

DEVI project

Wojciech Jędrzejewski

May 18, 2023

Chapter 1

Introduction

Chapter 2

Database

Chapter 3

Error pages

3.1 Error 418 Teapot

route: `/teapot`

This error shows up when the HTCPCP requests returns an error.

Chapter 4

Commands

4.1 Admin

Commands to administrate system. **NOT for user!**

4.1.1 /admin/addNewCompany

Description

Add new company to system.

METHOD: PUT

Body parameters

name: name of company and database

email: company email

user: user name for first user

pass: password for first user

key: admin key

Return: NULL

Example body:

```
1  {
2      "name": "myCompany",
3      "email": "mycompanygmail.com",
4      "user": "admin",
5      "pass": "1qazXSW@",
6      "key": "<PASTEadminKEYhere>"
7  }
```

4.2 SIN

SIN (Securry Indentyfication Number) - token to indentyfication user. It store user ip, user browser agent, user name and DB name. This number have 5 minutes time of live. Every call of **/getSIN** extends this time by next 5 minutes. You may have only one SIN per user.

4.2.1 /getSIN

Description

Get SIN from server or when you have it extend it expire time.

METHOD: POST

Haeder parameters

User-Agent: browser name (It's automaticly added by our browser)

Body parameters

user: user name

pass: user password

db: admin key

Return:

sin: your SIN

Example body:

```
1  {
2      "user": "admin",
3      "pass": "1qazXSW@",
4      "db": "myCompany"
5  }
```

Example response:

```
1  {
2      "sin": "<yourSIN>"
3  }
```

4.2.2 /dropSIN

Description

Drop your SIN from server. It work like logout.

METHOD: DELETE

Body parameters

SIN: SIN number

Return: NULL

Example body:

```

1  {
2      "sin": "<yourSIN>"
3  }

```

4.3 Tables

4.3.1 /tables

Description

Returns the names of all tables.

METHOD: POST

Body parameters

SIN: SIN number

Return:

tables: array with tables names

Example body:

```

1  {
2      "sin": "<yourSIN>"
3  }

```

Example response:

```

1  {
2      "tables": [
3          "clients",
4          "employees",
5          "products"
6      ]
7  }

```

4.3.2 /tables/add

Description

Add new table.

Name parameter cannot start with "log_" or "system_" or "dic_"

METHOD: PUT

Body parameters

name: table name

SIN: SIN number

Return: NULL

Example body:

```

1  {
2      "name": "suppliers",
3      "sin": "<yourSIN>"
4  }

```

4.3.3 /tables/<table>/addColumn

Description

Add new column to table.

'ID' is not allowed name of column.

Allowed types:

- INT - integer
- TEXT - text
- DATETIME - date and time
- TIME - time
- DATE - date
- FLOAT - floating point number
- BOOL - true/false value
- KEY - key for relations with other table

METHOD: PUT

URL parameters

table: table name

Body parameters

name: column name

type: column type.

SIN: SIN number

***reference:** name table of reference. Only for KEY

* - optional

Return: NULL

Example body:

```

1  {
2      "name": "price",
3      "type": "INT",
4      "sin": "<yourSIN>"
5  }

```



```

1  {
2      "name": "client",
3      "type": "KEY",
4      "references": "clients",
5      "sin": "<yourSIN>"
6  }

```

4.3.4 /tables/<table>/insert

Description

Insert new row into table

METHOD: PUT

URL parameters

table: table name

Body parameters

data: assoc array key is named like column

SIN: SIN number

The name of the column you are inserting the value into is name of parameter.

You can insert more than one value, but only one row. Values you do not send will be set to NULL.

Return: NULL

Example body:

```

1  {
2      "data":
3      {
4          "name": "Bob",
5          "lastName": "Smith",
6          "phone": "123987465"
7      },
8      "sin": "<yourSIN>"
9  }

```

4.3.5 /tables/<table>/update/<columnID>

Description

Update row.

