

CaoSQLWebAPI

Version 1.1.3

ユーザーズ ガイド

March 29, 2021

備考:

【改版履歴】

| バージョン | 日付 | 内容 |
|-------|-----------|--|
| 1.0.0 | 2019-2-11 | 初版. |
| 1.1.0 | 2019-5-30 | バイナリのアップロード/ダウンロード機能を追加 |
| 1.1.2 | 2020-1-29 | ・値の取得時にバイト配列を Number の配列で返すように修正 ・値の取得時にバイト配列を Base64 の文字列で返すオプションを追加 |
| 1.1.3 | 2021-3-29 | 読み取り専用のアイテムに対して putvalue, putvalues を実行したときにエラーになるように修正. |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

【対応機器】

| 機種 | バージョン | 注意事項 |
|----|-------|------|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

目次

| | |
|-------------------------------|----|
| 1. はじめに | 4 |
| 2. 概要 | 5 |
| 2.1. インストール..... | 5 |
| 2.2. 設定 | 5 |
| 2.2.1. ポートの指定..... | 5 |
| 2.2.2. セキュア通信(HTTPS)の設定 | 5 |
| 2.2.3. セキュリティトークンの指定 | 6 |
| 2.3. 起動..... | 6 |
| 3. CaoSQLWebAPI のコマンド..... | 7 |
| 3.1. コマンドの指定方法..... | 7 |
| 3.2. コマンドの戻り値 | 7 |
| 3.3. エラー時の戻り値..... | 7 |
| 3.4. コマンド一覧 | 8 |
| 3.4.1. コントローラー一覧の取得 | 8 |
| 3.4.2. アイテム一覧の取得..... | 8 |
| 3.4.3. アイテム値の取得 | 9 |
| 3.4.4. アイテム値の一括取得 | 10 |
| 3.4.5. アイテム値設定 | 11 |
| 3.4.6. アイテム値の一括設定 | 12 |
| 3.4.7. バイナリイメージのアップロード | 13 |
| 3.4.8. バイナリイメージのダウンロード | 13 |
| 4. ライセンス表示 | 14 |
| 4.1. Json.NET | 14 |

1. はじめに

CaoSQLWebAPI は CaoSQL の操作を HTTP プロトコルで行うための Web サービスです。

CaoSQL のプロジェクトは IoT Data Share または IoT Data Server で作成されたプロジェクトが対応です。

※CaoSQLConfig で作成されたプロジェクト(*.csq)は動作対象外です。

2. 概要

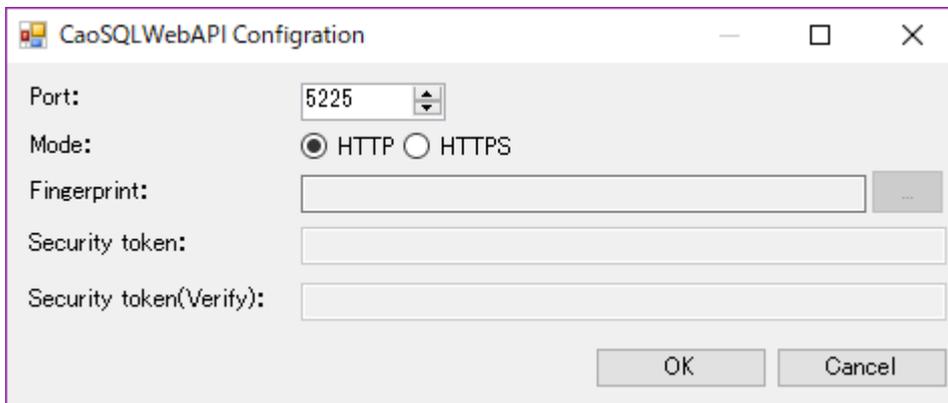
2.1. インストール

CaoSQLWebAPI をインストールするには ORiN2 SDK をインストールしたフォルダの下にある CaoSQLWebAPI\Bin\install.bat を実行してください。

CaoSQLWebAPI をアンインストールするには ORiN2 SDK をインストールしたフォルダの下にある CaoSQLWebAPI\Bin\uninstall.bat を実行してください。

2.2. 設定

CaoSQLWebAPI の設定は ORiN2 SDK をインストールしたフォルダの下にある CaoSQLWebAPI\Bin\CaoSQLWebConfig.exe を使用して行います。



