

イットbuilder

API 利用ガイドブック

第 1.2 版 2022 年 4 月 20 日改訂

株式会社
中電シーティーアイ

■変更履歴

版数	変更日	変更内容
1.0	2017/8/1	初版発行
1.1	2018/9/1	ファイルアップロード API の追加
1.2	2022/4/20	データベースデータ取得 API の取得項目に最終更新日時を追加 データベースデータ更新 API の入出力パラメータを変更

目次

1	はじめに	1
1.1	本書の位置付	1
2	API 一覧	2
3	API の概要	3
3.1	アクセス方式	3
3.2	リクエスト	3
3.3	エスケープ文字／エスケープでの文字の表現方法	3
3.4	レスポンス	3
3.5	認証	4
4	データ入出力 API	5
4.1	データベースデータ登録 API	5
4.2	データベースデータ取得 API	7
	(補足) 設定可能な検索方法	11
4.3	データベースデータ更新 API	12
4.4	データベースデータ削除 API	15
4.5	ファイルアップロード	16

1 はじめに

1.1 本書の位置付

想定読者：アプリ設定者

本書は、イット builder が提供する API の利用方法について示す。

2 API 一覧

イト builder が提供する API の一覧は下表の通り。

No	カテゴリ	API 名称	説明
1	データ入出力 API	データベースデータ登録	データベースにデータを登録する。
2		データベースデータ取得	データベースのデータを取得する。
3		データベースデータ更新	データベースのデータを更新する。
4		データベースデータ削除	データベースのデータを削除する。
5		ファイルアップロード	ファイルをアップロードし、ファイル登録・更新用のファイル ID を生成する。

3 API の概要

イト builder が提供する API に関する共通的な仕様を定義する。

個別に指定がない場合、以下の仕様となる。

3.1 アクセス方式

Web リクエスト

3.2 リクエスト

Accept: application/json

HTTP メソッド : POST

3.3 エスケープ文字／エスケープでの文字の表現方法

エスケープ表記	元の文字	説明
¥"	"	ダブルクォーテーション
¥¥	¥	バックスラッシュ
¥/	/	スラッシュ
¥b		バックスペース
¥f		改ページ
¥n		キャリジリターン（改行）
¥r		ラインフィード
¥t		タブ
¥uXXXX		4 桁の 16 進数で表記された Unicode 文字

3.4 レスポンス

HTTP レスポンスコードおよびレスポンス BODY とする。HTTP レスポンスコードの仕様は下表の通り。

レスポンス BODY の内容は、各 API 実装によるものとする。

No	コード	意味	解説	レスポンス Body
1	200	正常終了	API の処理結果が正常である場合を示す。	あり
2	400	Bad Request	その他のエラーを示す。	あり
3	500	Internal Server Error	予期しないシステムエラーを示す。	無し

3.5 認証

データベースごとに定義されたアクセスキーをリクエストパラメータとして受け取ることで認証を行う。

 **データベース設定**

1 sampledб(test)

データ件数: 0 -- 更新

全般

データ項目

データ

チューニング

関連リンク

変更

基本設定

データベース名	必須	sampledb(test)	
APIによるデータの操作	必須	☒ する ☐ しない	
		アクセスキー: F7EWMsuXwZmnSVuUGGJm6r8npG9rjil	☐ 新しいアクセスキーを再作成する
メモ			

4 データ入出力 API

4.1 データベースデータ登録 API

- ・エンドポイント

https://【お使いのドメイン】/app/api/v1/database/insert

- ・入力パラメータ仕様

No	項目（JSON キー）	項目名	データ型	必須	備考
1	accessKey	アクセスキー	文字列	○	データベースごとに発行されたアクセスキー
2	dataItems	登録項目	配列	○	
3	id	データ項目 ID	文字列	○	
4	value	データ項目値	配列	○	項目種別がファイルの場合、ファイル ID を指定する（ファイル ID は「ファイルアップロード API」参照）