METHOD: POST

URL parameters

table: table name

columnID: column ID

Body parameters

SIN: SIN number

data: assoc array key is named like column

Return: NULL

Example body:

```
1  {
2      "data":
3      {
4          "name": "Bob",
5          "lastName": "Smith",
6          "phone": "987123456 "
7      },
8      "sin": "<yourSIN>"
9  }
```

4.3.6 /tables/<table>/delete/<columnID>

Description

Delete row.

METHOD: DELETE

URL parameters

table: table name

columnID: column ID

Body parameters

SIN: SIN number

Return: NULL

Example body:

```
1  {
2      "sin": "<yourSIN>"
3  }
```

4.3.7 /tables/<table>/select/<columnID>

OUTDATED

Description

Select row.

METHOD: POST

URL parameters

table: table name

columnID: column ID

Body parameters

SIN: SIN number

Return: Name of cell is name of parameter

4.3.8 /tables/<table>/select/all

Description

Select all row from table.

METHOD: POST

URL parameters

table: table name

Query parameters

***limit:** records per page

***page:** page number

***orderBy:** column name

***orderByType:** ASC/DESC

***** - optional

Body parameters

SIN: SIN number

***query:** array: key - column name; value - your parameter

Return:

Legend: array contain name of all columns

Types: array contain type of all columns. Each value is array with values order like columns at Legend parameter.

Next parameters name 'row' and it number (starts from 0). Each value is array with values order like columns at Legend parameter.

Example body:

```
1  {
2      "sin": "<yourSIN>"
3  }
```

```
1  {
2      "query":
3      {
4          "name": "Bob",
5          "lastName": "Smith"
6      },
7      "sin": "<yourSIN>"
8  }
```

Example response:

```
1  {
2      "Legend": [
3          "name",
```

```

4         "lastName",
5         "phone"
6     ],
7     "Types": [
8         "TEXT",
9         "TEXT",
10        "INT"
11    ],
12    "row0": [
13        "Bob",
14        "Smith",
15        "123098822"
16    ],
17    "row1": [
18        "Anna",
19        "Smith",
20        "123788822"
21    ],
22    "row2": [
23        "Bob",
24        "Smith",
25        "111222333"
26    ],
27    "row4": [
28        "Tom",
29        "White",
30        "111222333"
31    ],
32 }

```

4.3.9 /tables/<table>/alter

Description

Change name and type of column.

METHOD: POST

URL parameters

table: table name

Body parameters

name: actual name of column

newName: new name of column. If you would not change name send actual

newType: new type of column. If you would not change type send actual

SIN: SIN number

Return: NULL

Example body:

```

1  {
2      "name": "pirec",
3      "newName": "price",
4      "newType": "INT",
5      "sin": "<yourSIN>"
6  }

```

4.3.10 /tables/<table>/rename

Description

Rename table.

METHOD: POST

URL parameters

table: table name

Body parameters

name: new name of table

SIN: SIN number

Return: NULL

Example body:

```

1  {
2      "name": "products",
3      "sin": "<yourSIN>"
4  }

```

4.3.11 /tables/<table>/dropColumn

Description

Delete column.

METHOD: DELETE

URL parameters

table: table name

Body parameters

name: name of column

SIN: SIN number

Return: NULL

Example body:

```

1  {
2      "name": "phone",
3      "sin": "<yourSIN>"
4  }

```

4.3.12 /tables/<table>/dropTable

Description

Delete table.

METHOD: DELETE

URL parameters

table: table name

Body parameters

SIN: SIN number

Return: NULL

Example body:

```
1  {
2      "sin": "<yourSIN>"
3  }
```

4.3.13 /tables/DictionaryesAdd

Description

Add dictionaryes table.

METHOD: PUT

Body parameters

name: table name

SIN: SIN number

Return: NULL

Example body:

```
1  {
2      "name": "status",
3      "sin": "<yourSIN>"
4  }
```

4.4 Search

4.4.1 /search/<phrase>

Description

Search phrase in all tables.

