

Connector 設定ガイド

PlanetCross Connector

PlanetCross Connector では、サービスを SQL から簡単に作成できます。MySQL/MariaDB、PostgreSQL、SQLite、Oracle、SQL Server および IBM DB2 に対応しています。

目次

1. はじめに.....	2
1.1 動作の仕組み.....	2
2. 設定の概要.....	3
3. Connector の設定	4
3.1 設定変更の適用.....	5
4. サービスの設定.....	6
4.1 サービスの追加.....	6
4.1.1 名前	6
4.1.2 接続文字列	7
4.1.3 シンプルモードとアドバンスモード	9
4.1.4 保存	13
4.2 サービスの更新.....	14
4.3 サービスの削除.....	14
4.4 許諾のサポート.....	15
付録 A 注意事項	16
A.1 MySQL にて指定する変数名はカラム名と同一値に設定できません.....	16
A.2 PostgreSQL にて指定する変数名の注意事項	16
改訂履歴	17

商標について

- ・ Microsoft、SQL Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標です。
- ・ MS は Microsoft の略称です。
- ・ Amazon Web Services、“Powered by Amazon Web Services”ロゴ、[およびかかる資料で使用されるその他の AWS 商標] は、米国その他の諸国における、[Amazon.com](https://www.amazon.com), Inc. またはその関連会社の商標です。
- ・ IBM、DB2、IBM Cloud は、世界の多くの国で登録された International Business Machines Corporation の商標です。
- ・ PostgreSQL は、PostgreSQL の米国およびその他の国における商標または登録商標です。
- ・ Planetway、PlanetCross は Planetway Japan 株式会社の商標または登録商標です。
- ・ その他記載の会社名、製品名は、それぞれの会社の商標または登録商標です。

1. はじめに

1.1 動作の仕組み

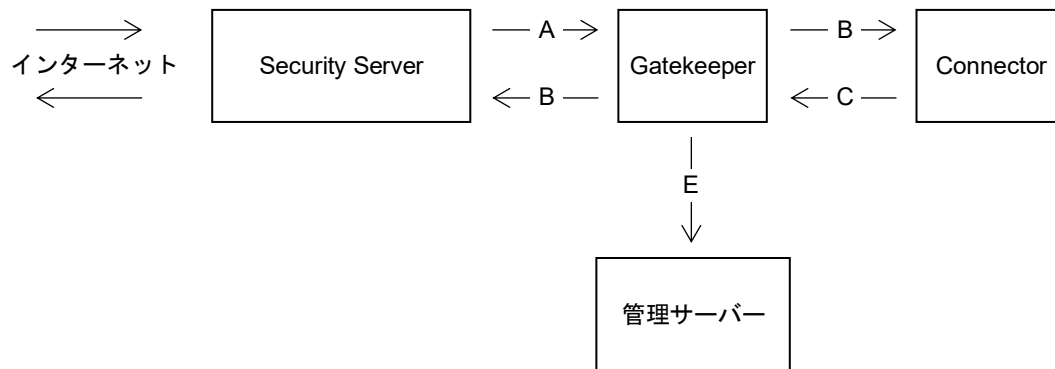


図 1 Connector を使用したネットワーク

Connector では、SOAP リクエストを処理する SOAP サーバーと、サービス設定用の UI を提供する管理サーバーの 2 台の Web サーバーが実行されます。

Security Server から Connector に SOAP 要求が送信されると(A)、Connector はデータベースに SQL クエリーを送信して(B)、データベースよりレスポンスを受信し(C)、SOAP レスポンスを生成して Security Server に返します(D)。

管理サーバーは、サービス管理者がサービスを設定するための Web UI を提供します(E)。サービスが設定されると、Connector はサービス設定をファイルに保存し、WSDL URL を生成します。この URL は、Security Server がサービスを公開するためにフェッチ(A)することができます。

2. 設定の概要

Connector の設定の概要は以下のとおりです。

データベースでの準備

- (1) ConnectorからDBへ接続するため、ログイン可能なログインユーザ・パスワードを準備

Connectorに対しての作業

- (1) データベースへの接続設定を実施(Database Connections)。
- (2) データサービスの設定(データベースに対してどのようなSQLを発行させるか等)を実施(Services)。
- (3) 設定したデータサービスを元に、Security Serverに読み込ませるためのWSDLファイルを設定。

Security Server (データプロバイター)に対しての作業

- (1) Clientとしてサービスを登録(Security Server Clients => ADD CLIENTS)。
- (2) PlanetCross Engineへ登録したCLIENTSをサービスとして登録依頼(Planetwayへ Member Name、Member Class、Member Code、Subsystem Codeを伝達)
- (3) データサービスを登録(Servicesタブ)
- (4) Connectorで生成されたWSDLファイルを読み込み(ADD WSDL)、Connectorで生成されたURLを読み込む
- (5) 読み込まれたWSDLを展開し(+)、各サービスに対してどの組織のSecurity Serverやアプリケーションにサービスを公開するかアクセス権利を登録。(Access Rights)
ADD SUBJECTSボタン入力後、Searchでアクセス権を与える対象を検索、対象を選択後、ADD SELECTED TO ACLでアクセス権を付与。