- ・入力パラメータ JSON 形式のイメージ

```
{
  "accessKey": "ABCD12345",
  "dataItems": [
    {
      "id": "D001",
      "value": ["中電 CTI"]
    },
    {
      "id": "D002",
      "value": ["営業部", "開発部"]
    },
    {
      "id": "D003",
      "value": ["aa11bb22cc33"]
    }
  ]
}
```

・出力パラメータ仕様

No	項目（JSON キー）	項目名	データ型	必須	備考
1	status	ステータス	文字列	○	成功：OK 失敗：NG
2	errorMessage	エラーメッセージ	文字列		
3	dataId	データ ID	文字列		
4	serialNo	システム連番	文字列		

・出力パラメータ JSON 形式のイメージ（成功の場合）

```
{
  "status": "OK",
  "errorMessage": "",
  "dataId": "100010",
  "serialNo": "ABC-100010"
}
```

・出力パラメータ JSON 形式のイメージ（失敗の場合）

```
{
  "status": "NG",
  "errorMessage": "登録データにエラーがありました",
  "dataId": "",
  "serialNo": ""
}
```

4.2 データベースデータ取得 API

・エンドポイント

https://【お使いのドメイン】/app/api/v1/database/select

・入力パラメータ仕様

No	項目（JSON キー）	項目名	データ型	必須	備考
1	accessKey	アクセスキー	文字列	○	データベースごとに発行されたアクセスキー
2	selectItems	取得項目	配列	○	データ ID（data_id）、最終更新日時（last_update_dt）、データ項目 ID を指定可能 ファイル項目：ファイル名のみ
3	sortItems	ソート項目	配列		
4	id	データ項目 ID	文字列		
5	type	ソート種別	文字列		昇順：ASC、降順：DESC
6	conditionItems	検索条件項目	配列		
7	id	データ項目 ID	文字列		データ ID（data_id）、データ項目 ID を指定可能
8	value	条件値	配列		
9	type	検索方法	文字列		検索方法は「（補足）設定可能な検索方法」参照
1 0	startIndex	開始位置	数字		初期設定は 1（先頭から）とする
1 1	recordNumber	取得件数	数字		初期設定は 0（無制限）とする

・ 入力パラメータ JSON 形式のイメージ

```
{
  "accessKey": "ABCD12345",
  "selectItems": [
    "data_id",
    "last_update_dt",
    "D001",
    "D002"
  ],
  "sortItems": [
    {
      "id": "D001",
      "type": "ASC"
    },
  ],
  "conditionItems": [
    {
      "id": "D002",
      "value": ["営業部", "開発部"],
      "type": "IN"
    }
  ],
  "startIndex": 1,
  "recordNumber": 10
}
```

・出力パラメータ仕様

No	項目（JSON キー）	項目名	データ型	必須	備考
1	status	ステータス	文字列	○	成功：OK 失敗：NG
2	errorMessage	エラーメッセージ	文字列		
3	rows	行集合	配列		
4	columns	列集合	配列		
5	id	データ項目 ID	文字列		
6	value	データ項目値	配列		

・出力パラメータ JSON 形式のイメージ（成功の場合）

```
{
  "status": "OK",
  "errorMessage": "",
  "rows": [
    {
      "columns": [
        {
          "id": "D001",
          "value": ["中電 CTI"]
        },
        {
          "id": "D002",
          "value": ["営業部", "開発部"]
        }
      ]
    },
    {
      "columns": [
        {
          "id": "D001",
          "value": ["中電 CTI"]
        },
        {
          "id": "D002",
          "value": ["営業部", "開発部"]
        }
      ]
    }
  ]
}
```

- ・ 出力パラメータ JSON 形式のイメージ（失敗の場合）

```
{  
  "status": "NG",  
  "errorMessage": "検索条件に誤りがありました",  
  "rows": []  
}
```

