_device_file`, `sms_save_path`, and `nra_access_url` are ignored on write and are still returned by subsequent GET requests.

Application status codes (JSON `status`): `0` success, `1001` invalid request JSON/body, `1503` config save/update failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- `receipt_config` >

  object
  The receipt configuration object

  - `ip_whitelist`

    Array of strings
    Client-IP allow-list for incoming requests, HTTP and HTTPS alike. When set to a non-empty array, only requests whose client address matches an entry are served; all others receive `403` (status `2350`). When absent or an empty array, the allow-list is disabled and every client is accepted. IPv4-mapped IPv6 clients are normalized to plain IPv4. A single comma/whitespace-separated string is also accepted. A non-empty list must

include `127.0.0.1`: the internal callers (nra_client, nra_cron, auto-Z) reach the service over loopback and are subject to the same list. For HTTPS the list is enforced by nsys-tls-proxy on the real client address before the TLS handshake — the service sees only the proxy's loopback address — so a denied HTTPS client gets the connection closed rather than a `403` body.

`operator_label`

string
Operator label for receipts

`footer_line1`

string
First footer line for receipts

`footer_line2`

string
Second footer line for receipts

`localization_language`

string
Active localization resource to load from `resource_path/localization/<language>.json`

`auto_z_report_enabled`

boolean
Enable automatic Z report generation after the service detects that a new day has started

---

`allow_empty_z_report`

boolean
Allow generating Z reports with no transactions

`auto_z_report_virtual`

boolean
Use `virtual=true` when the automatic Z report flow calls `POST /report/Z_REPORT`

`auto_backup_enabled`

boolean
Enable automatic daily database backup generation shortly after midnight

---

`config_migration_level`

integer
Level of the last one-shot config migration applied to this device. Maintained by the service at startup: any migration above this level runs once, then the level is stamped so it never runs again. Level 1 moved `receipt_config.print_dark` from the old default 2 to 30. Informational — do not set it by hand.

- `global_settings_update_interval_sec` integer
  Interval in seconds for global settings updates
- `max_fm_blocks_warning_threshold` integer
  Threshold for remaining free fiscal-memory Z-report slots. When the number of free slots drops below this value, a warning is printed as a comment inside the Z-report itself (and reprinted on copies/reports). Must be non-negative.
- `max_klen_db_size_mb` integer
  Maximum size of the KLEN (SQLite electronic journal) database file, in megabytes. Once the file reaches this size the device goes read-only (only reports and exports are allowed), exactly like a full fiscal memory. Must be positive. Defaults to `2048` (2 GB) when not set.
- `klen_db_size_warning_threshold_mb` integer
  Free-space threshold for the KLEN database file, in megabytes. When the remaining space (`max_klen_db_size_mb` minus the current file size) drops below this value, a warning is printed as a comment inside the Z-report itself (and reprinted on copies/reports). Must be non-negative. Defaults to `100` when not set (so the warning starts at ~1.9 GB).
- `update_config` › object
  Firmware-update behaviour for the out-of-process update client. Keys are read **only** from this section; a copy placed at the top level of the configuration is ignored. Any subset may be sent — omitted keys keep their defaults.

Responses

> **200** Configuration update was successful

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **500** Unexpected server-side failure.

PUT /config

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
 - "receipt_config": {
      "cut_type": "FULL",
      "delimiter_char": "-",
      "before_print_feed": 3,
      "before_cut_feed": 5,
      "encoding": "CP1251",
      "receipt_ext_format": true,
      "receipt_print_vat": true,
      "print_dark": 30,
      "print_speed": 2,
      "header_logo_enabled": true,
      "header_logo_max_height": 300,
    + "header_style": { … },
    + "body_style": { … },
    + "footer_style": { … },
    + "total_style": { … }
   },
 - "ip_whitelist": [
      "127.0.0.1",
      "192.168.1.50"
   ],
   "operator_label": "----",
   "footer_line1": "",
   "footer_line2": "",
   "localization_language": "bg",
   "auto_z_report_enabled": false,
   "allow_empty_z_report": false,
   "auto_z_report_virtual": true,
   "auto_backup_enabled": true,
```

```
    "config_migration_level": 1,
    "global_settings_update_interval_sec": 1800,
    "max_fm_blocks_warning_threshold": 100,
    "max_klen_db_size_mb": 2048,
    "klen_db_size_warning_threshold_mb": 100,
  - "update_config": {
        "update_check_interval_sec": 3600,
        "update_check_retry_interval_sec": 300,
        "update_apn_check_interval_sec": 86400,
        "command_max_per_run": 100,
        "update_via_apn": "on_failure",
        "update_apn": "",
        "update_apn_route_mode": "default",
        "update_apn_server_url": "",
        "reboot_after_z_for_update": "never"
    }
}
```


Response samples

200 | 400 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "port": 8888,
    "bind_address": "127.0.0.1",
    "db_path": "nsys-fiscal.db",
  - "printer_config": {
        "device_path": "/dev/ttyp0",
        "baud_rate": 115200,
        "data_bits": 8,
        "parity": 0,
        "stop_bits": 1,
        "is_rtscts": 0
    },
```

```
- "receipt_config": {
      "cut_type": "FULL",
      "delimiter_char": "-",
      "before_print_feed": 3,
      "before_cut_feed": 5,
      "encoding": "CP1251",
      "receipt_ext_format": true,
      "receipt_print_vat": true,
      "print_dark": 30,
      "print_speed": 2,
      "header_logo_enabled": true,
      "header_logo_max_height": 300,
    + "header_style": { … },
    + "body_style": { … },
    + "footer_style": { … },
    + "total_style": { … }
  },
  "printer_type": "aclas",
  "fm_backend_type": "aclas",
  "modem_status_script": "./system-scripts/get_device_info.sh --json --port=/d
  "modem_device_file": "/dev/ttyUSB0",
  "sms_save_path": "./sms-messages",
  "nra_access_url": "http://10.199.1.11:7000/fdmon3/FDMonitorPort",
- "ip_whitelist": [
      "127.0.0.1",
      "192.168.1.50"
  ],
  "operator_label": "----",
  "footer_line1": "",
  "footer_line2": "",
  "localization_language": "bg",
  "resource_path": "/tmp/sd/fiscal-nsys/resources",
  "auto_z_report_enabled": false,
  "allow_empty_z_report": false,
  "auto_z_report_virtual": true,
  "auto_backup_enabled": true,
  "config_migration_level": 1,
  "global_settings_update_interval_sec": 1800,
  "max_fm_blocks_warning_threshold": 100,
  "max_klen_db_size_mb": 2048,
  "klen_db_size_warning_threshold_mb": 100,
```

```
  - "update_config": {
        "update_check_interval_sec": 3600,
        "update_check_retry_interval_sec": 300,
        "update_apn_check_interval_sec": 86400,
        "command_max_per_run": 100,
        "update_via_apn": "on_failure",
        "update_apn": "",
        "update_apn_route_mode": "default",
        "update_apn_server_url": "",
        "reboot_after_z_for_update": "never"
    },
    "gsm_max_z_failures": 3,
    "unsent_receipts_max_age_hours": 72
}
```

Creates a new configuration

Saves the persisted configuration. Read-only startup fields such as `port`, `db_path`, `printer_config`, `printer_type`, `fm_backend_type`, `modem_status_script`, `modem_device_file`, `sms_save_path`, and `nra_access_url` are ignored on write and are still returned by subsequent GET requests.

Application status codes (JSON `status`): `0` success, `1001` invalid request JSON/body, `1503` config save/create failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- `receipt_config` >
  object
  The receipt configuration object
- `ip_whitelist`
  Array of strings
  Client-IP allow-list for incoming requests, HTTP and HTTPS alike. When set to a non-empty array, only requests whose client address matches an entry are served; all others receive `403` (status `2350`). When absent or an empty array, the allow-list is disabled and every client is accepted. IPv4-mapped IPv6 clients are

normalized to plain IPv4. A single comma/whitespace-separated string is also accepted. A non-empty list must include `127.0.0.1`: the internal callers (nra_client, nra_cron, auto-Z) reach the service over loopback and are subject to the same list. For HTTPS the list is enforced by nsys-tls-proxy on the real client address before the TLS handshake — the service sees only the proxy's loopback address — so a denied HTTPS client gets the connection closed rather than a `403` body.

`operator_label`

string
Operator label for receipts

`footer_line1`

string
First footer line for receipts

`footer_line2`

string
Second footer line for receipts

`localization_language`

string
Active localization resource to load from `resource_path/localization/<language>.json`

`auto_z_report_enabled`

boolean
Enable automatic Z report generation after the service detects that a new day has started

---

`allow_empty_z_report`

boolean
Allow generating Z reports with no transactions

`auto_z_report_virtual`

boolean
Use `virtual=true` when the automatic Z report flow calls `POST /report/Z_REPORT`

`auto_backup_enabled`

boolean
Enable automatic daily database backup generation shortly after midnight

---

`config_migration_level`

integer
Level of the last one-shot config migration applied to this device. Maintained by the service at startup: any migration above this level runs once, then the level is stamped so it never runs again. Level 1 moved `receipt_config.print_dark` from the old default 2 to 30. Informational — do not set it by hand.

---

global_settings_update_interval_sec

integer

Interval in seconds for global settings updates

---

max_fm_blocks_warning_threshold

integer

Threshold for remaining free fiscal-memory Z-report slots. When the number of free slots drops below this value, a warning is printed as a comment inside the Z-report itself (and reprinted on copies/reports). Must be non-negative.

---

max_klen_db_size_mb

integer

Maximum size of the KLEN (SQLite electronic journal) database file, in megabytes. Once the file reaches this size the device goes read-only (only reports and exports are allowed), exactly like a full fiscal memory. Must be positive. Defaults to `2048` (2 GB) when not set.

---

klen_db_size_warning_threshold_mb

integer

Free-space threshold for the KLEN database file, in megabytes. When the remaining space (`max_klen_db_size_mb` minus the current file size) drops below this value, a warning is printed as a comment inside the Z-report itself (and reprinted on copies/reports). Must be non-negative. Defaults to `100` when not set (so the warning starts at ~1.9 GB).

---

update_config ›

object

Firmware-update behaviour for the out-of-process update client. Keys are read **only** from this section; a copy placed at the top level of the configuration is ignored. Any subset may be sent — omitted keys keep their defaults.

Responses

› **200** Configuration creation was successful

› **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

› **500** Unexpected server-side failure.