PlanetCross Engineを通じて、どのようなサービスが企業(組織)で公開しているかを検索、確認できるようになります。

3. Connector の設定

Connector 自体は/opt/planetx/connector/フォルダにインストールされます。

Web サーバーの動作を設定するには、/etc/planetx/connector/main-conf.json を編集します。

デフォルト設定は以下のようになっています。

```
{
  "conf": "/etc/planetx/connector/services.json",

  "services":
  { "port": 8003
    , "host": "0.0.0.0"
    , "schema": "http"
    , "hostname" : null
    , "log": {
      "access"      : "/var/log/planetx/connector/pw-access.log"
      , "error"      : "/var/log/planetx/connector/pw-error.log"
      , "sql"        : "/var/log/planetx/connector/pw-sql.log"
      , "soapAll"    : ""
      , "soapHeader" : "/var/log/planetx/connector/pw-soapheader.log"
    }
  },

  "admin":
  { "port": 8002
    , "host": "127.0.0.1"
  }
}
```

以下は設定ファイルの説明です。

- “conf” : SOAP リクエストを処理するために管理サーバーが書き込み、SOAP サーバーが読み取る json ファイルの完全なパス
- “services” : SOAP サーバーを設定するオブジェクト
- “services” . “port” : SOAP サーバーが listen するポート
- “services” . “schema” : http または https のいずれか
- “services” . “host” : バインドする TCP ホスト(パブリックアクセスの場合は 0.0.0.0)
- “services” . “hostname” : 生成された WSDL URL のホスト名と、WSDL 内に書き込まれた URL。null または未設定の場合は、システム設定が使用されます。
- “services” . “log” : ログの動作を設定するオブジェクト
- “services” . “log” . “access” : SOAP サーバーの HTTP アクセスログ
- “services” . “log” . “error” : エラーログ
- “services” . “log” . “sql” : SQL クエリーログ
- “services” . “log” . “soapAll” : SOAP 要求および応答の全体がこのファイルに記録されます。空の場合、ログが無効になります。

- “services” . “log” . “soapHeader” : 要求および応答の SOAP ヘッダーがこのファイルに記録されます。
- “admin” : 管理サーバーを設定するオブジェクト
- “admin” . “port” : 管理サーバーが listen するポート
- “admin” . “host” : バインドする TCP ホスト

管理サーバーは、デフォルトでは localhost で listen します。

admin.host を、管理サーバーへのアクセスに使用するネットワークインターフェースの IP アドレスに変更します。

3.1 設定変更の適用

Connector の現在のステータスを確認します。

```
$ systemctl status planetx-connector.service
planetx-Connector.service - PlanetX Connector
   Loaded: loaded (/lib/systemd/system/planetx-connector.service; enabled;
   vendor preset: enabled)
   Active: active (running) since Thu 2018-08-02 09:08:20 UTC; 34s ago
   Main PID: 1220 (node)
     Tasks: 10
    Memory: 50.9M
       CPU: 333ms
    CGroup: /system.slice/planetx-connector.service
            └─1220 /opt/planetx/connector/node/bin/node
               /opt/planetx/connector/bin/planetx-Connector.js

Aug 02 09:08:20 ip-xxx-xxx-xxx-xxx systemd[1]: Started PlanetX Connector.
Aug 02 09:08:21 ip-xxx-xxx-xxx-xxx node[1220]: UI service running at ...
Aug 02 09:08:21 ip-xxx-xxx-xxx-xxx node[1220]: SOAP service running at ...
Aug 02 09:08:21 ip-xxx-xxx-xxx-xxx node[1220]: Services are not defined
```

Connector を再起動して設定ファイルの変更を適用します。

4. サービスの設定

サービスを設定するには、管理サーバーの UI を表示します。

管理サーバーの UI は `http://{host}:{admin.port}/` にあります。

ここで、`{host}` はブラウザから管理サーバーにアクセス可能なホスト名または IP アドレス、`{admin.port}` は設定ファイルで指定されているポート番号です。

