

日々是Oracle APEX

Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2020年11月12日 木曜日

REST Data Source Connector Plug-Inについて

Oracle APEX 20.2で新たに提供された、REST Data Source Connector Plug-Inについて、どうやって作るのか不明だったので、少々調べてみました。簡単なものでもひとつ、作ってみたかったのですが、そこまでは出来ていません。

REST Data Source Connector Plug-Inもプラグインなので、**共有コンポーネント**の他のコンポーネントに含まれる**プラグイン**から作成します。



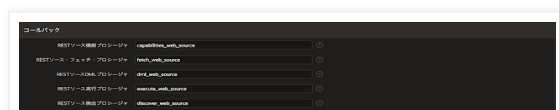
プラグインを開いて、**作成**を実行します。



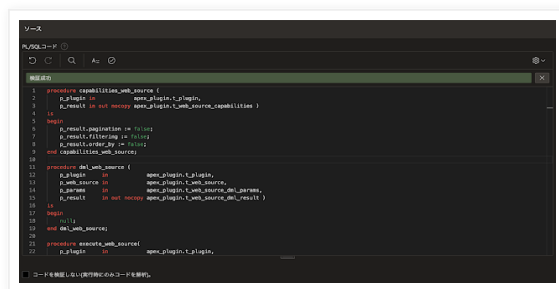
名前は任意で、ここではSimple REST Connectorとしています。**内部名**は全世界で一意になるような（プラグインをエクスポートして、他の環境にインポートしたときに内部名が競合しないように）文字列を設定します。**タイプ**は**RESTデータ・ソース**を選択します。



タイプとしてRESTデータ・ソースを選択すると、5つの**コールバック**の指定が現れます。これらのコールバックされる**プロシージャ**を実装すると、RESTデータ・ソースとして扱うことができるプラグインの出来上がり、ということになります。



コールバック・プロシージャには、**ソース**に直書きしたプロシージャの名前を指定するか、データベースに作成したプロシージャやパッケージに含まれるプロシージャ名を指定します。



RESTデータ・ソースのプラグインが登録されていると、**共有コンポーネントのRESTデータ・ソース**を作成する際の**RESTデータ・ソース・タイプ**として、作成済みのプラグインを選択することができます。



5つのコールバック・プロシージャですが、以下になります。

1. RESTソース機能プロシージャ
2. RESTソース・フェッチ・プロシージャ
3. RESTソースDMLプロシージャ
4. RESTソース実行プロシージャ
5. RESTソース検出プロシージャ

これらの定義について、それぞれまとめてみます。

RESTソース機能プロシージャ

以下の定義を持つプロシージャを実装します。プラグインに実装されている機能をAPEX側に伝えます。プラグインが提供可能な機能としては3つ、ページネーション、フィルタ、そして、ソート (order_by) です。

```
procedure capabilities_web_source (  
    p_plugin in apex_plugin.t_plugin,  
    p_result in out nocopy apex_plugin.t_web_source_capabilities )
```

apex_plugin.t_pluginやapex_plugin.t_web_source_capabilitiesの定義はどこを参照するのか探したところ、APEXのインストール・イメージを展開したディレクトリに含まれる core/www_flow_plugin_api.sql(パッケージWWW_FLOW_PLUGIN_APIの仕様部)から確認することができました。

最初の引数p_pluginの定義、**apex_plugin.t_plugin**は以下でプラグインの名前とか基本情報を渡してくれます。

```
-----  
-- Public type definitions  
-----
```

```
type t_plugin is record (  
    name          varchar2(45),  
    file_prefix   varchar2(4000),  
    attribute_01  varchar2(32767),  
    attribute_02  varchar2(32767),  
    attribute_03  varchar2(32767),  
    attribute_04  varchar2(32767),  
    attribute_05  varchar2(32767),  
    attribute_06  varchar2(32767),  
    attribute_07  varchar2(32767),  
    attribute_08  varchar2(32767),  
    attribute_09  varchar2(32767),  
    attribute_10  varchar2(32767),  
    attribute_11  varchar2(32767),  
    attribute_12  varchar2(32767),  
    attribute_13  varchar2(32767),  
    attribute_14  varchar2(32767),  
    attribute_15  varchar2(32767) );
```

p_resultとして返す値ですが、**apex_plugin.t_web_source_capabilities**として定義されています。

```
-----  
-- get capabilities response structure  
-----
```

```
type t_web_source_capabilities is record(  
    pagination          boolean default false,  
    filtering            boolean default false,  
    order_by            boolean default false );
```