POST /config

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
 - "receipt_config": {
      "cut_type": "FULL",
      "delimiter_char": "-",
      "before_print_feed": 3,
      "before_cut_feed": 5,
      "encoding": "CP1251",
      "receipt_ext_format": true,
      "receipt_print_vat": true,
      "print_dark": 30,
      "print_speed": 2,
      "header_logo_enabled": true,
      "header_logo_max_height": 300,
    + "header_style": { … },
    + "body_style": { … },
    + "footer_style": { … },
    + "total_style": { … }
   },
 - "ip_whitelist": [
      "127.0.0.1",
      "192.168.1.50"
   ],
   "operator_label": "----",
   "footer_line1": "",
   "footer_line2": "",
   "localization_language": "bg",
   "auto_z_report_enabled": false,
   "allow_empty_z_report": false,
   "auto_z_report_virtual": true,
   "auto_backup_enabled": true,
```

```
    "config_migration_level": 1,
    "global_settings_update_interval_sec": 1800,
    "max_fm_blocks_warning_threshold": 100,
    "max_klen_db_size_mb": 2048,
    "klen_db_size_warning_threshold_mb": 100,
  - "update_config": {
        "update_check_interval_sec": 3600,
        "update_check_retry_interval_sec": 300,
        "update_apn_check_interval_sec": 86400,
        "command_max_per_run": 100,
        "update_via_apn": "on_failure",
        "update_apn": "",
        "update_apn_route_mode": "default",
        "update_apn_server_url": "",
        "reboot_after_z_for_update": "never"
    }
}
```


Response samples

200 | 400 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "port": 8888,
    "bind_address": "127.0.0.1",
    "db_path": "nsys-fiscal.db",
  - "printer_config": {
        "device_path": "/dev/ttyp0",
        "baud_rate": 115200,
        "data_bits": 8,
        "parity": 0,
        "stop_bits": 1,
        "is_rtscts": 0
    },
```

```
- "receipt_config": {
      "cut_type": "FULL",
      "delimiter_char": "-",
      "before_print_feed": 3,
      "before_cut_feed": 5,
      "encoding": "CP1251",
      "receipt_ext_format": true,
      "receipt_print_vat": true,
      "print_dark": 30,
      "print_speed": 2,
      "header_logo_enabled": true,
      "header_logo_max_height": 300,
    + "header_style": { … },
    + "body_style": { … },
    + "footer_style": { … },
    + "total_style": { … }
  },
  "printer_type": "aclas",
  "fm_backend_type": "aclas",
  "modem_status_script": "./system-scripts/get_device_info.sh --json --port=/d
  "modem_device_file": "/dev/ttyUSB0",
  "sms_save_path": "./sms-messages",
  "nra_access_url": "http://10.199.1.11:7000/fdmon3/FDMonitorPort",
- "ip_whitelist": [
      "127.0.0.1",
      "192.168.1.50"
  ],
  "operator_label": "----",
  "footer_line1": "",
  "footer_line2": "",
  "localization_language": "bg",
  "resource_path": "/tmp/sd/fiscal-nsys/resources",
  "auto_z_report_enabled": false,
  "allow_empty_z_report": false,
  "auto_z_report_virtual": true,
  "auto_backup_enabled": true,
  "config_migration_level": 1,
  "global_settings_update_interval_sec": 1800,
  "max_fm_blocks_warning_threshold": 100,
  "max_klen_db_size_mb": 2048,
  "klen_db_size_warning_threshold_mb": 100,
```

```
  - "update_config": {
        "update_check_interval_sec": 3600,
        "update_check_retry_interval_sec": 300,
        "update_apn_check_interval_sec": 86400,
        "command_max_per_run": 100,
        "update_via_apn": "on_failure",
        "update_apn": "",
        "update_apn_route_mode": "default",
        "update_apn_server_url": "",
        "reboot_after_z_for_update": "never"
    },
    "gsm_max_z_failures": 3,
    "unsent_receipts_max_age_hours": 72
}
```

Updates the company information

Partially updates the company/header data used on receipts.

Application status codes (JSON `status`): `0` success, `1001` invalid request JSON/body, `1503` company/header update failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- companyName
  string
  The name of the company
- companyAddress
  string
  The address of the company
- storeName
  string
  The name of the store
- storeAddress
  string
  The address of the store
- companyVat
  string

The company Vat

companyNumber

string

Deprecated

Deprecated legacy company identifier. Prefer registration data instead.

Responses

> **200** Company update was successful

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **500** Unexpected server-side failure.

PUT /header

Request samples

Payload

Content type

application/json

Copy

```
{
    "companyName": "ЕНСИС ООД",
    "companyAddress": "гр.София",
    "storeName": "ЕНСИС ООД",
    "storeAddress": "ИЛИНДЕН",
    "companyVat": "BG112412412",
    "companyNumber": "123456789"
}
```

Response samples

200 400 500

Content type
application/json

Copy

```
{
    "companyName": "ЕНСИС ООД",
    "companyAddress": "гр.София",
    "storeName": "ЕНСИС ООД",
    "storeAddress": "ИЛИНДЕН",
    "companyVat": "BG112412412",
    "companyNumber": "123456789"
}
```

Creates a new company

Saves the company/header data used on receipts.

Application status codes (JSON `status`): `0` success, `1001` invalid request JSON/body, `1503` company/header save failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- `companyName` — string
  The name of the company
- `companyAddress` — string
  The address of the company
- `storeName` — string
  The name of the store
- `storeAddress` — string
  The address of the store
- `companyVat` — string
  The company Vat

companyNumber

string

Deprecated

Deprecated legacy company identifier. Prefer registration data instead.

Responses

> **200** Company creation was successful

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **500** Unexpected server-side failure.

POST /header

Request samples

Payload

Content type

application/json

Copy

```
{
    "companyName": "ЕНСИС ООД",
    "companyAddress": "гр.София",
    "storeName": "ЕНСИС ООД",
    "storeAddress": "ИЛИНДЕН",
    "companyVat": "BG112412412",
    "companyNumber": "123456789"
}
```

Response samples

200 400 500

Content type

application/json

Copy

```
{
    "companyName": "ЕНСИС ООД",
    "companyAddress": "гр.София",
    "storeName": "ЕНСИС ООД",
    "storeAddress": "ИЛИНДЕН",
    "companyVat": "BG112412412",
    "companyNumber": "123456789"
}
```

Upload the company logo

Sets the company logo printed centered at the top of every fiscal receipt and invoice, above the company header - also on the sales of a device that is not registered yet. It is not printed on duplicates, copies reprinted from the journal, service (non-fiscal) receipts or reports; `receipt_config.header_logo_enabled: false` switches it off without deleting it. Printed on the ACLAS printer only.

The body is a BMP file (`Content-Type: image/bmp`, at most 4 MB and 2048x2048 pixels): uncompressed 1/4/8-bit palettized, 24-bit, or 32-bit (BI_RGB or BI_BITFIELDS), bottom-up or top-down. The device converts it to black and white (50% threshold; transparent pixels print white), scales it down to the 576-dot paper width when it is wider (never up), centers it on a full-width white canvas and stores it in the KLEN, replacing any previous logo. The response is that converted image.

The printed height (after scaling, and after cropping with `trim=true`) may not exceed `receipt_config.header_logo_max_height` (default 300 dots, about 37 mm).

Margins in the file are kept unless `trim=true` is given. Independently of the image, the printer library feeds one blank text line after it.

With `print=true` the printer is checked first (`503` / `2377` and nothing is stored when it cannot print); after storing, the logo is printed once followed by a cut - no receipt and no journal entry - and `X-Logo-Printed` reports the outcome. A failed test print does not undo the upload.

Refused (`423`) while the device is read-only for any reason - foreign or uninitialized KLEN, swapped SIM, deregistration, full storage, the GSM no-connection lock - and while emergency mode is active (`2375`). Application status codes (JSON `status`): `2371` not a valid BMP (empty, truncated, inconsistent header), `2372` unsupported BMP (compression, bit depth, size), `2373` too tall, `2374` store failed,

`2375` emergency mode active, `2377` printer cannot print (with `print=true`), `2301` / `2320` / `2330` / `2335` device read-only.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

⊣ `print` boolean
Default: `false`
`true` prints the logo once after storing it.

⊣ `trim` boolean
Default: `false`
`true` crops the white rows and columns around the drawing before scaling and centering, so the logo uses the paper width and no blank paper is printed above or below it. The height limit applies to the cropped size. A picture with nothing that prints is kept as it is.

REQUEST BODY SCHEMA: image/bmp
required

string <binary>

Responses

> **200** The logo was stored; the body is the converted image that will be printed

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **422** Request body is structurally valid but fails local business validation.

> **423** The logo cannot be changed now: the device is read-only, or emergency mode is active (2375).

> **500** Unexpected server-side failure.

> **503** Service dependency is temporarily unavailable.

PUT /printer/logo

Response samples

400 | 409 | 422 | 423 | 500 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
   "status": 1001,
   "message": "Failed to parse JSON",
 - "logs": [
      "string"
   ]
}
```


Remove the company logo

Removes the logo; receipts print without it from then on. Refused (`423`) in the same states as PUT. A factory reset removes the logo too. Application status codes (JSON `status`): `0` deleted, `2370` no logo configured, `2375` emergency mode active, `2376` delete failed.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Logo removed

> **404** Requested resource was not found.

> **423** The logo cannot be changed now: the device is read-only, or emergency mode is active (2375).

> **500** Unexpected server-side failure.

DELETE /printer/logo

Response samples

200 | 404 | 423 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
   "status": 1001,
   "message": "Failed to parse JSON",
 - "logs": [
      "string"
   ]
}
```


Print the stored company logo

Prints the stored logo once, followed by a cut, to check on paper how it comes out - the same print as `PUT /printer/logo?print=true`. It is not a receipt: no header, no footer, no journal entry. Changes nothing, so it is allowed in every device state. Application status codes (JSON `status`): `0` printed, `2370` no logo configured, `2379` the printer cannot print now, `2378` the print failed.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Logo printed

> **404** Requested resource was not found.

> **500** Unexpected server-side failure.

> **503** Service dependency is temporarily unavailable.