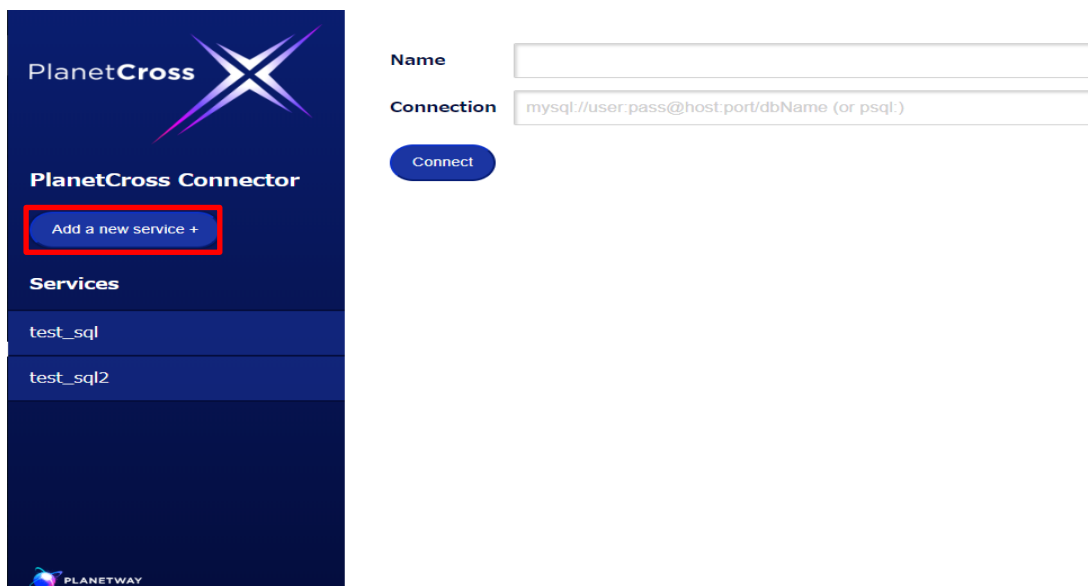
管理サーバーの UI では以下のブラウザがサポートされます。

- 最新バージョンの Safari
- 最新バージョンの Google Chrome

注：Internet Explorer はサポートしていません。

4.1 サービスの追加

新規サービスの設定を開始するには、画面左側にある Add a new service ボタンを押します。



4.1.1 名前

[Name]フィールドは WSDL URL で使用され、以下の 1 バイト(半角)文字のみを使用できます。

- 大文字または小文字のアルファベット
- 数字
- アンダースコア(_)
- ダッシュ記号(-、ただし先頭文字には使用不可)

名前は、1 つの Connector のインストール内で一意である必要があります。

4.1.2 接続文字列

接続文字列は、以下のいずれかでなければなりません。

- [mysql://user:passwd@host:port/databasename](#)
- [psql://user:passwd@host:port/databasename](#)
- [oracle://user:passwd@host:port/databasename](#)
- [db2://user:passwd@host:port/databasename](#)
- [sqlite:///filename](#)
- [mssql://user:passwd@host:port/databasename](#)

ポートがデフォルトに設定されている場合は、ポートを省略できます。

さらに高度な接続パラメーターの場合は、URL 引数を使用します。これらの引数はデータベース接続ドライバーに直接渡されます。各ドライバーのマニュアルは以下のサイトで参照できます。

- mysql: <https://www.npmjs.com/package/mysql>
- postgresql: <https://www.npmjs.com/package/pg>
- oracle: <https://www.npmjs.com/package/oracledb>
- db2: https://www.npmjs.com/package/ibm_db
- sqlite: <https://www.npmjs.com/package/sqlite3>
- SQL Server: <https://www.npmjs.com/package/mssql>

たとえば、プーリングを設定するには、以下の接続文字列を使用します。

- [mysql://user:passwd@host:port/databasename?connectionLimit=10&acquireTimeout=2000](#)
- [psql://user:passwd@host:port/databasename?max=10&idleTimeoutMillis=1000](#)
- [oracle://user:passwd@host:port/databasename?poolMax=10](#)

接続にサブオブジェクトがある場合は、ドットを使用してサブオブジェクトを作成します。

たとえば、TLS を使用して接続する場合がこれに該当します。

AWS 内の mysql に接続するための URL は、

[mysql://root:pw@yourhost.amazonaws.com/db?ssl.ca=/home/me/rds-combined-ca-bundle.pem&connectionLimit=10](#)

のようになります(AWS の詳細については、

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_MySQL.html#MySQL.Concepts.SSLSupport を参照)。

また、サーバーに rds-combined-ca-bundle.pem をダウンロードし、そのファイルの完全なパスを指定する必要があります(ファイルが自動的に読み取られます)。

4.1.2.1 接続文字列のパーセントエンコーディング

接続文字列内の文字は、必要に応じて [RFC3986](#) に従ってパーセントエンコードする必要があります。

RFC3986 で定められている予約文字は、以下のようにパーセントエンコードする必要があります。

Reserved characters	!	#	\$	%	&	'	(	)	*	+	,
/	:	;	=	?	@	[	]				
Percent encoded	%21	%23	%24	%25	%26	%27	%28	%29	%2A	%2B	
	%2C	%2F	%3A	%3B	%3D	%3F	%40	%5B	%5D		

