GMA024F0

GridDB
Web API 説明書

東芝デジタルソリューションズ株式会社

は じ め に

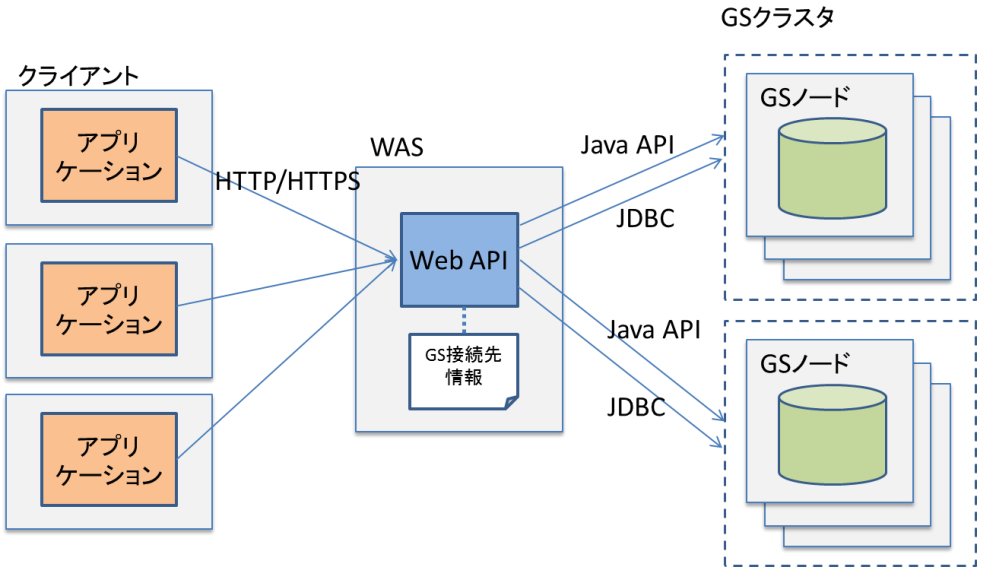
本書では、GridDB における Web API の機能・構築方法および、注意事項について記載しています。
GridDB Starndard Edition / Advanced Edition / Vector Edition をご使用になる前に、必ずお読みください。

商標

目次

1. GridDB Web API とは

GridDB では、GridDB のクラスタに対して、ロウの登録や取得、SQL SELECT 文実行の操作を行うことができる Web API を提供します。Web API は Web アプリケーションとして構成されます。



[メモ]

・クラスタに対して、データ登録や検索などの操作が行えます

・Web API は、Apache Tomcat で動作する Web アプリケーションです

・複数のクラスタをひとつの Web API(Web アプリケーション)から操作することができます

2. 機能

本章では、Web API で使用できる機能について説明します。

機能	説明
ロウ登録	コンテナに対してロウを登録します
ロウ取得	コンテナからロウを取得します
SQL SELECT 文実行	指定したデータベースで SQL SELECT 文を実行します

2.1 ロウ登録

コンテナにロウを登録します。

登録するロウは、JSON 形式で指定します。ひとつのコンテナに複数のロウを登録することもできます。

[メモ]

・ひとつのコンテナに１ロウ/複数ロウのデータを指定して登録することができます。

・登録先のコンテナは存在している必要があります。

・ロウキーが指定されていないコンテナの場合、ロウが登録されます。

・ロウキーが指定されているコンテナの場合、既にコンテナに存在するロウキーと同一のロウキーを指定するとロウが更新されます。ロウキーが異なる場合は、登録されます。

・ロウ登録処理の途中で例外が発生した場合、一部のロウに対する登録のみが反映されたままとなる場合があります。そのため、例外発生時に HTTP クライアントでリトライする場合、ロウキーが指定されていないコンテナの場合は同じデータが重複して登録される場合があります。

■コマンドパス

/:cluster/dbs/:database/containers/:container/rows

項目	説明
:cluster	クラスタ名
:database	データベース名 (public データベースの場合は "public"を指定してください)
:container	コンテナ・テーブル名

■HTTP メソッド

POST

■リクエスト

□リクエストヘッダ

名前	説明	必須
Content-Type	"application/json"を指定します。	○
X-GridDB-User	GridDB へアクセスするユーザを指定します	○
X-GridDB-Password	GridDB へ接続するユーザのパスワードを指定します	○