POST /printer/logo/print

Response samples

200 | 404 | 500 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "status": 1001,
    "message": "Failed to parse JSON",
  - "logs": [
        "string"
    ]
}
```


Replace the current cliche array

Replaces the legacy cliche representation. The request body must contain exactly 5 strings in this order: company name, company address, store name, store address, company VAT. EIK is not accepted here and is injected only in GET responses. Application status codes (JSON `status`): `0` success, `1001` invalid request JSON/body, `1503` cliche update failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

```
Array (= 5 items) [
  string
]
```

Responses

> **200** Cliche array replaced successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **500** Unexpected server-side failure.

PUT /cliche

Request samples

Payload

Content type
application/json

Copy

```
[
   "string",
   "string",
   "string",
   "string",
   "string"
]
```

Response samples

200 400 500

Content type
application/json

Copy

```
[
    "string",
    "string",
    "string",
    "string",
    "string",
    "string"
]
```

Create a new cliche array

Saves the legacy cliche representation. The request body must contain exactly 5 strings in this order: company name, company address, store name, store address, company VAT. EIK is not accepted here and is injected only in GET responses. Application status codes (JSON `status`): `0` success, `1001` invalid request JSON/body, `1503` cliche save failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

```
Array (= 5 items) [
  string
]
```

Responses

> **200** Cliche array created successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **500** Unexpected server-side failure.

POST /cliche

Request samples

Payload

Content type
application/json

Copy

```
[
   "string",
   "string",
   "string",
   "string",
   "string"
]
```

Response samples

200 400 500

Content type
application/json

Copy

```
[
   "string",
   "string",
   "string",
   "string",
   "string",
   "string"
]
```

Updates the VAT information

Updates the VAT configuration and returns the saved VAT-change receipt. The operation is rejected when there are open reports, so it should be performed only when the Z-report state is empty.

Operator identity: The request may include `operator: { id, name }` or `operatorId`, but not both. Operator fields are receipt metadata only and are not part of VAT-rate validation. If both fields are supplied, the request fails with `400`. If both fields are omitted, legacy behavior is preserved.

Application status codes (JSON `status`): `0` success, `1001` invalid request JSON/body, `1507` open-reports conflict, `1503` VAT update failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

- `virtual` — boolean
  Indicates if the receipt is virtual.

REQUEST BODY SCHEMA: application/json
required

- `taxA` — number <float>
  Tax A rate
- `taxB` — number <float>
  Tax B rate
- `taxC` — number <float>
  Tax C rate
- `taxD` — number <float>
  Tax D rate
- `taxE` — number <float>
  Tax E rate
- `taxF` — number <float>
  Tax F rate
- `taxG` — number <float>
  Tax G rate
- `taxH` — number <float>
  Tax H rate
- `operator` > — object (OperatorRef)

Minimal operator identity stored in receipt-producing requests. If a receipt request sends this object, do not also send `operatorId`.

operatorId

integer `>= 1`

Optional existing operator id to load and store in the VAT-change receipt. Do not send together with `operator`.

Responses

> **200** VAT update was successful

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

PUT /vat

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
    "taxA": 20,
    "taxB": 9,
    "taxC": 9,
    "taxD": -1,
    "taxE": -1,
    "taxF": -1,
    "taxG": -1,
    "taxH": -1,
```

```
  - "operator": {
        "id": 1,
        "name": "Ivan Ivanov"
    },
    "operatorId": 1
}
```

Response samples

200 | 400 | 409

Content type
application/json

Copy

```
{
    "id": 12,
    "timestamp": "2026-03-10T15:35:39",
    "type": 2,
    "sha1": "0284765de7924ea507ad2bc178f506f5dda55ea7",
    "duplicate_of": 0,
    "send_to_nra_tid": -1,
    "zid": 1,
    "is_virtual": true
}
```

Creates a new VAT

Creates the VAT configuration and returns the saved VAT-change receipt. The operation is rejected when there are open reports, so it should be performed only when the Z-report state is empty.

Operator identity: The request may include `operator: { id, name }` or `operatorId`, but not both. Operator fields are receipt metadata only and are not part of VAT-rate validation. If both fields are supplied, the request fails with `400`. If both fields are omitted, legacy behavior is preserved.

Application status codes (JSON `status`): `0` success, `1001` invalid request JSON/body, `1507` open-reports conflict, `1503` VAT save failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

- `virtual` — boolean
  Indicates if the receipt is virtual.

REQUEST BODY SCHEMA: application/json
required

- `taxA` — number <float>
  Tax A rate
- `taxB` — number <float>
  Tax B rate
- `taxC` — number <float>
  Tax C rate
- `taxD` — number <float>
  Tax D rate
- `taxE` — number <float>
  Tax E rate
- `taxF` — number <float>
  Tax F rate
- `taxG` — number <float>
  Tax G rate
- `taxH` — number <float>
  Tax H rate
- `operator` › — object (OperatorRef)
  Minimal operator identity stored in receipt-producing requests. If a receipt request sends this object, do not also send `operatorId`.
- `operatorId` — integer >= 1
  Optional existing operator id to load and store in the VAT-change receipt. Do not send together with `operator`.

Responses

› **200** VAT creation was successful

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

POST /vat

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
   "taxA": 20,
   "taxB": 9,
   "taxC": 9,
   "taxD": -1,
   "taxE": -1,
   "taxF": -1,
   "taxG": -1,
   "taxH": -1,
 - "operator": {
      "id": 1,
      "name": "Ivan Ivanov"
   },
   "operatorId": 1
}
```


Response samples

200 400 409

Content type
application/json

Copy

```
{
    "id": 12,
    "timestamp": "2026-03-10T15:35:39",
    "type": 2,
    "sha1": "0284765de7924ea507ad2bc178f506f5dda55ea7",
    "duplicate_of": 0,
    "send_to_nra_tid": -1,
    "zid": 1,
    "is_virtual": true
}
```


Updates payment labels

Partially updates the stored labels for `OTHER` and `OTHER2`.

Application status codes (JSON `status`): `0` success, `1001` invalid request JSON/body, `1503` payment-labels update failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

label_payment_other required	string Label for payment type OTHER
label_payment_other2 required	string Label for payment type OTHER2

Responses

> **200** Payment labels update was successful

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **500** Unexpected server-side failure.

PUT /payment/labels

Request samples

Payload

Content type
application/json

Copy

```
{
    "label_payment_other": "с други средства",
    "label_payment_other2": "с други средства 2"
}
```

Response samples

200 | 400 | 500

Content type
application/json

Copy

```
{
    "label_payment_other": "с други средства",
    "label_payment_other2": "с други средства 2"
}
```

Creates new payment labels entry

Saves the labels for `OTHER` and `OTHER2`.

Application status codes (JSON `status`): `0` success, `1001` invalid request JSON/body, `1503` payment-labels save failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- `label_payment_other` (required) — string
  Label for payment type OTHER
- `label_payment_other2` (required) — string
  Label for payment type OTHER2

Responses

> **200** Payment labels creation was successful

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **500** Unexpected server-side failure.

POST /payment/labels

Request samples

Payload

Content type
application/json

Copy

```
{
    "label_payment_other": "с други средства",
    "label_payment_other2": "с други средства 2"
}
```

Response samples

200 | 400 | 500

Content type
application/json

Copy

```
{
    "label_payment_other": "с други средства",
    "label_payment_other2": "с други средства 2"
}
```

Set system date and time

Sets the system time. Accepts exactly one of 'timestamp' (UNIX time) or 'datetime' (ISO 8601 string). Both or none are not allowed.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

One of | SystemDateTimeTimestampRequest | SystemDateTimeIsoRequest

⊣ `timestamp`
required

integer
UNIX timestamp to set.

Responses

> **200** System time updated successfully

> **400** Bad request

POST /system/datetime

Request samples

Payload

Content type
application/json

Example
Set system time using UNIX timestamp

Copy

```
{
    "timestamp": 1776670200
}
```

Response samples

200 400

Content type
application/json

Copy

```
{
    "status": 0,
    "message": "Hello, world!"
}
```

service: master-data

Petrol-station structure and numbering configuration - terminals, dispensers, nozzles, petrol-station, invoice range, and USN writes.

Create a new terminal **Deprecated**

Internal, unofficial local setup endpoint. It should not be used by external integrations. In real operation, initial petrol-station creation belongs to `POST /registration`, and later petrol-station changes should be submitted through `POST /change-registration` so that NRA stays in sync. This operation is rejected while registration is active and should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- **TNo**
  integer
  Unique terminal number. Required for create; ignored when the number is provided by the path on update.
- **VolAct**
  required
  number <float>
  Fuel volume at normal temperature
- **Vol15C**
  required
  number <float>
  Fuel volume at 15°C
- **FL**
  required
  number <float>
  Fuel level
- **Tmpr**
  required
  number <float>
  Current temperature
- **FTNom**
  required
  string
  Fuel tank nomenclature / label
- **FTT**
  required
  integer
  Fuel type code
- **TVol**
  required
  number <float>
  Tank capacity
- **IsCalib**
  required
  integer
  Is calibrated (0/1)
- **FuelName**
  string

required

Fuel name

taxId

integer

VAT group ID (1–8) for the fuel article that will be created from this terminal. Optional — if omitted the article is stored with `taxId=0` (unset) and must be configured later via `PUT /fuels/{FTT}` before any sale can be processed.

price

number <float>

Unit price of the fuel article. Optional — if omitted the article is stored with a negative sentinel price (unset) and must be configured later via `PUT /fuels/{FTT}` before any sale can be processed.

CalibDate
required

string

Calibration date (YYYY-MM-DD)

Responses

> **200** Terminal created successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

POST /terminals

Request samples

Payload

Content type
application/json

Copy

```
{
    "TNo": 1,
    "VolAct": 0.01,
```

```
    "Vol15C": 0.22,
    "FL": 45,
    "Tmpr": 0,
    "FTNom": "27101141",
    "FTT": 101,
    "TVol": 5000,
    "IsCalib": 1,
    "FuelName": "Дизел 1",
    "taxId": 3,
    "price": 2.35,
    "CalibDate": "2012-11-01"
}
```


Response samples

200 | 400 | 409 | 500

Content type
application/json

Copy