例:

```
mysql://A1b}_.-%23:1Abc%40.-_~%28%29@mysql:3306/databasename
```

上記の接続文字列のユーザー名は A1b}_.-#、パスワードは 1Abc@.-_~() です。

Name と Connection を入力した後[Connect]ボタンを押します。接続に成功すると Config Mode 以下の設定ができるようになります。

4.1.3 シンプルモードとアドバンスモード

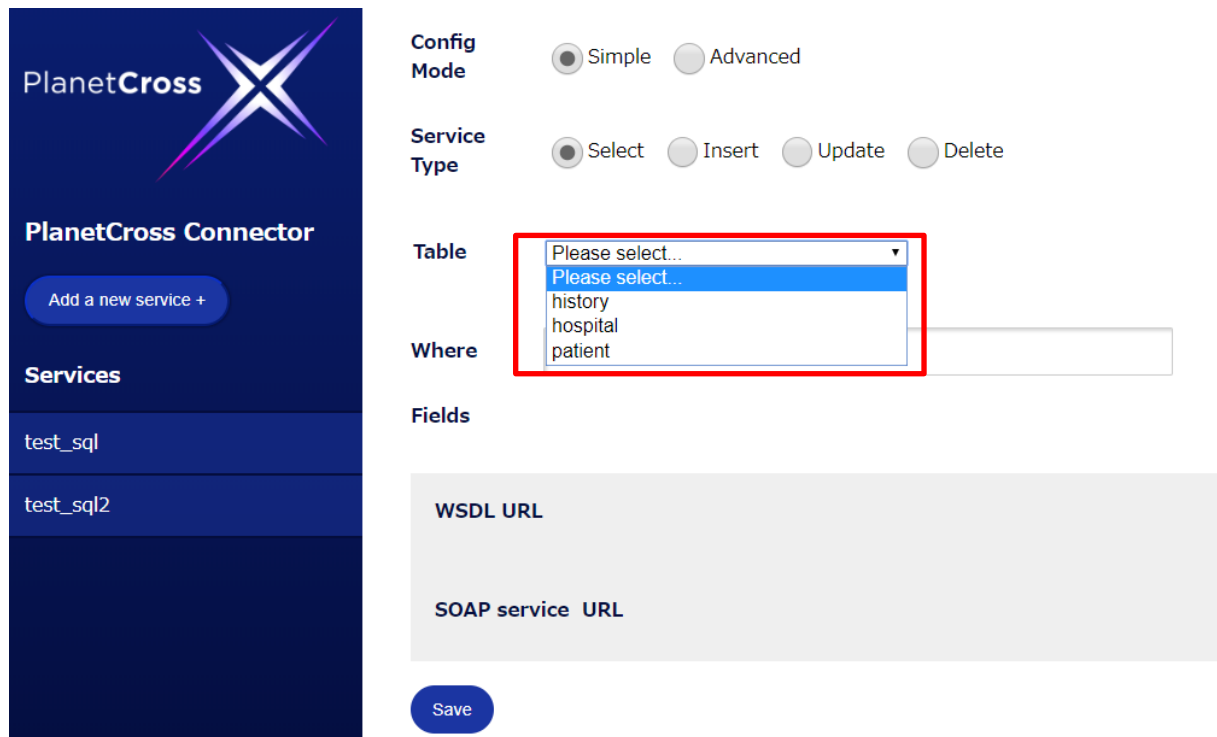
設定モードには、シンプルモードとアドバンスモードの 2 つがあります。それぞれ「Config Mode」で「Simple」か「Advanced」を指定してください。

4.1.3.1 シンプルモード

シンプルモードでは、データベースから読み込まれたテーブルとフィールド（カラム）が表示されます。サービスの応答で返されるフィールドを選択することができます。

「Service Type」で使用したい SQL データ操作を選びます。

「Table」を選択すると、接続したデータベースのテーブルがプルダウン表示されます。選択したテーブルのフィールドが「Fields」に表示されますので任意のフィールドを指定してください(複数選択可)。



The screenshot shows the PlanetCross Connector configuration page. On the left is a dark blue sidebar with the PlanetCross logo and a list of services: 'test_sql' and 'test_sql2'. The main area is white and contains the following sections:

- Config Mode:** Two radio buttons, 'Simple' (selected) and 'Advanced'.
- Service Type:** Four radio buttons, 'Select' (selected), 'Insert', 'Update', and 'Delete'.
- Table:** A dropdown menu with a red border. The dropdown is open, showing 'Please select...' at the top and a list of tables: 'history', 'hospital', and 'patient'.
- Where:** An empty text input field.
- Fields:** A section with two labels, 'WSDL URL' and 'SOAP service URL', each followed by a large, empty text input area.
- Save:** A blue button at the bottom.