□リクエストボディ

ロウを下記の JSON 形式で指定してください。

項目	説明	必須
rows	ロウ(カラム値の配列)の配列	○

例）3 つのロウを指定する場合

```
{
    "rows" : [
        ["2016-01-16T10:25:00.253Z", 100.5, "success", true ],
        ["2016-01-16T10:35:00.691Z", 173.9, "error", false ],
        ["2016-01-16T10:45:00.691Z", 328.2, null, false ],
    ]
}
```

・ロウのカラム値の記述

カラムのデータ型に応じて、下記の JSON データ型で記述してください。

カラムのデータ型			JSON データ型	例
基本型	ブール型	BOOL	真偽値（ true | false ） または 文字列（ "true" | "false"）	true
	文字列型	STRING	文字列	"GridDB"
	整数型	BYTE/SHORT/INTEGER/LONG	数値 または 文字列	512
	浮動小数点数型	FLOAT/DOUBLE	数値 または 文字列	593.5
	時刻型	TIMESTAMP	文字列 ・UTC ・フォーマット YYYY-MM-DDThh:mm:ss.SSSZ	"2016-01-16T10:25:00.253Z"
	空間型	GEOMETRY	文字列（WKT 表現）	"POLYGON((0 0,10 0,10 10,0 10,0 0))"
配列型	ブール型	BOOL	真偽値の配列 または 文字列の配列	[true, false, true]
	文字列型	STRING	文字列の配列	["A","B","C"]
	整数型	BYTE/SHORT/INTEGER/LONG	数値の配列 または 文字列の配列	[100, 39, 535]
	浮動小数点数型	FLOAT/DOUBLE	数値の配列 または 文字列の配列	[3.52, 6.94, 1.83]
	時刻型	TIMESTAMP	文字列の配列 （書式は単数型の時刻型と同じ）	["2016-01-16T10:25:00.253Z", "2016-01-17T01:42:53.038Z"]

[メモ]

・BLOB 型はサポートしていません。BLOB 型のカラムを持つコンテナを指定するとエラーになります。

・カラムのデータに NULL 値（JSON データ型の null）を指定した場合、以下のように動作します。

・対応するカラムに NOT NULL 制約が設定されている場合　：登録エラー
・対応するカラムに NOT NULL 制約が設定されていない場合：NULL 値が登録される

■レスポンス

□ステータスコード

「2.4.4」をご参照ください。

□レスポンスボディ

処理に成功した場合は、下記の JSON データが返ります。

項目	説明
count	処理を行ったロウの数

失敗した場合のボディは、「2.4.5」をご参照ください。

2.2 ロウ取得

コンテナやテーブルのロウを取得します。条件を指定して、取得するロウを絞込むこともできます。

■コマンドパス

/:cluster/dbs/:database/containers/:container/rows

項目	説明
:cluster	クラスタ名
:database	データベース名 (public データベースの場合は "public"を指定してください)
:container	コンテナ・テーブル名

■HTTP メソッド

GET

■リクエスト

□リクエストヘッダ

名前	説明	必須
Content-Type	リクエストボディで値(JSON 形式)を送信する場合は、"application/json"を指定します。	ボディを指定する場合は必須
X-GridDB-User	GridDB へアクセスするユーザを指定します	○
X-GridDB-Password	GridDB へ接続するユーザのパスワードを指定します	○

□リクエストボディ

取得するロウの数や条件を指定することができます。

指定をすべて省略した場合は、コンテナのすべてのロウデータが取得できます。

項目	説明	JSON データ型	必須
offset	取得開始位置	数値(0 からの整数)	-
limit	取得数	数値(0 からの整数)	-
condition	条件	文字列	-
sort	ソート	文字列	-

[メモ]

・limit で 0 を指定した場合は、条件に合致するすべてのロウが返ります。

・offset は、必ず limit と同時に使用する必要があります。

例）カラム id の値が 50 以上のロウデータを、id の値で降順ソートし、11 番目から 100 個取得する

```
{
    "offset" : 10,
    "limit"  : 100,
    "condition" : "id >= 50",
    "sort" : "id desc"
}
```

■レスポンス

□レスポンスボディ

取得したロウは、下記の形式の JSON データで返ります。

項目	説明	JSON データ型
columns	カラム情報の配列	配列
name	カラム名	文字列
type	データ型	文字列
rows	ロウの配列	配列

例）

```
{
    "columns" : [
        {"name": "date", "type": "TIMESTAMP" },
        {"name": "value", "type": "DOUBLE" },
        {"name": "str", "type": "STRING" }
    ],
    "rows" : [
        ["2016-01-16T10:25:00.253Z", 100.5, "normal" ],
        ["2016-01-16T10:35:00.691Z", 173.9, "normal" ],
        ["2016-01-16T10:45:00.032Z", 173.9, null ]
    ]
}
```