METHOD: POST

URL parameters

phrase: phrase

Body parameters

SIN: SIN number

Return:

Returning array named as table containing data like /tables/<table>/select/all

Example body:

```
1  {
2      "sin": "<yourSIN>"
3  }
```

Example response:

```
1  {
2      "clients":{
3          "Legend": [
4              "name",
5              "lastName",
6              "phone"
7          ],
8          "Types": [
9              "TEXT",
10             "TEXT",
11             "INT"
12         ],
13         "row0":[
14             "Bob",
15             "Smith",
16             "123098822"
17         ],
18         "row2":[
19             "Bob",
20             "Smith",
21             "111222333"
22         ]
23     },
24     "suppliers":{
25         "Legend": [
26             "name",
27             "lastName",
28             "phone",
29             "company",
30             "email"
31         ],
32         "Types": [
33             "TEXT",
34             "TEXT",
35             "INT"
36         ],
37         "row0":[
38             "Bob",
39             "Smith",
```

```

40         "123098822",
41         "superFactory",
42         "contact@superFactory.com"
43     ],
44 },
45 }

```

4.5 Client

Commands for managing client account

4.5.1 /client/changeName

Description

Changing name for your. **After executing this command you will be logged out!**

METHOD: POST

Body parameters

name: new name for your account

SIN: SIN number

Return: NULL

Example body:

```

1  {
2      "name": "bob",
3      "sin": "<yourSIN>"
4  }

```

4.5.2 /client/changePass

Changing password for your. **After executing this command you will be logged out!**

METHOD: POST

Body parameters

pass: new password for your account

SIN: SIN number

Return: NULL

Example body:

```

1  {
2      "pass": "@WSXzaq1",
3      "sin": "<yourSIN>"

```



```
4      }
```

4.6 Sheet

4.6.1 /sheets

Description

Returns the names of all sheets.

METHOD: POST

Body parameters

SIN: SIN number

Return:

tables: array with sheets names

Example body:

```
1      {
2          "sin": "<yourSIN>"
3      }
```

Example response:

```
1      {
2          "sheets": [
3              "clients",
4              "employees",
5              "products"
6          ]
7      }
```

4.6.2 /sheet/<sheet>/update

Update cell in sheet.

METHOD: POST

URL parameters

sheet: sheet name

Body parameters

coord: coordinate of cell

value: cell value

SIN: SIN number

Return: NULL

Example body:

```

1  {
2      "coord": "A2",
3      "value": "=SUM(`products`.`price`)" ;
4      "sin": "<yourSIN>"
5  }

```

4.6.3 /sheet/<sheet>/get

Get sheet with resolved DB functions.

METHOD: POST

URL parameters

sheet: sheet name

Body parameters

SIN: SIN number

Return:

data: array contain calculated values from DB

sheet: array contain all cells

Example body:

```

1  {
2      "sin": "<yourSIN>"
3  }

```

```

1  {
2      "data": {
3          "SUM(`products`.`price`)": 1353.5,
4          "SUM_IF(`products`.`price`,<100)": 630.4,
5          "COUNT(`products`)" : 50.0,
6          "AVG(`products`.`price`)" : 27.07,
7          "MAX(`products`.`price`)" : 310.0,
8          "MIN(`products`.`price`)" : 5.0,
9          "COUNT_IF(`products`.`price`,>300)": 1.0
10     },
11     "sheet": {
12         "A1": "=SUM(`products`.`price`)",
13         "B1": "=SUM_IF(`products`.`price`,<100)/AVG(`
14             ↳ products`.`price`)",
15         "C2": "17",
16         "C1": "=CELI(MAX(`products`.`price`)/2) + B1",
17         "B5": "=MAX(`products`.`price`)/MIN(`products`.`
18             ↳ price`)",
19         "D1": "=MIN(`products`.`price`)",
20         "A2": "=COUNT_IF(`products`.`price`,>300)"
21     }
22 }

```