設定を行った後に OK ボタンを押すと設定が CaoSQLWebAPI.config ファイルに保存されますが、直接編集しないでください。

保存した設定を適用するにはサービスの再起動が必要です。

設定ファイルのフォルダパスは環境変数 CAOSQLWEBAPI_CONFIG で指定可能です。設定されていない場合は CaoSQLWebConfig.exe と同じカレントフォルダになります。

2.2.1. ポートの指定

CaoSQLWebAPI はデフォルトで 5225 番のポートを使用して通信を行います。

通信に使用するポートを変更したい場合は Port の値を変更してください。

2.2.2. セキュア通信(HTTPS)の設定

CaoSQLWebAPI は初期設定の状態では HTTP プロトコルを用いて通信を行います。

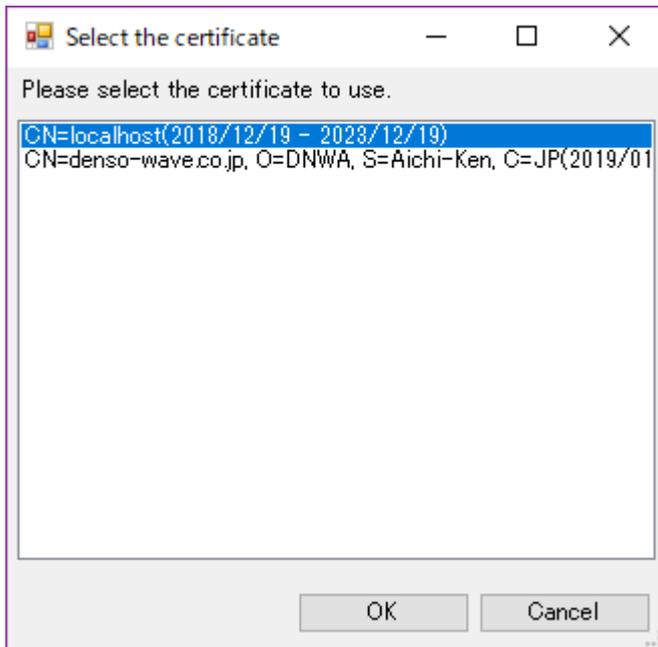
Mode を HTTPS に変更することで HTTPS プロトコルでの通信が可能になります。

セキュア通信を行うためには以下の設定が必要です。

- pfx 形式のサーバー証明書を証明書ストアのローカルコンピューターにインストールする必要がある

あります。

- ▶ Windows 付属の CertUtil.exe を使用すると簡単にインストールを行うことができます。
- インストールした証明書の拇印を **Fingerprint** に設定する必要があります。
 - ▶ **Fingerprint** の右側にあるボタンを押すとインストールされている証明書の一覧が表示されますので、使用する証明書を選択して **OK** を押してください。



- 自己認証を行った証明書はセキュリティ上問題が発生する可能性がありますので、**HTTPS** プロトコルを用いる場合は公式の認証機関に認証を受けた証明書を購入することをお勧めします。

2.2.3. セキュリティアークンの指定

HTTPS モードの時はアクセスを制限するためのセキュリティアークンを指定することができます。

セキュリティアークンを使用する場合は **Security token** および **Security token(Verify)** に同じ文字列を指定してください。

セキュリティアークンを指定した CaoSQLWebAPI にアクセスする場合はリクエストのヘッダに以下の項目を指定してください。

Authorization: Bearer <Base64 でエンコードした UTF-8 のセキュリティアークン>

2.3. 起動

CaoSQL を起動するには管理ツールのサービスから CaoSQLWebAPI を開始するか、管理者権限で起動したコマンドプロンプトから以下のコマンドを実行してください。

```
sc start CaoSQLWebAPI
```

3. CaoSQLWebAPI のコマンド

3.1. コマンドの指定方法

CaoSQLWebAPI のベースアドレスは HTTP プロトコルの場合は

```
http://<サーバーアドレス>/api/v1/
```

HTTPS プロトコルの場合は

```
https://<サーバーアドレス>/api/v1/
```

となります。

また、WEB メソッドとして GET と POST をサポートします。

ただし、一部機能に関しては GET のみサポートするものと POST のみをサポートするものがあります。

GET 時の引数は URL に続けて?aaa=bbb&ccc=ddd の形式で指定してください。

upload 以外の POST 時の引数は本文に json 形式で指定してヘッダの Content-Type に”application/json”を指定してください。

upload の場合は本文にアップロードするデータのバイナリイメージを格納して Content-Type は”application/octet-stream”を指定してください。