```
{
    "TNo": 1,
    "VolAct": 0.01,
    "Vol15C": 0.22,
    "FL": 45,
    "Tmpr": 0,
    "FTNom": "27101141",
    "FTT": 101,
    "TVol": 5000,
    "IsCalib": 1,
    "FuelName": "Дизел 1",
    "taxId": 3,
    "price": 2.35,
    "CalibDate": "2012-11-01",
    "lastUpdated": 1719936000,
    "lastLevelUpdated": 1719936000
}
```

Update terminal by number **Deprecated**

Internal, unofficial local update endpoint. It should not be used by external integrations. Petrol-station changes must go through `POST /change-registration` so that NRA stays in sync. Any local change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

- `number` required
  integer
  Numeric path identifier for number-based resources.

REQUEST BODY SCHEMA: application/json
required

- `TNo`
  integer
  Unique terminal number. Required for create; ignored when the number is provided by the path on update.
  - `VolAct` required
    number <float>
    Fuel volume at normal temperature
  - `Vol15C` required
    number <float>
    Fuel volume at 15°C
  - `FL` required
    number <float>
    Fuel level
  - `Tmpr` required
    number <float>
    Current temperature
  - `FTNom` required
    string
    Fuel tank nomenclature / label
  - `FTT` required
    integer
    Fuel type code
  - `TVol` required
    number <float>
    Tank capacity

IsCalib required	integer	Is calibrated (0/1)
FuelName required	string	Fuel name
taxId	integer	VAT group ID (1–8) for the fuel article that will be created from this terminal. Optional — if omitted the article is stored with `taxId=0` (unset) and must be configured later via `PUT /fuels/{FTT}` before any sale can be processed.
price	number <float>	Unit price of the fuel article. Optional — if omitted the article is stored with a negative sentinel price (unset) and must be configured later via `PUT /fuels/{FTT}` before any sale can be processed.
CalibDate required	string	Calibration date (YYYY-MM-DD)

Responses

> **200** Terminal updated successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **500** Unexpected server-side failure.

PUT /terminals/{number}

Request samples

Payload

Content type
application/json

Copy


```
{
    "TNo": 1,
    "VolAct": 0.01,
    "Vol15C": 0.22,
    "FL": 45,
    "Tmpr": 0,
    "FTNom": "27101141",
    "FTT": 101,
    "TVol": 5000,
    "IsCalib": 1,
    "FuelName": "Дизел 1",
    "taxId": 3,
    "price": 2.35,
    "CalibDate": "2012-11-01"
}
```


Response samples

200 | 400 | 404 | 500

Content type

application/json

Copy

```
{
    "TNo": 1,
    "VolAct": 0.01,
    "Vol15C": 0.22,
    "FL": 45,
    "Tmpr": 0,
    "FTNom": "27101141",
    "FTT": 101,
    "TVol": 5000,
    "IsCalib": 1,
    "FuelName": "Дизел 1",
    "taxId": 3,
    "price": 2.35,
    "CalibDate": "2012-11-01",
    "lastUpdated": 1719936000,
    "lastLevelUpdated": 1719936000
}
```


Delete terminal by number **Deprecated**

Internal, unofficial local delete endpoint. It should not be used by external integrations. Petrol-station changes must go through `POST /change-registration` so that NRA stays in sync. This operation is rejected while registration is active and should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

number required	integer Numeric path identifier for number-based resources.

Responses

> **200** Terminal deleted successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

DELETE `/terminals/{number}`

Response samples

200 | 400 | 404 | 409 | 500

Content type
application/json

Copy

```
{
    "status": 0,
    "message": "Hello, world!"
}
```

Create a new dispenser **Deprecated**

Internal, unofficial local setup endpoint. It should not be used by external integrations. In real operation, initial petrol-station creation belongs to `POST /registration`, and later petrol-station changes should be submitted through `POST /change-registration` so that NRA stays in sync. This operation is rejected while registration is active and should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json

required

- DNo
  integer
  Unique dispenser number. Required for create; ignored when the number is provided by the path on update.

- DNType
  required
  integer
  Enum: 1 2 3
  Device type (1=pump, 2=flowmeter, 3=measuring system)

- Doos
  required
  integer
  Enum: 1 2
  Device status (1=approved type, 2=conformity assessed)

- DModel
  required
  string
  Device model

- DType
  required
  integer
  Enum: 1 2
  High flow flag (1=yes, 2=no)

Responses

> **200** Dispenser created successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

POST /dispensers

Request samples

Payload

Content type
application/json

Copy


```
{
    "DNo": 1,
    "DNType": 1,
    "Doos": 2,
    "DModel": "Model X",
    "DType": 1
}
```


Response samples

200 | 400 | 409 | 500

Content type
application/json

Copy


```
{
    "DNo": 1,
    "DNType": 1,
    "Doos": 2,
    "DModel": "Model X",
    "DType": 1
}
```


Update dispenser by number **Deprecated**

Internal, unofficial local update endpoint. It should not be used by external integrations. Petrol-station changes must go through `POST /change-registration` so that NRA stays in sync. This operation is rejected while registration is active and should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

- number
  required
  integer
  Numeric path identifier for number-based resources.

REQUEST BODY SCHEMA: application/json
required

- DNo
  integer
  Unique dispenser number. Required for create; ignored when the number is provided by the path on update.
- DNType
  required
  integer
  Enum: 1 2 3
  Device type (1=pump, 2=flowmeter, 3=measuring system)
- Doos
  required
  integer
  Enum: 1 2
  Device status (1=approved type, 2=conformity assessed)
- DModel
  required
  string
  Device model
- DType
  required
  integer
  Enum: 1 2
  High flow flag (1=yes, 2=no)

Responses

> **200** Dispenser updated successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

PUT /dispensers/{number}

Request samples

Payload

Content type
application/json

Copy

```
{
    "DNo": 1,
    "DNType": 1,
    "Doos": 2,
    "DModel": "Model X",
    "DType": 1
}
```

Response samples

200 | 400 | 404 | 409 | 500

Content type
application/json

Copy

```
{
    "DNo": 1,
    "DNType": 1,
    "Doos": 2,
    "DModel": "Model X",
    "DType": 1
}
```

Delete dispenser by number Deprecated

Internal, unofficial local delete endpoint. It should not be used by external integrations. Petrol-station changes must go through `POST /change-registration` so that NRA stays in sync. This operation is rejected while registration is active and should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

number required	integer Numeric path identifier for number-based resources.

Responses

> **200** Dispenser deleted successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

DELETE /dispensers/{number}

Response samples

200 | 400 | 404 | 409 | 500

Content type
application/json

Copy

```
{
    "status": 0,
```

```
    "message": "Hello, world!"
}
```

Create a new nozzle **Deprecated**

Internal, unofficial local setup endpoint. It should not be used by external integrations. In real operation, initial petrol-station creation belongs to `POST /registration`, and later petrol-station changes should be submitted through `POST /change-registration` so that NRA stays in sync. This operation is rejected while registration is active and should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- **NNo**
  integer
  Unique nozzle number. Required for create; ignored when the number is provided by the path on update.
- **TNo** *required*
  integer
  Related terminal number
- **DNo** *required*
  integer
  Related dispenser number
- **CVal** *required*
  number <float>
  Current counter value
- **updateCounterTimestamp** *required*
  integer <int64>
  Counter timestamp supplied by the caller
- **NType** *required*
  integer
  Enum: `1` `2`
  Nozzle high flow type (1=yes, 2=no)

Responses

> **200** Nozzle created successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

POST /nozzles

Request samples

Payload

Content type
application/json

Copy

```
{
    "NNo": 1,
    "TNo": 1,
    "DNo": 1,
    "CVal": 123.45,
    "updateCounterTimestamp": 1719936000,
    "NType": 1
}
```

Response samples

200 400 409 500

Content type
application/json

Copy

```
{
    "NNo": 1,
    "TNo": 1,
```

```
    "DNo": 1,
    "CVal": 123.45,
    "updateCounterTimestamp": 1719936000,
    "NType": 1,
    "FTT": 123
}
```


Delete all nozzles **Deprecated**

Internal, unofficial local delete endpoint. It should not be used by external integrations. Petrol-station changes must go through `POST /change-registration` so that NRA stays in sync. This operation is rejected while registration is active and should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** All nozzles deleted successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

DELETE /nozzles

Response samples

200 | 400 | 409 | 500

Content type
application/json

Copy

```
{
    "status": 0,
    "message": "Hello, world!"
}
```

Update nozzle by number **Deprecated**

Internal, unofficial local update endpoint. It should not be used by external integrations. Petrol-station changes must go through `POST /change-registration` so that NRA stays in sync. This operation is rejected while registration is active and should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

- `number` required — integer
  Numeric path identifier for number-based resources.

REQUEST BODY SCHEMA: application/json
required

- `NNo` — integer
  Unique nozzle number. Required for create; ignored when the number is provided by the path on update.
- `TNo` required — integer
  Related terminal number
- `DNo` required — integer
  Related dispenser number
- `CVal` required — number <float>
  Current counter value

updateCounterTimestamp required	integer <int64> Counter timestamp supplied by the caller
NType required	integer Enum: 1 2 Nozzle high flow type (1=yes, 2=no)

Responses

- **200** Nozzle updated successfully.
- **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.
- **404** Requested resource was not found.
- **409** Request conflicts with the current resource or device state.
- **500** Unexpected server-side failure.

PUT /nozzles/{number}

Request samples

Payload

Content type
application/json

Copy

```
{
    "NNo": 1,
    "TNo": 1,
    "DNo": 1,
    "CVal": 123.45,
    "updateCounterTimestamp": 1719936000,
    "NType": 1
}
```


Response samples

200 | 400 | 404 | 409 | 500

Content type
application/json

Copy


```
{
    "NNo": 1,
    "TNo": 1,
    "DNo": 1,
    "CVal": 123.45,
    "updateCounterTimestamp": 1719936000,
    "NType": 1,
    "FTT": 123
}
```


Delete nozzle by number **Deprecated**

Internal, unofficial local delete endpoint. It should not be used by external integrations. Petrol-station changes must go through `POST /change-registration` so that NRA stays in sync. This operation is rejected while registration is active and should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

PATH PARAMETERS

number
required

integer
Numeric path identifier for number-based resources.

Responses

> **200** Nozzle deleted successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

DELETE /nozzles/{number}

Response samples

200 | 400 | 404 | 409 | 500

Content type
application/json

Copy

```
{
    "status": 0,
    "message": "Hello, world!"
}
```

Create a new petrol station (all terminals, dispensers, nozzles in one request) Deprecated

Internal, unofficial local setup endpoint. It should not be used by external integrations. In real operation, the initial petrol-station creation belongs to `POST /registration`, and later petrol-station changes should be submitted through `POST /change-registration` so that NRA stays in sync. This operation is rejected while registration is active because petrol-station changes must go through NRA, and any local change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

GNo	string Petrol-station registration number (ГРР) — a station-level characteristic. Required for FDType 31; omit for other device types.
terminals > required	Array of objects (TerminalRequest)
dispensers > required	Array of objects (DispenserRequest)
nozzles > required	Array of objects (NozzleRequest)

Responses

> **200** Petrol station created successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

POST /petrol-station

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
    "GNo": "P-4914",
  - "terminals": [
      + { … }
    ],
  - "dispensers": [
      + { … }
    ],
  - "nozzles": [
      + { … }
    ]
}
```