・カラムのデータ型

カラムのデータ型に応じて、下記の JSON データ型で出力されます。

データ型			JSON データ型	例
基本型	ブール型	BOOL	真偽値 （ true | false ）	true
	文字列型	STRING	文字列	"GridDB"
	整数型	BYTE/SHORT/INTEGER/LONG	数値	512
	浮動小数点数型	FLOAT/DOUBLE	数値	593.5
	時刻型	TIMESTAMP	文字列 ・UTC ・フォーマット YYYY-MM-DDThh:mm:ss.SSSZ	"2016-01-16T10:25:00.253Z"
	空間型	GEOMETRY	文字列 (WKT 表現)	"POLYGON((0 0,10 0,10 10,0 10,0 0))"
配列型	ブール型	BOOL	真偽値の配列	[true, false, true]
	文字列型	STRING	文字列の配列	["A","B","C"]
	整数型	BYTE/SHORT/INTEGER/LONG	数値の配列	[100, 39, 535]
	浮動小数点数型	FLOAT/DOUBLE	数値の配列	[3.52, 6.94, 1.83]
	時刻型	TIMESTAMP	文字列の配列 （書式は単数型の時刻型と同じ）	["2016-01-16T10:25:00.253Z", "2016-01-17T01:42:53.038Z"]

[メモ]

・BLOB 型は未サポートです。BLOB 型のカラムを持つコンテナを指定した場合、BLOB のカラムは空文字が返ります。

・カラムのデータが NULL 値の場合は、JSON データ型の null が返ります。

・パーティションされたコンテナは、ロウ取得のサポート対象外です。

2.3 SQL SELECT 文実行

指定したデータベースで SQL SELECT 文を実行します。
本機能は、GridDB Advanced Edition/Vector Edition でのみ使用できます。

■コマンドパス

/:cluster/dbs/:database/sql

項目	説明
:cluster	クラスタ名
:database	データベース名 (public データベースの場合は "public"を指定してください)

■HTTP メソッド

GET、あるいは、POST

■リクエスト

□リクエストヘッダ

名前	説明	必須
Content-Type	"application/json"を指定します。	○
X-GridDB-User	GridDB へアクセスするユーザを指定します	○
X-GridDB-Password	GridDB へ接続するユーザのパスワードを指定します	○

■リクエストボディ

SQL SELECT 文を下記の JSON 形式で指定してください。

項目	説明	JSON データ型	必須
type	クエリ文の種別 ("sql-select"だけを指定できます)	文字列	省略可
stmt	SQL SELECT 文	文字列	必須

例）

```
{
    "type" : "sql-select",
    "stmt" : "select * from emp"
}
```

■レスポンス

□レスポンスボディ

取得したロウは、下記の形式の JSON データで返ります。

項目	説明	JSON データ型
columns	カラム情報の配列	配列
name	カラム名	文字列
type	データ型	文字列
rows	ロウの配列	配列

例）

```
{
    "columns" : [
        {"name": "date", "type": "TIMESTAMP" },
        {"name": "value", "type": "DOUBLE" },
        {"name": "str", "type": "STRING" }
    },
    "rows" : [
        ["2016-01-16T10:25:00.253Z", 100.5, "normal" ],
        ["2016-01-16T10:35:00.691Z", 173.9, "normal" ],
        ["2016-01-16T10:45:00.032Z", 173.9, null ]
    ]
}
```

・カラムのデータ型

カラムのデータ型に応じて、下記の JSON データ型で出力されます。

データ型	型名	JSON データ型	例
ブール型	BOOL	真偽値 （ true | false ）	true
文字列型	STRING	文字列	"GridDB"
整数型	BYTE/SHORT/INTEGER/LONG	数値	512
浮動小数点数型	FLOAT/DOUBLE	数値	593.5
時刻型	TIMESTAMP	文字列 ・UTC ・フォーマット YYYY-MM-DDThh:mm:ss.SSSZ	"2016-01-16T10:25:00.253Z"

[メモ]