「Where」は SQL 文の Where 句を記述します。変数を与えたい場合は「\$」ドル記号を使用します。たとえば「patient_id」フィールド（カラム）を変数名「patient_id1」で取得させる場合は以下の通り入力します。

```
(where) patient_id = $patient_id1
```

シンプルモードの where 句では、以下の演算子がサポートされます。

- AND、OR
- =, >, >=, <, <=

シンプルモードは、MySQL、PostgreSQL、および SQLite でのみサポートされます。その他のデータベースでは、アドバンスモードを使用してください。

4.1.3.2 アドバンスモード

アドバンスモードでは、SQL クエリー全体を記述します。変数を与える場合は、シンプルモード同様に「\$」ドル記号を使います。

LIKE 演算子はアドバンスモードでのみサポートされます。LIKE クエリーでの変数指定を行う場合、concat 関数をご使用ください。concat 関数の構文につきましては、各データベースの仕様をご確認願います。MySQL、postgreSQL における記述例を以下に示します。

MySQL（変数名「name」の前方一致）

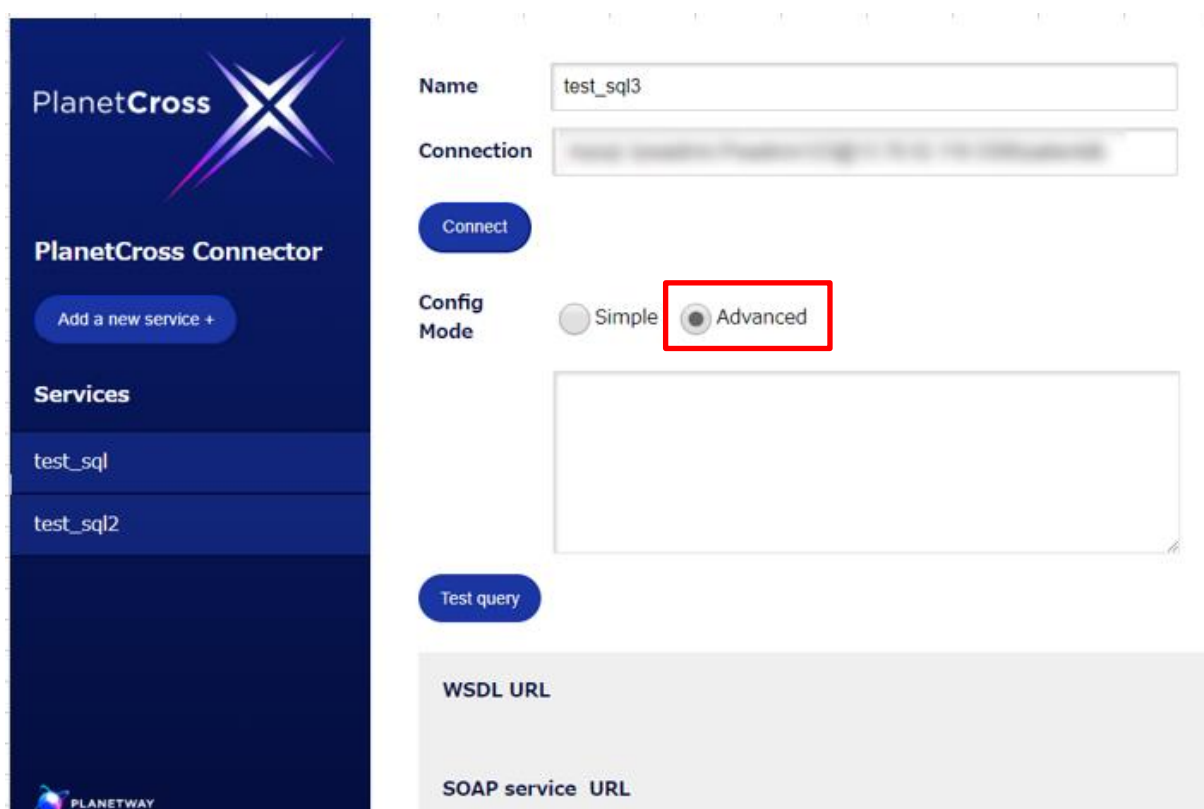
```
SELECT * FROM users WHERE username LIKE CONCAT($name, '%');
```

postgreSQL (変数名「name」の前方一致、ユーザー型が varchar 型の場合)

```
SELECT * FROM users WHERE username LIKE CONCAT($name::varchar, '%');
```

尚、以下記述形式はサポートされませんのでご注意ください。

```
SELECT * FROM users WHERE username LIKE '$name%';
```