Response samples

200 400 409 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "GNo": "P-4914",
  - "terminals": [
      + { … }
    ],
  - "dispensers": [
      + { … }
    ],
  - "nozzles": [
      + { … }
    ]
}
```

Update the petrol station (all terminals, dispensers, nozzles in one request) **Deprecated**

Internal, unofficial local update endpoint. It should not be used by external integrations. Petrol-station changes must go through `POST /change-registration` so that NRA stays in sync. This operation is rejected while registration is active because petrol-station changes must go through NRA, and any local change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

- GNo
  string
  Petrol-station registration number (ГРР) — a station-level characteristic. Required for FDType 31; omit for other device types.
- terminals >
  required
  Array of objects (TerminalRequest)
- dispensers >
  required
  Array of objects (DispenserRequest)
- nozzles >
  required
  Array of objects (NozzleRequest)

Responses

> **200** Petrol station updated successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.

PUT /petrol-station

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
    "GNo": "P-4914",
  - "terminals": [
      + { … }
    ],
  - "dispensers": [
      + { … }
    ],
  - "nozzles": [
      + { … }
    ]
}
```


Response samples

200 400 404 409 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "GNo": "P-4914",
  - "terminals": [
      + { … }
    ],
```

```
  - "dispensers": [
      + { … }
    ],
  - "nozzles": [
      + { … }
    ]
}
```

Delete the entire petrol station (all terminals, dispensers, and nozzles) Deprecated

Internal, unofficial local delete endpoint. It should not be used by external integrations. Petrol-station changes must go through `POST /change-registration` so that NRA stays in sync. This operation is rejected while registration is active because petrol-station changes must go through NRA, and any local change should be performed only when the Z-report state is empty.

Application status codes (JSON `status`): `0` indicates success; non-zero values are application-specific error/flow codes. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

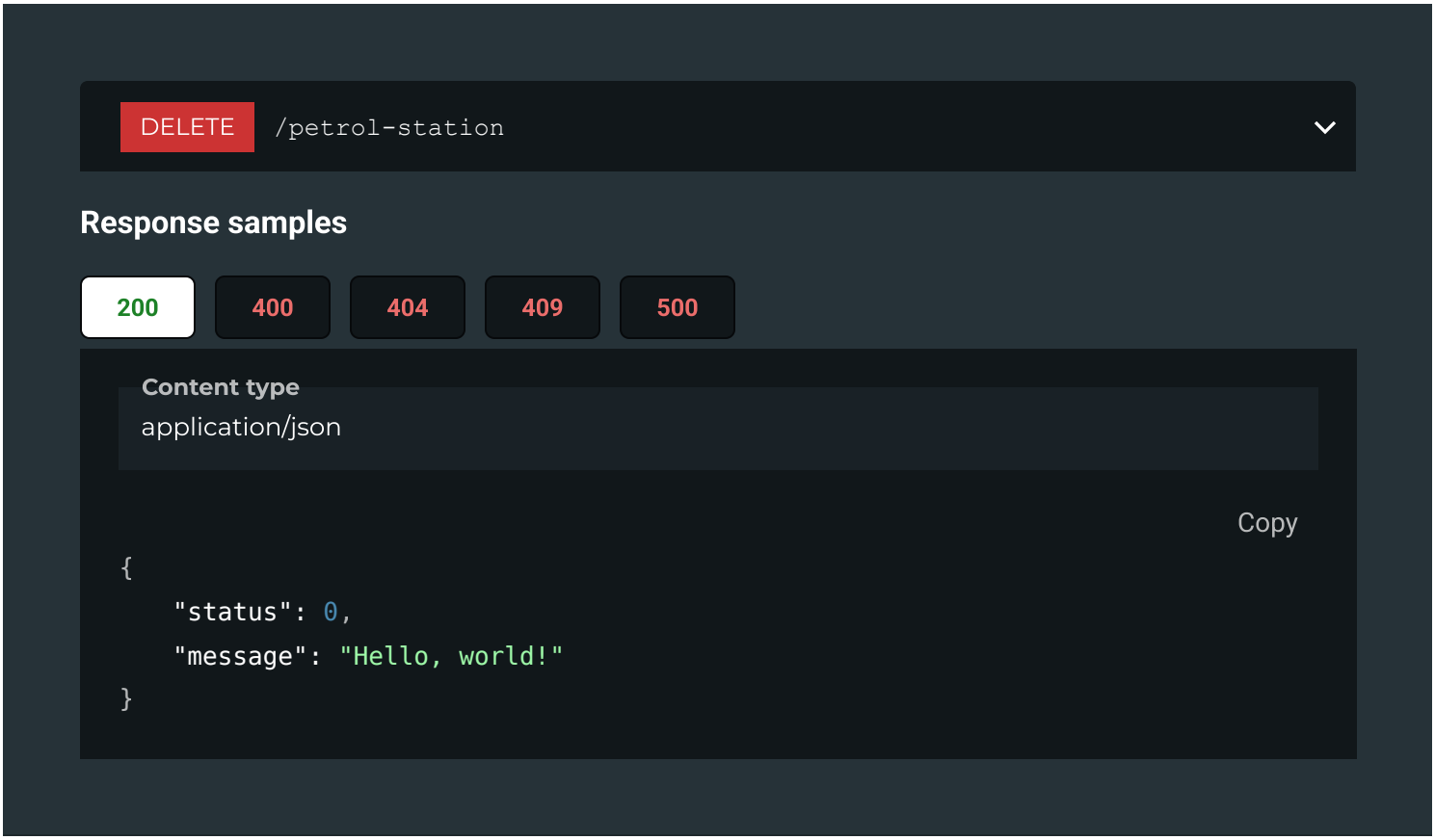
> **200** Petrol station deleted successfully.

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **404** Requested resource was not found.

> **409** Request conflicts with the current resource or device state.

> **500** Unexpected server-side failure.



service: nra

NRA registration lifecycle, communication records, task management, and proxying.

Registers the device

Starts the NRA registration flow. For petrol-station devices (`FDType` 3 or 31), the initial petrol-station configuration must be sent in this request as the `petrol_station` object. Local validation failures return `422`. If NRA rejects an otherwise valid request with a structured response, the endpoint returns that payload with HTTP `422`.

`FDType` **3 vs 31 petrol stations**: `FDType` 31 is a petrol station *with* a level-gauge system — its full `petrol_station` layout (terminals, dispensers, nozzles and `GNo`) is reported to NRA. `FDType` 3 is a petrol station *without* a level gauge: its `petrol_station` payload is used only to define the fuels sold (from `petrol_station.terminals`) and is **not** sent to NRA — the fuels reach NRA only as sales inside the X/Z reports. Any level-gauge/counter data supplied for `FDType` 3 (tank levels, `dispensers`, `nozzles`, `GNo`) is silently dropped and not stored. Consequently the level-gauge and delivery endpoints (`/petrol-station/gauge-state*`, `/petrol-station/delivery-*`) are available for `FDType` 31 only.

Operator identity: The request may include `operator: { id, name }` or `operatorId`, but not both. The operator is not part of the NRA registration payload; it is stored in the locally printed registration receipt so the receipt can be recreated later. If both fields are supplied, the request fails with `400`. If both fields are omitted, legacy behavior is preserved.

KLEN and registration-state preconditions: a *first* registration (device never registered) is refused while the KLEN already holds fiscal activity (any journal entry beyond configuration and cliché) — the endpoint returns `409` with `status 2321` and the KLEN must be reset first. After a *deregistration*, re-registration with the current data is allowed only when the deactivation reason was `10` ("other"); for reasons `1`–`8` (fiscal memory removed/scrapped) and reason `9` (temporary — use `POST /change-registration` instead) the endpoint returns `423` with `status 2320`.

Application status codes (JSON `status`): `0` success, `1305` registration already exists, `1306` provision data missing, `1307` petrol-station payload required, `1308` local validation failed, `1309` NRA create failed, `1310` post-registration task creation failed, `1311` local persistence update failed, `2321` KLEN not empty (reset required), `2320` device deregistered (re-registration not allowed for this deactivation reason). See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

One of **RegistrationRequestVariantWithPS** | RegistrationRequestVariantNoPS

- `FDType` — integer
  Enum: `3` `31` `1` `2` `21` `5`
  Fiscal device type
- `EIK` — string
  EIK (Unified Identification Code)
- `EIKType` — integer
  Enum: `0` `1` `2` `3`
  EIK type
- `FDCert` — string
  required
  Fiscal device certificate
- `MSISDN` — string
  required
  Telephone number
- `OrgName` — string
  Company name

PSNum required	string Store number
PSType required	integer Store type
SEKATTE	string Code EKATTE
Settl	string City
AEkatte	string City code EKATTE (may include suffix)
Area	string Area
StreetCode required	string Street code
Street required	string Street name
StrNo required	string Street number
Block	string Block
En	string Entry number, if any
Fl	string Floor number, if any
Ap	string Additional parameter, if any
PSName required	string Store name
ServiceEIK required	string Service EIK
ServiceEIKType	integer

required	Enum: `0` `1` `2` `3` Service EIK type
ServiceContractExpiration required	string <date-time> Service contract expiration date in ISO 8601 format
petrol_station > required	object (PetrolStation) Petrol station configuration including all terminals, dispensers, and nozzles.
operator >	object (OperatorRef) Minimal operator identity stored in receipt-producing requests. If a receipt request sends this object, do not also send `operatorId`.
operatorId	integer `>= 1` Optional existing operator id to load and store in the local change-registration receipt. Do not send together with `operator`.

Responses

> **200** A successful registration response

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **422** Local validation failed or NRA rejected a structurally valid registration request

> **423**
Device is deregistered and re-registration is not allowed for the recorded deactivation reason (status 2320). Reasons 1–8 are terminal; reason 9 must use POST /change-registration.

> **502** Upstream dependency failed or returned an unusable response.

POST `/registration`

Request samples

Payload

Content type

application/json

Example

Standard registration object with FDType 2

Copy