なので、このプロシージャに関しては実装されている機能を設定して、以下のように返してあげればよいでしょう。

```
procedure capabilities_web_source (  
    p_plugin in          apex_plugin.t_plugin,  
    p_result in out nocopy apex_plugin.t_web_source_capabilities )  
is  
begin  
    p_result.pagination := false;  
    p_result.filtering  := false;  
    p_result.order_by   := false;  
end capabilities_web_source;
```

RESTソース・フェッチ・プロシージャ

以下の定義を持つプロシージャを実装します。いわゆるSELECT文の実装を行います。

```
procedure fetch_web_source(  
    p_plugin      in          apex_plugin.t_plugin,  
    p_web_source in          apex_plugin.t_web_source,  
    p_params      in          apex_plugin.t_web_source_fetch_params,
```

```
p_result      in out nocopy apex_plugin.t_web_source_fetch_result )
```

p_web_sourceは、このプラグインを使って作成されたRESTデータ・サービス(旧名Webソース・モジュール)で利用可能な操作やカラム名など、RESTデータ・ソースの構成情報を渡してくれます。型の定義はapex_plugin.t_web_sourceです。

```
-----
-- complete Web Source record passed to the Plug-In developer. Contains meta
-- data for all required web source operations.
-----

type t_web_source is record(
    id                number,
    --
    -- meta data array for all required Web Source Operations. For the "Fetch" procedure
    -- this is the Web Source operation assigned to the "Fetch Rows" database operation.
    -- For the "DML" procedure, the array will contain all Web Source Operations assigned
    -- to the Database Operations. For instance, the "Insert Row" operation can be accessed
    -- as follows: l_operation := p_web_source.operations( c_db_operation_insert );
    operations         t_web_source_operations,
    --
    profile_id         number,
    profile_columns    t_web_source_columns,
    --
    -- if the Plug-In developer has enabled the "Pass ECID" attribute, this attribute will
    -- contain the ECID to be passed to the REST Service. Otherwise this will be NULL.
    ecid              varchar2(32767),
    --
    -- Module Plug-In attributes configured within Web Source Module.
    attribute_01       varchar2(32767),
    attribute_02       varchar2(32767),
    attribute_03       varchar2(32767),
    attribute_04       varchar2(32767),
    attribute_05       varchar2(32767),
    attribute_06       varchar2(32767),
    attribute_07       varchar2(32767),
    attribute_08       varchar2(32767),
    attribute_09       varchar2(32767),
    attribute_10       varchar2(32767),
    attribute_11       varchar2(32767),
    attribute_12       varchar2(32767),
    attribute_13       varchar2(32767),
    attribute_14       varchar2(32767),
    attribute_15       varchar2(32767) );
```

operationsとして渡されるt_web_source_operationsの定義は以下です。

```
-----
-- table type containing multiple Web Source Operations. Required for the
-- DML procedure which needs Web Source operation meta data for the corresponding
-- database operations:
-- * Insert Row
-- * Update Row
-- * Delete Row
-- * Fetch Single Row (for Lost Update Detection / ETag)
-----
```

```
type t_web_source_operations is table of t_web_source_operation index by t_db_operation;
```

指定可能なデータベースの操作として、以下が定義されています。

```
-----  
-- Database operation constants  
-----
```

```
subtype t_db_operation          is pls_integer range 1..6;  
  
c_db_operation_fetch_rows      constant t_db_operation          := 1;  
c_db_operation_insert          constant t_db_operation          := 2;  
c_db_operation_update          constant t_db_operation          := 3;  
c_db_operation_delete          constant t_db_operation          := 4;  
c_db_operation_fetch_row       constant t_db_operation          := 5;  
c_db_operation_execute         constant t_db_operation          := 6;
```

