Изображение выглядит как знак

Автоматически созданное описание

HTTP Database API

Version 1.00

Contents

[Overview 2](#_Toc36397487)

[URL 2](#_Toc36397488)

[Requirements for requests 2](#_Toc36397489)

[Responses 2](#_Toc36397490)

[Operations 2](#_Toc36397491)

[SELECT 3](#_Toc36397492)

[INSERT / UPSERT 4](#_Toc36397493)

[UPDATE 5](#_Toc36397494)

[DELETE 6](#_Toc36397495)

# Overview

The Database API provides an interface to interact with the 5gVision database via HTTP requests with a JSON payload.

# URL

<https://host/instance_name/db_api/>

# Requirements for requests

All requests should contain a JSON-encoded payload and use the Content-Type header as

Content-Type: application/json

# Responses

The API uses conventional HTTP response codes.

Code in the 2xx range indicates success.

If the request assumes the presence of data in the response, then the data will be in the JSON format.

Code in the 4xx and 5xx ranges indicates failure.

The response body can contain error description in the JSON format:

{

"error": "<error description>"

}

# Operations

The API supports the following database operations:  SELECT, INSERT, UPSERT, UPDATE, DELETE.

Access to the table and its columns is made through identifiers that can be obtained by Alt+Right-click on a column name in the Web interface.

Изображение выглядит как снимок экрана

Автоматически созданное описание

## SELECT

Supported HTTP methods: GET, POST.

Format:

{

"operation": "select",

"table\_id": table\_id,

"aux\_table\_id": aux\_table\_id,

"column\_ids": [col\_id, ...] |

{

"col\_id": "alias",

"col\_id": "" | null,

...

},

"filter": "condition",

"order": "col\_id asc | desc",

"limit": number

}

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Value type | Mandatory | Description |
| operation | string | yes | Operation type |
| table\_id | number | yes | Table ID |
| aux\_table\_id | number | no | Additional table ID (such a user ID, invoice ID, ...) needed for some tables |
| column\_ids | array | no | Selected column IDs, if absent - all columns will be selected |
| object | no | Mapping selected column IDs to aliases that will be used in a response, if absent - all columns will be selected |
| filter | string | no | Condition to filter rows, possible to use column IDs as {{col\_id}} |
| order | string | no | Column ID to sort rows and sorting method ('asc' by default) |
| limit | number | no | Maximum number of the selected rows |

Request example:

{

"operation": "select",

"table\_id": 609,

"column\_ids": [60900, 60910],

"filter": "{{60910}} = 2024410000",

"order": "60900",

"limit": 1

}

Response example:

[

    {

        "60900": "2019174",

        "60910": "2024410000"

    }

]

## INSERT / UPSERT

Supported HTTP methods: POST, PUT.

Format:

{

"operation": "insert",

"table\_id": table\_id,

"aux\_table\_id": aux\_table\_id,

"rows": [

{

"col\_id": "value",

...

},

...

],

"conflict\_target": "conflict\_target",

"conflict\_action": "conflict\_action",

"get\_affected\_rows\_num": true,

"return": [col\_id, ...] |

{

"col\_id": "alias",

"col\_id": "" | null,

...

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Value type | Mandatory | Description |
| operation | string | yes | Operation type |
| table\_id | number | yes | Table ID |
| aux\_table\_id | number | no | Additional table ID (such a user ID, invoice ID, ...) needed for some tables |
| rows | array | yes | Rows to insert (or update in case of UPSERT) |
| conflict\_target | string | no | Specifies conflicts (for example, column ID with unique constraint), used for UPSERT only |
| conflict\_action | string | no | Specifies action for conflict (for example, "update set {{col\_id}} = excluded.{{col\_id}}", "nothing"), used for UPSERT only |
| get\_affected\_rows\_num | boolean | no | Flag to return the number of rows affected by INSERT/UPSERT in a response, ignored if the "return" parameter is set |
| return | array | no | Column IDs to return inserted (or updated in case of UPSERT) values in a response |
| object | no | Mapping column IDs to aliases to return inserted (or updated in case of UPSERT) values in a response |

Request example (UPSERT):

{

"operation": "insert",

"table\_id": 609,

"conflict\_target": "60900",

"conflict\_action": "update set {{60910}} = excluded.{{60910}}",

"rows": [

{

"60900": 2028609,

"60910": 2023409210

},

{

"60900": 2028804,

"60910": 2024410000

}

],

"get\_affected\_rows\_num": true

}

Response example:

{

    "affected\_rows\_num": 2

}

## UPDATE

Supported HTTP methods: POST, PUT.

Format:

{

"operation": "update",

"table\_id": table\_id,

"aux\_table\_id": aux\_table\_id,

"values": {

"col\_id": "value",

...

},

"filter": "condition",

"get\_affected\_rows\_num": true,

"return": [col\_id, ...] |

{

"col\_id": "alias",

"col\_id": "" | null,

...

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Value type | Mandatory | Description |
| operation | string | yes | Operation type |
| table\_id | number | yes | Table ID |
| aux\_table\_id | number | no | Additional table ID (such a user ID, invoice ID, ...) needed for some tables |
| values | object | yes | Values to update |
| filter | string | no | Condition to filter records, possible to use column IDs as {{col\_id}} |
| get\_affected\_rows\_num | boolean | no | Flag to return the number of rows affected by UPDATE in a response, ignored if the "return" parameter is set |
| return | array | no | Column IDs to return updated values in a response |
| object | no | Mapping column IDs to aliases to return updated values in a response |

Request example:

{

"operation": "update",

"table\_id": 609,

"values": {

"60910": 2023409210

},

"filter": "{{60900}} = 2028804",

"get\_affected\_rows\_num": true

}

Response example:

{

    "affected\_rows\_num": 1

}

## DELETE

Supported HTTP methods: DELETE, POST.

Format:

{

"operation": "delete",

"table\_id": table\_id,

"aux\_table\_id": aux\_table\_id,

"filter": "condition",

"get\_affected\_rows\_num": true,

"return": [col\_id, ...] |

{

"col\_id": "alias",

"col\_id": "" | null,

...

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Value type | Mandatory | Description |
| operation | string | yes | Operation type |
| table\_id | number | yes | Table ID |
| aux\_table\_id | number | no | Additional table ID (such a user ID, invoice ID, ...) needed for some tables |
| filter | string | no | Condition to filter records, possible to use column IDs as {{col\_id}} |
| get\_affected\_rows\_num | boolean | no | Flag to return the number of rows affected by DELETE in a response, ignored if the "return" parameter is set |
| return | array | no | Column IDs to return deleted values in a response |
| object | no | Mapping column IDs to aliases to return deleted values in a response |

Request example:

{

"operation": "delete",

"table\_id": 609,

"filter": "{{60910}} = 2023409210",

"return": {

"60900": "DST",

"60910": "LRN"

}

}

Response example:

[

    {

        "DST": "2028609",

        "LRN": "2023409210"

    },

    {

        "DST": "2028804",

        "LRN": "2023409210"

    }

]