```
{
    "FDType": 2,
    "EIK": "175192260",
    "EIKType": 0,
    "FDCert": "864",
    "MSISDN": "359892698419",
    "OrgName": "ЕНСИС ООД",
    "PSNum": "12421",
    "PSType": 1,
    "SEKATTE": "68134",
    "Settl": "София",
    "AEkatte": "68134-17",
    "Area": "Витоша",
    "StreetCode": "00059",
    "Street": "АЛЕЯ ФЛОЙД БЛЯК",
    "StrNo": "1",
    "PSName": "Магазин Център",
    "ServiceEIK": "175192260",
    "ServiceEIKType": 0,
    "ServiceContractExpiration": "2027-07-23T23:59:59"
}
```


Response samples

200 | 400 | 409 | 422 | 423 | 502

Content type

application/json

Copy


```
{ }
```

Get stored registration data

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Stored local registration enriched with NRA and debug fields

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

GET /registration

Response samples

200 400

Content type
application/json

Example
RegistrationVariantWithPS

Copy Expand all Collapse all


```
{
    "FDType": 31,
    "EIK": "175192260",
    "EIKType": 0,
    "FDIN": "OS800150",
    "FMIN": "58800150",
    "FDRID": "161407",
    "FDCert": "864",
    "IMSI": "284050052821582",
    "MSISDN": "359892698419",
    "OPID": "1",
    "OrgName": "ЕНСИС ООД",
    "PSNum": "12421",
```

```
    "PSType": 1,
    "SEKATTE": "68134",
    "Settl": "София",
    "AEkatte": "68134-17",
    "Area": "Витоша",
    "StreetCode": "00059",
    "Street": "АЛЕЯ ФЛОЙД БЛЯК",
    "StrNo": "1",
    "Block": "1",
    "En": "1",
    "Fl": "1",
    "Ap": "1",
    "PSName": "тест-нсис",
    "SOD": "2025-08-13T14:50:17",
    "RegDate": "2025-08-13T14:50:17",
    "ServiceEIK": "175192260",
    "ServiceEIKType": 0,
    "ServiceContractExpiration": "2027-07-23T23:59:59",
  - "petrol_station": {
        "GNo": "P-4914",
      + "terminals": [ … ],
      + "dispensers": [ … ],
      + "nozzles": [ … ]
    },
  - "operator": {
        "id": 1,
        "name": "Ivan Ivanov"
    },
    "operatorId": 1,
    "CT": "2025-08-13T14:50:17",
    "Status": 0,
  - "ZRD": {
        "TID": 114898,
        "SD": "2025-08-14T14:50:17",
        "Days": 30
    },
    "state": 1,
    "AD": "2026-03-10T12:25:30.282Z",
    "request_xml": "string",
    "raw_response_xml": "string"
}
```

Changes the device registration

Applies a partial registration update through the NRA flow. For petrol-station devices (`FDType` 3 or 31), operational changes to the petrol-station configuration should be sent through this endpoint together with the `petrol_station` object. The initial petrol-station creation belongs to `POST /registration`. Local precondition conflicts return `409`; local validation failures return `422`; upstream NRA transport failures return `502`. If NRA rejects an otherwise valid change request with a structured payload, the endpoint returns that payload with HTTP `409`.

`FDType` **3 vs 31 petrol stations**: this endpoint is the supported way to change fuels for both types. For `FDType` 31 the petrol-station layout change (terminals/dispensers/nozzles) is mirrored to NRA. For `FDType` 3 the `petrol_station` payload is reduced to its fuel definitions only — the level-gauge/counter data (tank levels, `dispensers`, `nozzles`, `GNo`) is silently dropped — and the change is applied to the local fuels and sent to NRA as a plain `RCHANGE` with no petrol-station element operations.

Operator identity: The request may include `operator: { id, name }` or `operatorId`, but not both. The operator is not part of the NRA change-registration payload; it is stored in the locally printed change-registration receipt so the receipt can be recreated later. If both fields are supplied, the request fails with `400`. If both fields are omitted, legacy behavior is preserved.

Deregistration recovery: this endpoint also re-activates a *temporarily suspended* registration — one deactivated with reason `9`. In that case the device returns to the `REGISTERED` state on success and the deactivation reason is cleared. Other deregistered states are not reactivatable here: reasons `1`–`8` are terminal, and reason `10` is re-registered via `POST /registration` (both return `409` "registration not active").

Application status codes (JSON `status`): `0` success, `1303` registration not active, `1312` unsent reports conflict, `1313` pending-task flush failed, `1307` petrol-station payload required, `1314` NRA update failed, `1311` local persistence update failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

One of **RegistrationChangeRequestWithPS** | RegistrationChangeRequestNoPS

- `FDType` — integer
  Enum: `3` `31` `1` `2` `21` `5`
- `EIK` — string

EIKType	integer
FDCert	string
MSISDN	string
OrgName	string
PSNum	string
PSType	integer
SEKATTE	string
Settl	string
AEkatte	string
Area	string
StreetCode	string
Street	string
StrNo	string
Block	string
En	string
Fl	string
Ap	string
PSName	string
ServiceEIK	string
ServiceEIKType	integer
ServiceContractExpiration	string <date-time>
petrol_station > required	object (PetrolStation) Petrol station configuration including all terminals, dispensers, and nozzles.

Responses

> **200** A successful update registration response

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409**
Local conflict/precondition failure or NRA rejected the change request with a structured business response

> **422** Request body is structurally valid but fails local business validation.

> **502** Upstream dependency failed or returned an unusable response.

POST /change-registration

Request samples

Payload

Content type
application/json

Example
Standard change-registration object with FDType 2

Copy

```
{
    "FDType": 2,
    "PSName": "Магазин Център 2",
    "Street": "БУЛ. ВИТОША",
    "StrNo": "15",
    "MSISDN": "359892698420",
    "ServiceContractExpiration": "2028-07-23T23:59:59"
}
```

Response samples

200 400 409 422 502

Content type
application/json

Copy

```
{ }
```

Deactivates the device registration

Deactivates the registration through the NRA flow. Local lifecycle conflicts return `409`; invalid local request content returns `422`; upstream NRA transport failures return `502`. If NRA rejects an otherwise valid deactivation request with a structured payload, the endpoint returns that payload with HTTP `409`.

Operator identity: The request may include `operator: { id, name }` or `operatorId`, but not both. The operator is not part of the NRA deactivation payload; it is stored in the locally printed deactivation receipt so the receipt can be recreated later. If both fields are supplied, the request fails with `400`. If both fields are omitted, legacy behavior is preserved.

Effect on the device: once deactivated the device becomes read-only (like a foreign KLEN) — every fiscal/write operation is rejected with `423` (`status 2320`); only reports and exports remain available. The recovery path is determined by the `RCFD` reason: reasons `1`–`8` (fiscal memory removal/scrapping) are terminal with no recovery, reason `9` (temporary suspension) is re-activated via `POST /change-registration`, and reason `10` ("other") is re-registered with the current data via `POST /registration`.

Application status codes (JSON `status`): `0` success, `1303` registration not active, `1312` unsent reports conflict, `1313` pending-task flush failed, `1315` invalid/missing RCFD, `1314` NRA update failed, `1311` local persistence update failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

RCFD
required

integer
Enum: `1` `2` `3` `4` `5` `6` `7` `8` `9` `10`
The reason for deactivation (NRA reason code). 1 FM overflow, 2 owner change, 3 termination of registration, 4 scrapping, 5 FM damage, 6 FM block error, 7 putting-in-operation error, 8 testing completed (1–8 remove the fiscal memory

and are terminal), 9 temporary suspension (re-activate via /change-registration), 10 other (re-register via /registration).

operator >

object (OperatorRef)

Minimal operator identity stored in receipt-producing requests. If a receipt request sends this object, do not also send `operatorId`.

operatorId

integer

Optional existing operator id to load and store in the deactivation receipt.

Responses

> **200** A successful deactivation response

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409**
Local conflict/precondition failure or NRA rejected the deactivation request with a structured business response

> **422** Request body is structurally valid but fails local business validation.

> **502** Upstream dependency failed or returned an unusable response.

POST /deactivate-registration

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

```
{
    "RCFD": 1,
```

```
  - "operator": {
        "id": 1,
        "name": "Ivan Ivanov"
    },
    "operatorId": 0
}
```

Response samples

200 | 400 | 409 | 422 | 502

Content type
application/json

Copy

```
{ }
```

Get the NRA registration information

Queries NRA for the current registration information. If NRA responds with a valid payload whose `Status` is non-zero, the endpoint still returns `200` and passes that payload through. Transport-level NRA failures return `502`.

Application status codes (JSON `status`): `0` success, `1301` local registration not found, `1302` local registration fetch failed, `1303` registration not active, `1304` NRA fetch failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Live registration information fetched from NRA

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **409** Request conflicts with the current resource or device state.

> **502** Upstream dependency failed or returned an unusable response.

GET /nra-registration

Response samples

200 | 400 | 409 | 502

Content type
application/json

Copy Expand all Collapse all


```
{
   "EIK": "string",
   "EIKType": 0,
   "FDIN": "string",
   "IMSI": "string",
   "MSISDN": "string",
   "OPID": "string",
   "FDRID": "string",
   "CT": "2019-08-24T14:15:22Z",
   "AD": "2019-08-24T14:15:22Z",
   "Status": 0,
   "XC": 0,
   "ZC": 0,
   "RegDate": "2019-08-24T14:15:22Z",
   "FDType": 0,
 - "ZRD": {
      "TID": 10,
      "SD": "2030-03-10T12:25:30.282Z",
      "Days": 1,
      "Type": "Z"
   },
 - "XRep": {
      "TID": 1,
      "SD": "2019-01-17T00:00:00",
      "MIN": 10,
      "ED": "2019-02-18T23:59:00"
   },
```

```
  - "petrol_station": {
        "GNo": "P-4914",
      + "terminals": [ … ],
      + "dispensers": [ … ],
      + "nozzles": [ … ]
    },
    "request_xml": "string",
    "raw_response_xml": "string"
}
```


Override local registration with NRA data

Fetches registration information from NRA and persists it locally. If NRA responds with a valid payload whose `Status` is non-zero, that payload is returned with HTTP `409` and local state is not updated. Transport-level NRA failures return `502`.

Application status codes (JSON `status`): `0` success, `1303` registration not active, `1302` local registration fetch failed, `1304` NRA fetch failed, `1311` local persistence update failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** NRA registration information used to overwrite local registration state

> **400** Bad request

> **409** Local conflict/precondition failure or NRA returned a structured non-zero `Status` payload

> **502** Upstream dependency failed or returned an unusable response.

POST /nra-registration-sync

Response samples

200 | 400 | 409 | 502

Content type
application/json

Copy Expand all Collapse all