3.2. コマンドの戻り値

download 以外のコマンドの戻り値は json となります。

download のコマンドの戻り値はバイナリイメージとなります。

値の意味は各コマンドの項目を参照してください。

3.3. エラー時の戻り値

CaoSQL への接続がエラーになった場合、download 以外は以下の形の戻り値が返ります。

| | | |
|--------------|---|--------------------------------|
| result | : | コマンドの実行結果. ”OK”:成功, ”NG”:失敗 |
| errormessage | : | エラーメッセージ(NG の時のみ) |
| errorcode | : | エラーコード(COM の HRESULT. NG の時のみ) |

例)

```
{"result":"NG","errormessage":"対象のコンピューターによって拒否されたため、接続できませんでした。127.0.0.1:5191","errorcode":-2147467259}
```

download の場合はステータスコード 200 が成功、それ以外はエラーを表します。

3.4. コマンド一覧

3.4.1. コントローラー一覧の取得

URL : getcontrollernames
メソッド : GET : ○ POST : ×
GET 時の引数 : なし
例)
http://localhost:5225/api/v1/getcontrollernames
POST 時の引数 : -
戻り値 : result : コマンドの実行結果. "OK":成功, "NG":失敗
names : コントローラー名の配列
例)
{ "names":["COLLECT","FL","SET","STR_TEST"],"result":"OK" }

3.4.2. アイテム一覧の取得

URL : getitemnames
メソッド : GET : ○ POST : ○
GET 時の引数 : controller=<コントローラー名>
例)
http://localhost:5225/api/v1/getitemnames?controller=COLLECT
POST 時の引数 : ヘッダの Content-Type に" application/json"を指定し、本文を以下の形の json で指定してください。
{ "controller":<コントローラー名> }
例)
{ "controller":"COLLECT" }
戻り値 : result : コマンドの実行結果. "OK":成功, "NG":失敗
names : アイテム名の配列
例)
{ "names":["BOOL","DOUBLE","FLOAT","LONG","LONGLONG","SHORT","STRING","STRING2","ULONG","ULONGLONG","USHORT","WSTRING"],"result":"OK" }

3.4.3. アイテム値の取得

URL : getvalue

メソッド : GET : ○ POST : ○

GET 時の引数 : controller=<コントローラ名>&item=<アイテム名>&base64=<true または false>

base64 は省略可能で、省略時は false になります。

base64 に true を指定するとバイト配列を Base64 の文字列として返します。

例)

`http://localhost:5225/api/v1/getvalue?controller=COLLECT&item=LONG`

POST 時の引数 : ヘッダの Content-Type に” application/json”を指定し、本文を以下の形の json で指定してください。

```
{"controller":<コントローラ名>,"item":<アイテム名>,"base64":<true または false>}
```

base64 は省略可能で、省略時は false になります。

base64 に true を指定するとバイト配列を Base64 の文字列として返します。

例)

```
{"controller":"COLLECT","item":"LONG"}
```

戻り値 :

| | | |
|-----------|---|---|
| result | : | コマンドの実行結果. ”OK”:成功, ”NG”:失敗 |
| success | : | 値取得(CaoSQL 上の get_Value)の実行結果 |
| errorcode | : | get_Value のエラーコード(success が false の時のみ) |
| value | : | アイテムの値(success が true の時のみ) |

例)

```
{"value":3,"result":"OK","success":true}
```

3.4.4. アイテム値の一括取得

- URL : getvalues
 メソッド : GET : × POST : ○
 GET 時の引数 : -
 POST 時の引数 : ヘッダの Content-Type に” application/json”を指定し、本文を以下の形の json で指定してください。

```
{"controller":<コントローラ名>,"item":<アイテム名>,"base64":<true または false>}の配列
```

base64 は省略可能で、省略時は false になります。

base64 に true を指定するとバイト配列を Base64 の文字列として返します。

例)

```
[{"controller":"COLLECT","item":"LONG","base64":true}, {"controller":"COLLECT","item":"SHORT"}]
```

- 戻り値 :
- result : コマンドの実行結果. ”OK”:成功, ”NG”:失敗
 - items : 以下の要素の配列(引数で指定したものと対応する).
 - controller : 取得元のコントローラ名
 - item : 取得元のアイテム名
 - success : 値取得(CaoSQL 上の get_Value)の実行結果
 - errorcode : get_Value のエラーコード(success が false の時のみ)
 - value : アイテムの値(success が true の時のみ)