4.1.3.3 シンプルモードとアドバンスモードの共通点

前述の通りどちらのモードでも、変数 (=SOAP 入力パラメーター) をマークするために「\$」ドル記号が使用されます。

たとえば、`select * from foo where id = $i` と記述した場合、`i` は自動的に入力パラメーターになります。

SQL では変数名がバインドパラメーターとして使用され、Oracle では予約語を使用できないことに注意してください(予約語の一覧については、Oracle のマニュアル (https://docs.oracle.com/cd/B19306_01/em.102/b40103/app_oracle_reserved_words.htm) を参照)。

`insert`、`update`、および `delete` のクエリーも使用できます。フィールド名は入力パラメーターとして使用され、影響を受ける行数が返されます。

4.1.3.4 ストアドプロシージャの限定サポート

アドバンスモードの Connector では、各出力パラメーターを個別に返すように SELECT クエリーが記述されている場合に、PostgreSQL ストアドプロシージャがサポートされます。

以下のテーブルおよびストアドプロシージャがあるとします。

```
-- Create table
CREATE TABLE person (
    id    integer,
    name  varchar(100),
    sex   char(1)
);

-- Insert row
INSERT INTO person values(1, 'suzuki', 'A');

-- Create stored procedure
CREATE TYPE PersonEntity AS (id integer, name varchar(100), sex char(1));

CREATE OR REPLACE FUNCTION SampleStoredProcedure(argId integer)
RETURNS PersonEntity AS $$
DECLARE
    returnValue RECORD;
BEGIN
    SELECT id, name, sex FROM Person WHERE id = argId
    INTO returnValue;
    RETURN returnValue;
END;
$$ LANGUAGE PLpgSQL;
```

この場合、以下の SELECT SQL はアドバンスモードで動作し、SOAP レスポンスで id、name、および sex パラメーターを個別に返します。

```
SELECT * FROM SampleStoredProcedure($id)
```

【注記】

ストアドプロシージャのサポートは限定的であり、すべてのデータベース、すべての言語、およびストアドプロシージャを作成するためのあらゆる方法を完全にサポートする予定がないことに注意してください。

4.1.3.5 列名

データベースの列名により、SOAP レスポンス本文の XML 要素名が定義されます。

列名には、[XML 1.0 規格](#)の「2.3 Common Syntactic Constructs」で「Name」として定義されている、「:」、「-」、および「.」以外のすべての文字を使用できます。その他記述ルールにつきましては、同資料内の「B Character Classes」を参照願います（例えば、先頭が半角数字で始まるのは不可、一部の漢字（CJK 互換漢字）はサポート範囲外などがあります）。

この条件を満たしていない列名を使用する場合、SELECT column-name-with-dash AS column_name_without_dash FROM table WHERE ...のように、アドバンスモードで AS キーワードを使用することで、列名を変更してこの問題を回避できます。

また、SQL 関数を使用する場合、「AS 句」を使用して列名を必ず定義してください。AS 句を使用しない場合、MISP2 などのサービス実行時にエラーとなります。