(補足) 設定可能な検索方法

検索方法	検索条件項目種別								
	単一行 テキスト	複数行 テキスト	数値	日付	択一選択	複数選択	ファイル	自動連番	データ ID
"EQ" : =	○	○	○	○	×	×	○	○	○
"NEQ" : <>	○	○	○	○	×	×	○	○	×
"LK" : LIKE (部分一致)	○	○	×	×	×	×	○	○	×
"LKF" : LIKE (前方一致)	○	○	×	×	×	×	○	○	×
"LKB" : LIKE (後方一致)	○	○	×	×	×	×	○	○	×
"GT" : >	×	×	○	○	×	×	×	×	×
"LT" : <	×	×	○	○	×	×	×	×	×
"GE" : >=	×	×	○	○	×	×	×	×	×
"LE" : <=	×	×	○	○	×	×	×	×	×
"IN" : IN	○	○	○	○	○	○	○	○	○
"NIN" : NOT IN	○	○	○	○	○	○	○	○	×
"NU" : IS NULL	○	○	○	○	○	○	○	×	×
"NNU" : IS NOT NULL	○	○	○	○	○	○	○	×	×

4.3 データベースデータ更新 API

・エンドポイント

https://【お使いのドメイン】/app/api/v1/database/update

・入力パラメータ仕様

No	項目（JSON キー）	項目名	データ型	必須	備考
1	accessKey	アクセスキー	文字列	○	データベースごとに発行されたアクセスキー
2	dataId	データ ID	数字	○	
3	lastUpdateDt	最終更新日時	文字列	○	排他制御に利用する。 指定なし（要素はあるが値なし含む）：必ず更新する 指定あり：DB データの最終更新日時と一致しない場合、エラーを返却して更新しない
4	dataItems	更新項目	配列	○	
5	id	データ項目 ID	文字列	○	
6	value	データ項目値	配列	○	項目種別がファイルの場合、 ファイル ID を指定する（ファイル ID は「ファイルアップロード API」参照）

・ 入力パラメータ JSON 形式のイメージ

```
{
  "accessKey": "ABCD12345",
  "dataId": "100001",
  "lastUpdateDt": "2021/01/04 19:53:29",
  "dataItems": [
    {
      "id": "D001",
      "value": ["中電 CTI"]
    },
    {
      "id": "D002",
      "value": ["営業部", "開発部"],
    },
    {
      "id": "D003",
      "value": ["aa11bb22cc33"]
    }
  ]
}
```

・出力パラメータ仕様

No	項目（JSON キー）	項目名	データ型	必須	備考
1	status	ステータス	文字列	○	成功：OK 失敗：NG
2	errorMessage	エラーメッセージ	文字列		
3	dataId	データ ID	文字列		
4	lastUpdateDt	最終更新日時	文字列		

・出力パラメータ JSON 形式のイメージ（成功の場合）

```
{
  "status": "OK",
  "errorMessage": "",
  "dataId": "1",
  "lastUpdateDt": "2021/01/05 14:13:22"
}
```

・出力パラメータ JSON 形式のイメージ（失敗の場合）

```
{
  "status": "NG",
  "errorMessage": "更新データにエラーがありました",
  "dataId": "100001"
}
```

```
{
  "status": "NG",
  "errorMessage": "他のユーザーにより、更新されております。恐れ入りますが、初めから操作をやり直してください。",
  "dataId": "100001"
}
```

4.4 データベースデータ削除 API

- ・エンドポイント

https://【お使いのドメイン】/app/api/v1/database/delete

- ・入力パラメータ仕様

No	項目（JSON キー）	項目名	データ型	必須	備考
1	accessKey	アクセスキー	文字列	○	データベースごとに発行されたアクセスキー
2	dataId	データ ID	数字	○	

- ・入力パラメータ JSON 形式のイメージ

```
{
  "accessKey": "ABCD12345",
  "dataId": "100001"
}
```

- ・出力パラメータ仕様

No	項目（JSON キー）	項目名	データ型	必須	備考
1	status	ステータス	文字列	○	成功：OK 失敗：NG
2	errorMessage	エラーメッセージ	文字列		