例)

```
{"items":[{"value":3,"controller":"COLLECT","item":"LONG","success":true}, {"value":null,"controller":"COLLECT","item":"SHORT2","success":false,"errorcode":-2147024809}], "result":"OK" }
```

3.4.5. アイテム値設定

- URL : putvalue
- メソッド : GET : × POST : ○
- GET 時の引数 : -
- POST 時の引数 : ヘッダの Content-Type に” application/json”を指定し、本文を以下の形の json で指定してください。
- ```
{"controller":<コントローラ名>,"item":<アイテム名>,"value":<値>}
```
- 例)
- ```
{"controller":"SET","item":"LONG","value":10}
```
- 戻り値 :
- result : コマンドの実行結果. ”OK”:成功, ”NG”:失敗
 - success : 値設定(CaoSQL 上の put_Value)の実行結果
 - errorcode : get_Value のエラーコード(success が false の時のみ)
- 例)
- ```
{"result":"OK","success":true}
```
- 備考 : 属性が読み取りのアイテムに対して値を設定しようとすると設定に失敗し, success に false が戻ります.

### 3.4.6. アイテム値の一括設定

- URL : putvalues
- メソッド : GET : × POST : ○
- GET 時の引数 : -
- POST 時の引数 : ヘッダの Content-Type に” application/json”を指定し、本文を以下の形の json で指定してください。
- {"controller":<コントローラ名>,"item":<アイテム名>,"value":<アイテム値>}の配列  
例)
- [{"controller":"SET","item":"LONG","value":10},{ "controller":"COLLECT","item":"SHORT","value":20}]
- 戻り値 : result : コマンドの実行結果. ”OK”:成功, ”NG”:失敗
- items : 以下の要素の配列(引数で指定したものと対応する).
- controller : 設定先のコントローラ名
- item : 設定先のアイテム名
- success : 値設定(CaoSQL 上の put\_Value)の  
実行結果
- errorcode : get\_Value のエラーコード(success  
が false の時のみ)
- 例)
- {"items":[{"controller":"SET","item":"LONG","success":true},{ "controller":"SET","item":"SHORT2","success":false,"errorcode":-2147024809}], "result":"OK"}
- 備考 : 属性が読み取りのアイテムに対して値を設定しようとすると設定に失敗し, success に false が戻ります.

### 3.4.7. バイナリイメージのアップロード

|           |   |                                                                                                                                                                                    |
|-----------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| URL       | : | upload?controller=<コントローラ名>&item=<アイテム名>                                                                                                                                           |
| メソッド      | : | GET : × POST : ○                                                                                                                                                                   |
| GET 時の引数  | : | -                                                                                                                                                                                  |
| POST 時の引数 | : | ・URL のパラメータにアップロード先のコントローラ名/アイテム名を指定してください。<br>・ヘッダの Content-Type に” application/octet-stream”を指定してください。<br>・本文にアップロードするバイナリイメージを格納してください。<br>バイナリイメージはヘッダを含めて 10MB までアップロードが可能です。 |
| 戻り値       | : | result : コマンドの実行結果. ”OK”:成功, ”NG”:失敗                                                                                                                                               |

例)

```
{"result": "OK"}
```

### 3.4.8. バイナリイメージのダウンロード

|           |   |                                                                                                                                |
|-----------|---|--------------------------------------------------------------------------------------------------------------------------------|
| URL       | : | download                                                                                                                       |
| メソッド      | : | GET : ○ POST : ×                                                                                                               |
| GET 時の引数  | : | controller=<コントローラ名>&item=<アイテム名>[&filename=<ダウンロードファイル名>]<br>ダウンロードファイル名は省略可能です。                                              |
| POST 時の引数 | : | -                                                                                                                              |
| 戻り値       | : | アイテムに格納されているバイナリイメージ。                                                                                                          |
| 備考        | : | download コマンドはバイナリイメージをダウンロードするため、結果を json で戻すことができません。<br>そのため、download コマンドの結果は HTTP のステータスコードで戻します。<br>ステータスコードの意味は以下の通りです。 |

取得成功:200(Success)

アイテムが存在しないか、アイテムの値がバイナリイメージ以外の場合: 400(Bad Request)

内部処理でエラーが発生した場合:500(Internal Server Error)

## 4. ライセンス表示

### 4.1. Json.NET

Copyright (c) 2007 James Newton-King

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.