・カラムのデータが NULL 値の場合は、JSON データ型の null が返ります。

・BLOB 型、GEOMETRY 型、配列型は未サポートです。

・BLOB 型のカラムを持つコンテナを指定した場合、データ型は"BLOB"となりますが、カラムのデータとしては空文字が返ります。

・GEOMETRY 型と配列型カラムのデータ型は"UNKNOWN"となり、カラムのデータとしては null が返ります。

2.4 共通機能（HTTP リクエスト/レスポンス）

各機能において共通の HTTP リクエスト部分について説明します。

2.4.1 URI

Web API を利用する場合にアクセスする URI です。

http://WAS ドメイン/griddb/v1/(コマンドパス)

[メモ]
- WAS ドメインは、Web API を構築した Web アプリケーションの環境に応じて指定してください。
- (コマンドパス)は、各機能の節をご参照ください。
- 名前に記号('-' '.' '/' '=')を含むクラスタ・データベース・コンテナに対して、Web API で操作を行うことはできません。

2.4.2 リクエストヘッダ

ヘッダには、下記の値を指定してください。

名前	説明	必須
Content-Type	リクエストボディで値(JSON 形式)を送信する場合は、 "application/json"を指定します。	リクエストボディが無い場合は省略
X-GridDB-User	GridDB へアクセスするユーザを指定します	○
X-GridDB-Password	GridDB へ接続するユーザのパスワードを指定します	○

2.4.3 リクエストボディ

リクエストボディで値を送信する場合、JSON 形式で記述してください。JSON 形式のフォーマットは、各機能の節をご参照ください。

- JSON 形式のデータの文字コードは UTF-8 で記述してください。

・日時を記述する場合は、UTC で下記の形式で記述してください。

YYYY-MM-DDThh:mm:ss.SSSZ

・日付型の値で上記以外の形式を指定した場合は、エラーになります。

例）

2016/01/17T14:32:33.888Z　・・・年月日の区切り文字誤りでエラー

2016-01-18　・・・時間の指定が無いのでエラー

2.4.4 レスポンスコード

以下のレスポンスコードが返ります。

名前	説明
200	成功
400	リクエストデータの誤り
403	指定されたリソースが存在しない
500	Web API/GridDB でエラーが発生

2.4.5 レスポンスボディ

処理が成功した場合のレスポンスボディは、各機能の節をご参照ください。
処理が失敗した場合は、レスポンスボディには下記の形式でエラーメッセージが返ります。

```
{
    "version":"v1",
    "errorCode":"エラーコード",
    "errorMessage":"エラーメッセージ"
}
```

3. 構築手順

3.1 インストール

(1) クライアントパッケージのインストール

クライアントパッケージをインストールします。

```
rpm -Uvh griddb-xx-client-X.X.X-linux.x86_64.rpm
```

インストール後、Web API の WAR ファイルや設定ファイルは下記のように配置されます。

```
/usr/gridstore/webapi/griddb.war                              ・・Web API war ファイル

/var/lib/gridstore/webapi/conf/griddb_webapi.properties       ・・設定ファイル
                           /repository.json                   ・・クラスタ情報定義ファイル
                           /log                               ・・ログ出力ディレクトリ
```

(2) Web アプリケーションサーバへデプロイ

Web API の WAR ファイルを、Web アプリケーションサーバに配置して、デプロイします。

```
Web API war ファイル
        /usr/gridstore/webapi/griddb.war
```

3.2 環境設定

(1) 接続先のクラスタを設定　（必須）

Web API から接続するクラスタの情報を、クラスタ情報定義ファイルに設定します。

```
クラスタ情報定義ファイル
    /var/lib/gridstore/webapi/conf/repository.json
```

接続するクラスタのクラスタ定義ファイル(gs_cluster.json)の値を基に、mode にクラスタ構成の接続方式を指定し、方式に対応するアドレス情報の項目を記載してください。

・記載する項目

項目		説明	必須
clusters		クラスタ情報を記述	○
	name	クラスタ名	○
	mode	接続方式（MULTICAST ｜ FIXED_LIST ｜ PROVIDER ）	○
	address	(MULTICAST)ロウ登録／取得用マルチキャストアドレス	MULTICAST の場合必須
	port	(MULTICAST) ロウ登録／取得用ポート番号	MULTICAST の場合必須
	transactionMember	(FIXED_LIST) 固定リスト方式の場合、ロウ登録／取得用アドレスとポートを記述します。	FIXED_LIST の場合必須
	providerUrl	(PROVIDER) プロバイダ方式の場合、全機能共通のプロバイダ URL を記述します。	PROVIDER の場合必須
	jdbcAddress	(MULTICAST) SQL SELECT 文用マルチキャストアドレス (AE/VE の場合に設定してください)	MULTICAST の場合必須
	jdbcPort	(MULTICAST) SQL SELECT 文用ポート番号 (AE/VE の場合に設定してください)	MULTICAST の場合必須
	sqlMember	(FIXED_LIST) 固定リスト方式の場合、SQL SELECT 文用アドレスとポートを記述します。 (AE/VE の場合に設定してください)	FIXED_LIST の場合必須