例 (AS 句)

```
SELECT COUNT (*) as c FROM ...
```

【注記】

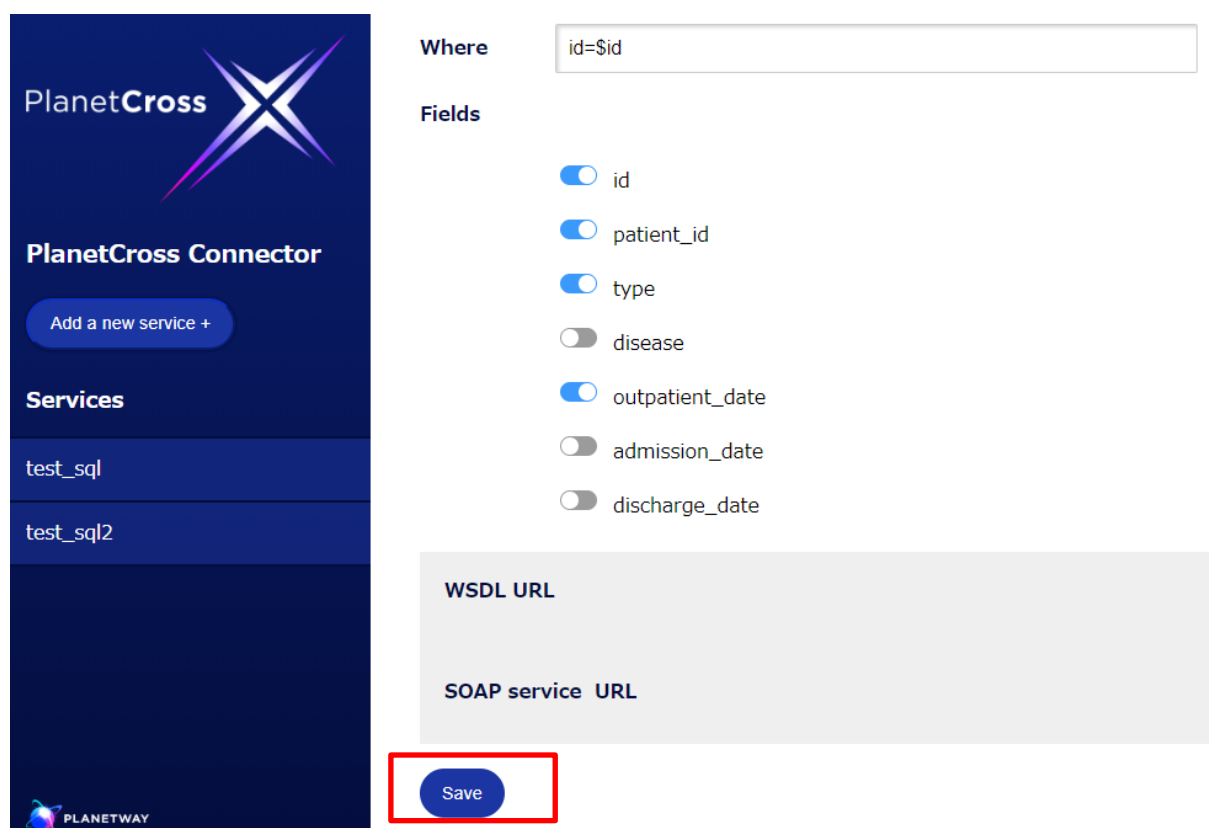
許容されていない列名のあるテーブルにアクセスするサービスは保存できません。

その他にも注意事項がございますので「[付録 A 注意事項](#)」を必ずご参照ください。

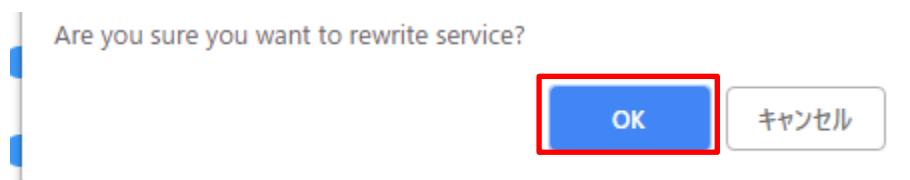
4.1.4 保存

サービスは使用前に保存する必要があります。保存後、WSDL URL を表示し、Security Server からサービスを利用できるようになります。

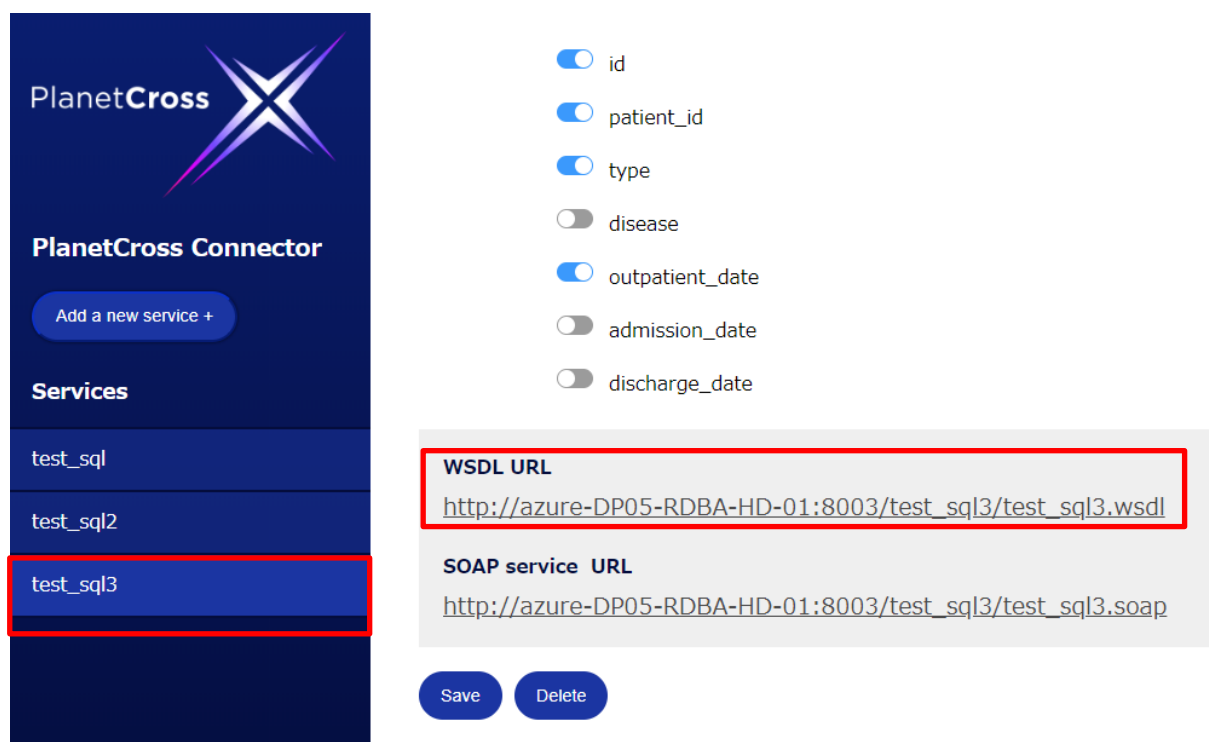
サービスを保存するには、[Save]ボタンを押します。



[Save]ボタンを押した後、確認のポップアップが表示されますので、[OK]ボタンを押してください。



Services に追加され、Security Server のサービス追加時に使用する WSDL URL が表示されます。



4.2 サービスの更新

Connector では、サービスのバージョン管理がサポートされません。管理 UI でサービス定義を更新して[Save]ボタンを押すと、以前のサービスが上書きされ、新しいサービスが直ちに使用可能になります。

たとえばクライアントにすでにサービスを提供している場合に以前のサービスが上書きされないようにするには、名前を変更してから[Save]ボタンを押して、サービスを別名で保存します。これにより、元のサービスはそのまま維持されます。

4.3 サービスの削除

サービスを削除するには、[Delete]ボタンをクリックします。削除の確定後、サービスは直ちに削除されます。

4.4 許諾のサポート

\$__targetUserId は、SOAP ヘッダーの targetUserId 要素に基づく、SQL で使用可能な特殊変数です。targetUserId は <http://xsd.planetcross.net/planetcross.xsd> で定義されます。

たとえば、アドバンスモードで作成されたサービスに以下のクエリーが含まれているとします。

```
SELECT * FROM users WHERE id = $__targetUserId;
```

ここで、以下の SOAP 要求が行われたとします。

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  ... snip ...
  xmlns:px="http://xsd.planetcross.net/planetcross.xsd"
  xmlns:xxforms="http://orbeon.org/oxf/xml/xforms">
  <SOAP-ENV:Header>
    <xrd:protocolVersion>4.0</xrd:protocolVersion>
    <xrd:id>c87e72c77f97581af1a7108b4f9d96b5e860a7f6</xrd:id>