- ・出力パラメータ JSON 形式のイメージ（成功の場合）

```
{
  "accessKey": "ABCD12345",
  "dataId": "100001"
}
```

- ・出力パラメータ JSON 形式のイメージ（失敗の場合）

```
{
  "status": "NG",
  "errorMessage": "削除条件にエラーがありました"
}
```

4.5 ファイルアップロード

・エンドポイント

https:// 【お使いのドメイン】 /app/api/v1/database/file/upload

・リクエスト仕様

Content-Type : multipart/form-data

No	送信するデータ	属性名	属性値	必須	備考
1	アクセスキー	name	accessKey	○	
2	対象ファイル	name	file	○	
		filename	ファイル名	○	

・リクエストボディのイメージ

```
-----WebKitFormBoundaryO5quBRiT4G7Vm3R7
Content-Disposition: form-data; name="accessKey"

ABCD123131XXXAEFAAEFAFA
-----WebKitFormBoundaryO5quBRiT4G7Vm3R7
Content-Disposition: form-data; name="file"; filename="sample.txt"
Content-Type: text/plain

txtInput
-----WebKitFormBoundaryO5quBRiT4G7Vm3R7--
```

・ JavaScript による API 呼び出しサンプル

```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.0/jquery.min.js"></script>
<script>
function testFormFileUpload() {
    // FormData オブジェクトを宣言する
    var formData = new FormData();

    // アクセスキーをフォームに追加する
    formData.append("accessKey", $("#accessKey").val());

    // ファイルをフォームに追加する
    var userfile = $("#userFile").prop('files')[0];
    formData.append('file', userfile, userfile.name);

    $.ajax({
        url: $("#uploadUrl").val(),
        type: "POST",
        data: formData,
        processData: false,
        contentType: false

        // 成功した時実行される
    }).done(function(result, textStatus, jqXHR) {
        $("#span1").text(jqXHR.status);
        $("#span2").text(textStatus);
        $("#span3").text(JSON.stringify(result));

        // 失敗した時に実行される
    }).fail(function(jqXHR, textStatus, errorThrown ) {
        $("#span1").text(jqXHR.status);
        $("#span2").text(textStatus);
        $("#span3").text(errorThrown);

        // 成功/失敗に関わらず実行される
    }).always(function(){
        alert("complete");
    });
}
</script>
```

次項へ

```
</heda>
<body>

<h1>イト Builder ファイルアップロード API の呼出サンプル</h1>
<form name="fileForm" method="post" enctype="multipart/form-data">
<br/>アップロード先 URL :
<br/><input type="text" name="uploadUrl" id="uploadUrl" size="100" value="https://【お使いのドメイン】/app/api/v1/database/file/upload"/>
<br/>
<br/>アクセスキー :
<br/><input type="text" name="accessKey" id="accessKey" size="100"
value="ABCD123131XXAEFAAEFAFA"/>
<br/>
<br/>アップロードファイル :
<br/><input type="file" id="userFile"/>
<br/>
</form>
<hr>
<br/><input type="button" value="アップロード実行" onclick="testFormFileUpload();" />
<br/>
<br/>【レスポンスステータス】 : <span id="span1"></span>&nbsp;<span id="span2"></span>
<br/>【API 返却結果】 : <span id="span3"></span>
<br/>

</body>
</html>
```

・出力パラメータ仕様

No	項目（JSON キー）	項目名	データ型	必須	備考
1	status	ステータス	文字列	○	成功：OK 失敗：NG
2	fileId	ファイル ID	文字列		
3	errorMessage	エラーメッセージ	文字列		

・出力パラメータ JSON 形式のイメージ（成功の場合）

```
{
  "status": "OK",
  "fileId": "aa11bb22cc33",
  "errorMessage": ""
}
```

・出力パラメータ JSON 形式のイメージ（失敗の場合）

```
{
  "status": "NG",
  "fileId": "",
  "errorMessage": "ファイルのサイズが上限を超えました"
}
```