```
{
   "EIK": "string",
   "EIKType": 0,
   "FDIN": "string",
   "IMSI": "string",
   "MSISDN": "string",
   "OPID": "string",
   "FDRID": "string",
   "CT": "2019-08-24T14:15:22Z",
   "AD": "2019-08-24T14:15:22Z",
   "Status": 0,
   "XC": 0,
   "ZC": 0,
   "RegDate": "2019-08-24T14:15:22Z",
   "FDType": 0,
 - "ZRD": {
      "TID": 10,
      "SD": "2030-03-10T12:25:30.282Z",
      "Days": 1,
      "Type": "Z"
   },
 - "XRep": {
      "TID": 1,
      "SD": "2019-01-17T00:00:00",
      "MIN": 10,
      "ED": "2019-02-18T23:59:00"
   },
 - "petrol_station": {
      "GNo": "P-4914",
    + "terminals": [ … ],
    + "dispensers": [ … ],
    + "nozzles": [ … ]
   },
   "request_xml": "string",
   "raw_response_xml": "string"
```

```
}
```

Recover NRA registration

Recovers NRA registration using EIK, EIKType, and MSISDN. If NRA responds with a valid payload whose `Status` is non-zero, the endpoint still returns `200` and passes that payload through unchanged. Transport-level NRA failures return `502`.

Application status codes (JSON `status`): `0` success, `1001` invalid request JSON/body, `1304` NRA fetch/recover failed, `1311` local persistence update failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

REQUEST BODY SCHEMA: application/json
required

EIK required	string
EIKType required	integer
MSISDN required	string
FDCert required	string
OrgName required	string
PSNum required	string
PSType required	integer
SEKATTE required	string
Settl required	string
StreetCode required	string

PSName required	string
ServiceEIK required	string
ServiceEIKType required	integer
ServiceContractExpiration required	string <date-time>

Responses

> **200** Registration recovery result

> **400** Bad request

> **502** Upstream dependency failed or returned an unusable response.

POST /nra-registration-recover

Request samples

Payload

Content type
application/json

Copy

```
{
    "EIK": "string",
    "EIKType": 0,
    "MSISDN": "string",
    "FDCert": "string",
    "OrgName": "string",
    "PSNum": "string",
    "PSType": 0,
    "SEKATTE": "string",
```

```
    "Settl": "string",
    "StreetCode": "string",
    "PSName": "string",
    "ServiceEIK": "string",
    "ServiceEIKType": 0,
    "ServiceContractExpiration": "2019-08-24T14:15:22Z"
}
```


Response samples

200 | 400 | 502

Content type
application/json

Example
RegistrationStoredResponse

Copy Expand all Collapse all


```
{
    "AD": "2026-03-10T12:25:30.282Z",
    "request_xml": "string",
    "raw_response_xml": "string",
    "FDType": 3,
    "EIK": "175192260",
    "EIKType": 0,
    "FDIN": "OS800150",
    "FMIN": "58800150",
    "FDRID": "161407",
    "FDCert": "864",
    "IMSI": "284050052821582",
    "MSISDN": "359892698419",
    "OPID": "1",
    "OrgName": "ЕНСИС ООД",
    "PSNum": "12421",
    "PSType": 1,
    "SEKATTE": "68134",
    "Settl": "София",
    "AEkatte": "68134-17",
    "Area": "Витоша",
    "StreetCode": "00059",
    "Street": "АЛЕЯ ФЛОЙД БЛЯК",
    "StrNo": "1",
```

```
    "Block": "1",
    "En": "1",
    "Fl": "1",
    "Ap": "1",
    "PSName": "тест-нсис",
    "SOD": "2025-08-13T14:50:17",
    "RegDate": "2025-08-13T14:50:17",
    "ServiceEIK": "175192260",
    "ServiceEIKType": 0,
    "ServiceContractExpiration": "2027-07-23T23:59:59",
  - "petrol_station": {
       "GNo": "P-4914",
     + "terminals": [ … ],
     + "dispensers": [ … ],
     + "nozzles": [ … ]
    },
  - "operator": {
       "id": 1,
       "name": "Ivan Ivanov"
    },
    "operatorId": 1,
    "CT": "2025-08-13T14:50:17",
    "Status": 0,
  - "ZRD": {
       "TID": 114898,
       "SD": "2025-08-14T14:50:17",
       "Days": 30
    },
    "state": 1
}
```


service: connectivity

Modem power/status, APN, GSM connectivity recheck, HTTP proxy, and Ethernet network config.

Get current modem APN

Reads the currently active APN from the modem via AT command (`get_apn.sh`). Returns the APN string on success or an error status.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Current APN value

> **500** Failed to read APN from modem

GET /modem-apn

Response samples

200 500

Content type
application/json

Copy

```
{
    "apn": "nap"
}
```

Get modem status snapshot

Returns the raw modem status payload produced by the configured `modem_status_script`.

Application status codes (JSON `status`): `0` success, `1924` modem info unavailable. If the underlying status script returns its own error object, the endpoint propagates that payload with HTTP `503`.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Modem status returned successfully

> **503** Service dependency is temporarily unavailable.

GET /system/modem-info

Response samples

200 | 503

Content type
application/json

Copy


```
{
    "status": "ok",
    "manualDisabled": false,
    "manufacturer": "Quectel",
    "model": "EC600U",
    "imei": "867951075269106",
    "IMSI": "284050052821582",
    "rssi": 31,
    "operator": "Yettel",
    "creg_status": 1,
    "modemIp": "10.212.25.199",
    "usbIp": "10.212.25.199",
    "deviceId": "unknown"
}
```

Get persistent Ethernet startup configuration

Returns the current file-backed Ethernet startup configuration. If the config file does not exist yet, the endpoint returns the built-in default configuration that the device would use on startup.

The response reflects persistent configuration only. It does not apply the configuration immediately to the running network stack.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Network configuration returned successfully

> **500** Unexpected server-side failure.

GET /system/network-config

Response samples

200 500

Content type
application/json

Copy


```
{
    "network_mode": "custom_static",
    "eth_device": "eth0",
    "static_ip": "192.168.1.55",
    "netmask": "255.255.0.0",
    "gateway": "192.168.1.1",
    "config_path": "/customer/nsys/network.conf",
    "file_exists": true,
```

```
    "pending_restart": true
}
```

Update persistent Ethernet startup configuration

Validates and saves the Ethernet startup configuration to the persistent network config file used by the boot scripts.

The new configuration is persistent, but it is not applied immediately to the running network stack. Clients should expect `pending_restart=true` and reboot or restart the startup flow to apply the change.

If `rebootDevice=true` is passed, the response is sent first and then the device reboots automatically after ~2 seconds.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

- `rebootDevice` — boolean
  If true, the device will reboot automatically after saving the configuration.

REQUEST BODY SCHEMA: application/json
required

- `network_mode` — string, required
  Enum: `"default_static"` `"dhcp"` `"custom_static"`
  Startup network mode. `default_static` uses the built-in fallback address, `dhcp` starts DHCP, and `custom_static` uses the provided `static_ip` / `netmask` / `gateway`.
- `eth_device` — string, required
  Network interface name.
- `static_ip` — string, required
  IPv4 address used for static modes.
- `netmask` — string, required
  IPv4 netmask used for static modes.
- `gateway` — string, required
  Optional IPv4 gateway. Empty string means no default gateway.

Responses

> **200** Network configuration saved successfully

> **400** Request rejected. Most handlers return the standard status wrapper with diagnostic logs.

> **500** Unexpected server-side failure.

PUT /system/network-config

Request samples

Payload

Content type
application/json

Example
Switch to DHCP

Copy

```
{
    "network_mode": "dhcp",
    "eth_device": "eth0",
    "static_ip": "192.168.1.87",
    "netmask": "255.255.0.0",
    "gateway": ""
}
```

Response samples

200 400 500

Content type
application/json

Copy

```
{
    "network_mode": "custom_static",
    "eth_device": "eth0",
    "static_ip": "192.168.1.55",
    "netmask": "255.255.0.0",
    "gateway": "192.168.1.1",
    "config_path": "/customer/nsys/network.conf",
    "file_exists": true,
    "pending_restart": true
}
```


service: maintenance

Logs, diagnostics, reboot, app state, currency apply, and data exports (FM/KLEN/logs/SMS).

Reboot the device

Requests an immediate system reboot and returns a status/message wrapper when the reboot syscall succeeds.

Application status codes (JSON `status`): `0` success, `1901` reboot syscall failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** Reboot requested successfully

> **500** Unexpected server-side failure.

GET /system/reboot

Response samples

200 500

Content type
application/json

Copy

```
{
    "status": 0,
    "message": "Hello, world!"
}
```

Get diagnostic information

Returns a diagnostic snapshot and also prints it as a non-fiscal receipt. The exact nested diagnostic keys depend on the runtime environment and enabled collectors.

Application status codes (JSON `status`): `0` success, `1902` diagnostic context unavailable, `1903` diagnostic print failed. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

⊣ `virtual` boolean
Indicates if the receipt is virtual.

Responses

> **200** Diagnostic information returned successfully

> **500** Unexpected server-side failure.

> **503** Service dependency is temporarily unavailable.

GET /system/diagnostic

Response samples

200 | 500 | 503

Content type
application/json

Copy Expand all Collapse all


```
{
    "version": "1.0.0",
    "sha1": "0284765de7924ea507ad2bc178f506f5dda55ea7",
    "database": { },
  - "printer": {
        "can_print": true,
        "status_initizalized": true,
        "printer_drawer_connector_pin_3": true,
        "printer_is_offline": true,
        "offline_cover_open": true,
        "offline_paper_end": true,
        "has_unfinished_print_job": true,
        "offline_error": true,
        "error_recoverable_occurred": true,
        "error_autocutter_occurred": true,
        "error_unrecoverable_occurred": true,
        "error_autorecoverable_occurred": true,
        "paper_near_end": true,
        "paper_not_present": true,
        "temperature_out_of_range": true,
        "power_fail_occurred": true
    },
    "process": { },
    "network": { },
    "fiscal_memory": { },
    "modem": { },
```