例） マルチキャスト方式の場合

```
{
  "clusters" : [
    {
      "name" : "defaultCluster",
      "mode" : "MULTICAST",
      "address" : "239.100.100.111",
      "port" : 31999
    }
  ]
}
```

[メモ]
GridDB Standard Edition で SQL SELECT 文実行機能を使用した場合、クラスタ情報定義ファイル repository.json に SQL 用の接続先が記載されていると、接続エラー[145028:JC_BAD_CONNECTION]になります。レスポンスコードとして 500 が、また、レスポンスボディとして下記のようなメッセージが返ります。
例）

```
{
    "version":"v1",
    "errorCode":"E2060B",
    "errorMessage":"[145028:JC_BAD_CONNECTION] Connection problem occurred (retryCount=x, elapsedMillis=xxxx, failureMillis=xxxx, queryTimeoutMillis=(not bounded), loginTimeoutMillis=xxxx, operation=LOGIN, reason=[145028:JC_BAD_CONNECTION] Failed to connect"
}
```

(2) 動作環境を設定　（任意）
Web API の動作を設定します。

```
設定ファイル
    /var/lib/gridstore/webapi/conf/griddb_webapi.properties
```

この設定の変更を行わずに、すべてデフォルト値のままでも Web API は動作します。システムの必要に応じて、値の変更を行ってください。

名前	説明	デフォルト値
GridDB に関する設定		
failoverTimeout	WebAPI から GridDB へのアクセスで、ノード障害を検知してからリトライを繰り返すフェイルオーバー時間(秒)を指定します。	5
transactionTimeout	トランザクション開始から終了までの最大時間(秒)を指定します。	30
containerCacheSize	コンテナ情報をキャッシュする数を指定します。	100

Web API の動作設定		
maxGetRowSize	ロウ取得、あるいは、SQL SELECT 文実行結果の上限のサイズ(MB) (1～2048 までの整数)	20
maxPutRowSize	ロウ登録の上限のサイズ(MB) (1～2048 までの整数)	20
loginTimeout	SQL SELECT 文実行時の接続タイムアウト(秒) (値は整数を指定します。0 以下のときは SQL SELECT 文実行を禁止します。)	5

[メモ]

・環境設定を反映させるには Tomcat を再起動する必要があります。

(3) ログ出力先を設定　(任意)

Web API のログは、デフォルトでは下記のディレクトリに出力されます。

```
ログ出力先ディレクトリ
    /var/lib/gridstore/webapi/log
```

出力先を変更する場合は、下記のファイルを変更してください。

```
ログ出力先ディレクトリ
    [WAS インストールディレクトリ]/webapps/griddb/v1/WEB-INF/classes/logback.xml
```

3.3 起動と停止

Web API の起動、停止方法は、Web API をインストールした Web アプリケーションサーバを起動、または停止してください。

3.4 動作確認

Web API の動作確認は、Linux の curl コマンド等で行ってください。

例） ロウ取得

```
curl -H "X-GridDB-User:user" -H "X-GridDB-Password:password" http://host:port/griddb
/v1/cluster/dbs/public/containers/test/rows
```

例） ロウ登録

```
curl -X POST -H "X-GridDB-User:user" -H "X-GridDB-Password:password" -H "Content-type:
application/json" -d '{"rows":[["value", 1]]}' http://host:port/griddb/v1/cluster/
dbs/public/containers/test/rows
```

例） SQL SELECT 文実行

```
curl -X POST -H "X-GridDB-User:user" -H "X-GridDB-Password:password" -H "Content-type:
application/json" -d '{"stmt":"select * from test"}' http://host:port/griddb/v1/cluster/
dbs/public/sql
```

3.5 アンインストール

Web アプリケーションサーバを停止して、[WAS インストールディレクトリ]/webapps/gridstore/v1 ディレクトリおよび配置した war ファイルを削除してください。

東芝デジタルソリューションズ株式会社