    <xrd:userId>JP111111111</xrd:userId>
    <px:targetUserId>12345</px:targetUserId> <!-- this becomes the
$__targetUserId variable -->
    <xrd:issue />
    <xrd:service iden:objectType="SERVICE">
      ... snip ...
    </xrd:service>
    <xrd:client iden:objectType="SUBSYSTEM">
      ... snip ...
    </xrd:client>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    ... snip ...
```

この結果、SQL は以下のようになります。

```
SELECT * FROM users WHERE id = 12345;
```

シンプルモードでも、where 句を `id = $__targetUserId` として記述できます。

付録 A 注意事項

本章では Connector 設定時の注意事項を示します。

A.1 MySQL にて指定する変数名はカラム名と同一値に設定できません

[内容]

サービス登録時 (SQL 生成)、MySQL にて指定する変数名はカラム名と同一値に設定できません。

[回避策]

以下の通り、変数名とカラム名を同一値にしないようお願いします。

例 :

誤 : `id=$id AND patient_id=$patient_id`

正 : `id=$id AND patient_id=$patient_id1`

A.2 PostgreSQL にて指定する変数名の注意事項

[内容]

サービス登録時 (SQL 作成)、カラム名に全角数字を使用した場合、データが取得できないケースがあります (unicode 変換プログラム上のエラー)。

[回避策]

アドバンスモードで "AS" オプションでカラム名に別名をつけて、全角数字を半角数字に置き換えることでの回避可能です。シンプルモードでは回避不可となります。

クエリーの実行例 :

```
select code, title, "列名が 2 バイト文字" as "列名が 2 バイト文字", character_varying_256,
varchar_256, char_256, text from kanji where (code = $code);
```

改訂履歴

バージョン	日付	変更履歴
第 1 版	2019 年 2 月 28 日	初版発行
第 1.1 版	2019 年 3 月 29 日	文書内の表現を更新
		「Oracle データベースのセットアップ」の章を「Oracle データベース利用時の設定」として『Connector インストールガイド』の付録に移行
第 1.2 版	2019 年 6 月 12 日	「付録 A」を追加
		各種注意事項の取り込み
第 1.3 版	2019 年 8 月 1 日	各種注意事項の取り込み
第 1.4 版	2020 年 2 月 5 日	商標についての記載を追加。フッターを更新