```
  - "deviceState": {
        "klenMode": "NORMAL",
        "simMode": "OK",
        "gsmMode": "OK",
        "emergencyMode": "OFF",
        "capacityExhausted": false,
        "deregistered": false,
        "writesBlocked": false,
        "blockReason": "NONE"
    }
}
```


Export Fiscal Memory

Exports fiscal memory data in binary or JSON format. The default format is `json`. JSON export returns a top-level object with `provision` metadata and `records` in natural fiscal-memory order. Binary export returns the raw fiscal-memory image.

Application status codes (JSON `status`): `0` success, `1701` service context unavailable, `1702` FM backend unavailable, `1703` FM size is zero, `1704` allocation failed, `1705` response build failed, `1706` unsupported format. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

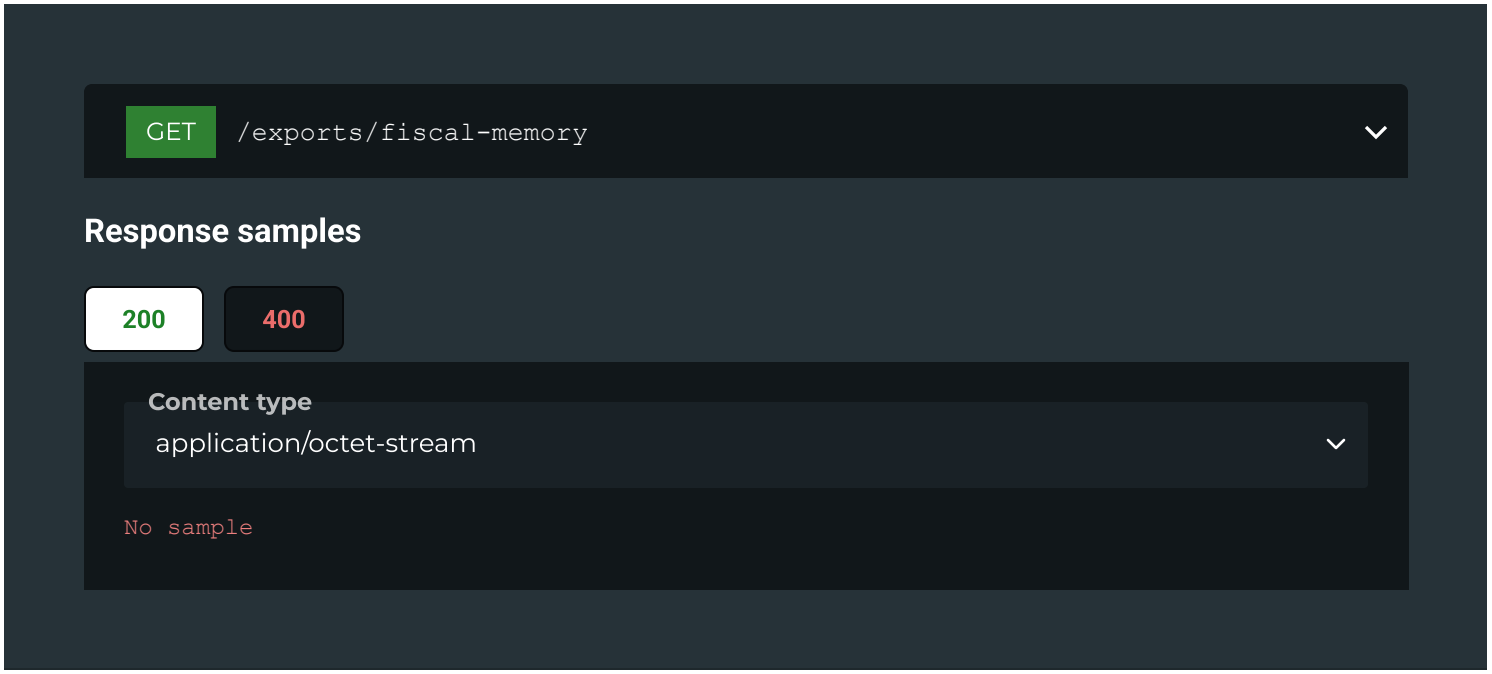
QUERY PARAMETERS

format
string
Default: `"json"`
Enum: `"binary"` `"json"`
Export format (binary returns raw FM file, json returns structured data)

Responses

> **200** Fiscal memory export successful

> **400** Bad request



Export KLEN/Journal data

Exports KLEN (journal) data as a downloadable file.

All non-binary KLEN exports are streamed in journal chunks. The service does not load the full matching journal range in memory before responding.

Binary format (`resultFormat=binary`): returns a backup copy of the full SQLite database as `nsys-fiscal.db`. Binary export intentionally ignores journal filters and always returns the complete database image.

JSON format (`resultFormat=json` or no explicit result format): streams matching journal entries as a JSON array and returns it as `klen-export.json`.

CSV format (`resultFormat=csv`): streams KLEN-style sale rows as `klen-export.csv`. CSV rows are produced for fiscal receipts (`JOURNAL_FISCAL_RECEIPT`), invoices (`JOURNAL_INVOICE`), and storno receipts (`JOURNAL_STORNO_RECEIPT`). Each line represents one item with the following columns:

- ФУ_Номер: Fiscal unit number
- ФБ_Номер: Fiscal receipt number
- УНП: Unique sales number
- Стока_Услуга_Наименование: Item/service name
- Стока_Услуга_Единична_Цена: Unit price
- Стока_Услуга_Количество: Quantity
- Стока_Услуга_Стойност: Item value

- Обща_Сума: Total receipt amount
- Номер_Фактура: Invoice number (for invoices only)
- ЕИК_Получател: Recipient EIK (for invoices only)
- Номер_Сторниран_ФБ: Stornoed receipt number (for storno only)
- Номер_Сторнирана_Фактура: Stornoed invoice number (for storno only)
- Причина_Сторниране: Storno reason (for storno only)

Formatted text (`resultFormat=formatted_text`): streams printable journal text in the same result-format style as `GET /journal?resultFormat=formatted_text` and returns it as `klen-export.txt`.

JSON, CSV, and formatted-text exports support the same journal filters as `GET /journal`: `fromId` / `toId`, `fromDateTime` / `toDateTime`, and `type`. ID range and date-time range are mutually exclusive. If no ID or date-time range is provided, the export streams all matching journal entries.

KLEN exports do not print a journal report receipt and do not intentionally affect KLEN, receipt, printer, or fiscal state.

Application status codes (JSON `status`): `0` success, `1601` invalid query parameters, `1701` service context unavailable, `1707` DB unavailable, `1708` backup/read failure, `1704` allocation failure, `1705` response build failure, `1706` unsupported format. See `#/components/schemas/StatusCode` for full mapping.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

QUERY PARAMETERS

resultFormat
string
Default: `"json"`
Enum: `"json"` `"csv"` `"formatted_text"` `"binary"`
Journal-style result format selector, aligned with `GET /journal`. `binary` is the KLEN export extension that returns the complete SQLite database backup and ignores journal filters.

fromId
integer `>= 0`
Inclusive journal entry ID lower bound. Use with `toId`; cannot be combined with `fromDateTime` / `toDateTime`.

toId
integer `>= 0`
Inclusive journal entry ID upper bound. Use with `fromId`; cannot be combined with `fromDateTime` / `toDateTime`.

fromDateTime
string <date-time>
Example: `fromDateTime=2026-02-10T00:00:00`
Inclusive local journal timestamp lower bound. Use with `toDateTime`; cannot be combined with `fromId` / `toId`.

toDateTime

string <date-time>

Example: `toDateTime=2026-02-10T23:59:59`

Inclusive local journal timestamp upper bound. Use with `fromDateTime`; cannot be combined with `fromId` / `toId`.

---

type

string

Enum: `"JOURNAL_FISCAL_RECEIPT"` `"JOURNAL_INVOICE"` `"JOURNAL_NON_FISCAL_RECEIPT"` `"JOURNAL_DUPLICATE_RECEIPT"` `"JOURNAL_Z_REPORT"` `"JOURNAL_X_REPORT"` `"JOURNAL_OPERATORS_REPORT"` `"JOURNAL_ARTICLES_REPORT"` `"JOURNAL_VAT_CHANGES"` `"JOURNAL_CASH_IN"` `"JOURNAL_CASH_OUT"` `"JOURNAL_STORNO_RECEIPT"` `"JOURNAL_REGISTRATION"` `"JOURNAL_CHANGE_REGISTRATION"` `"JOURNAL_DEACTIVATION"` `"JOURNAL_FM_DETAILED_REPORT"` `"JOURNAL_FM_SHORT_REPORT"` `"JOURNAL_REPORT"` `"JOURNAL_CONTRACTS_REPORT"` `"JOURNAL_ALL"`

Journal entry type filter. Defaults to `JOURNAL_ALL` when omitted.

---

virtual

string

Default: `"false"`

Enum: `"true"` `"false"`

When `true`, restricts the export to virtual receipts only (journal entries created in virtual mode). Any other value (or omission) exports both virtual and real entries. Ignored for `resultFormat=binary`, which returns the full database backup.

Responses

> **200** Successfully exported KLEN data as a downloadable attachment.

> **400** Invalid resultFormat value or query parameter combination.

— **500** Internal server error

GET /exports/klen

Response samples



Export everything a support case needs, as one archive

Returns a single `.tar.gz` bundling the three artefacts that are otherwise separate exports:

- the KLEN database, copied the same consistent way as the daily backup, so the bundle never contains a half-written file;
- the fiscal-memory image, the same bytes `/exports/fiscal-memory?format=binary` serves;
- the service logs, under `logs/`;
- the SMS store, under `sms/`.

Assembled in a staging directory next to the other archive exports. Files are hard-linked when they share a filesystem with the temp directory, so the bundle costs no extra space for them — which matters on a card that also holds the KLEN.

Everything except the KLEN is best effort: a missing fiscal-memory backend, log directory or SMS store is logged and left out, because a partial bundle still helps. Only a failure to copy the KLEN fails the request.

Note this is a large, blocking export — the database copy holds a read lock for its duration, so concurrent fiscal writes wait behind it.

AUTHORIZATIONS: > *ApiKeyHeader* or *ApiKeyQuery*

Responses

> **200** The bundle

> **400** The bundle could not be produced

> **503** Service dependency is temporarily unavailable.

GET /exports/all

Response samples

400 503

Content type
application/json

Copy Expand all Collapse all


```
{
   "status": 1001,
   "message": "Failed to parse JSON",
 - "logs": [
      "string"
   ]
}
```