それぞれのデータベースの操作をインデックスとして設定されている`t_web_source_operation`の定義は以下になります。

```
-----  
-- record type for one Web Source operation. This type is used to pass all  
-- Web Source Module meta data to the Plug-in.  
-----  
  
type t_web_source_operation is record(  
    --  
    -- module and operation ID, Plug-In developers can  
    -- use this to look up stuff in APEX dictionary views  
    module_id          number,  
    operation_id       number,  
    --  
    -- the URL is completely processed; all URL pattern replacements are done; all query string  
    -- parameters have been appended  
    url                varchar2(32767),  
    query_string       varchar2(32767),  
    --  
    -- HTTP Method configured within the Web Source Module. It's not mandatory for the  
    -- Plug-In developer to use this value; the Plug-In code can decide to execute  
    -- another HTTP method.  
    http_method        varchar2(255),  
    database_operation t_db_operation,  
    --  
    -- Request Body based on the Request Body Template configured in Web Source Module > Operation.  
    -- Parameter replacements are *not* made initially. Plug-In developers can call an "init"  
    -- procedure in order to perform placeholder replacements. Of course, Plug-In developers can also  
    -- generate a request body themselves.  
    request_body        clob,  
    request_body_format t_data_format,  
    --  
    -- Parameter values are already "resolved", i.e. the Plug-In code receives the  
    -- actual value based on the "Parameters" configuration for the APEX component.  
    parameters          t_web_source_parameters,  
    --  
    -- authentication information. Can be used to perform authentication calls.  
    auth_token_url      varchar2(32767),
```

```

auth_https_host      varchar2(500),
credential_static_id  varchar2(32767),
--
-- technical parameters required in order to execute the HTTP request.
encoding             varchar2(255),
https_host           varchar2(500),
timeout              pls_integer,
fetch_all_rows_timeout pls_integer );

```

つまり

`p_web_source.operations(apex_plugin.c_db_operation_fetch_rows).url`

や

`p_web_source.operations(apex_plugin.c_db_operation_fetch_rows).query_string`

という形でURLや問合せ文字列を取り出し、データベースの行フェッチに対応するREST APIの呼び出しを行うことになります。

データとして取り出せるカラムは`profile_columns`(定義は`t_web_source_columns`)として渡されます。

```

-----
-- record type for the Web Source data profile.
-----

```

```

type t_web_source_column is record(
    name                varchar2(255),
    is_primary_key      boolean                default false,
    is_filterable       boolean                default true,
    data_type           wwv_flow_exec_api.t_data_type,
    max_length          number,
    format_mask         varchar2(255),
    has_time_zone       boolean                default false,
    selector            varchar2(255),
    remote_attribute_name varchar2(255) );

```

```

type t_web_source_columns is table of t_web_source_column index by pls_integer;

```

カラムの名前、主キーかどうか、フィルタを使えるか、データ型などのカラム定義の配列です。

`p_params`として検索条件が渡されます。定義は`t_web_source_fetch_params`です。

```

-----
-- parameters record for the "Fetch Rows" procedure. Contains information about filters,
-- order bys and pagination.
-----

```

```

type t_web_source_fetch_params is record(
    --
    -- list of columns actually requested by the component. Can be used by the Plugin-Developer
    -- to restrict columns (or attributes) actually requested from the REST service. If an empty
    -- array is passed to the Plug-In, then APEX actually requests all data profile columns.
    requested_columns  t_web_source_columns,
    --
    -- Runtime filters coming from APEX components like IR, IG or Faceted Search

```

```

filters          wwv_flow_exec_api.t_filters,
--
-- External Filters configured at design time in Region Source. The Plug-In developer
-- is responsible for correctly merging this with the "filters" attribute.
external_filters  varchar2(32767),
--
-- Runtime order bys coming from APEX components like IR, IG or Faceted Search
order_bys        wwv_flow_exec_api.t_order_bys,
--
-- External Order bys configured at design time in Region Source. The Plug-In developer
-- is responsible for correctly merging this with the "order_bys" attribute.
external_order_bys varchar2(32767),
--
-- values context containing primary key values. This is used when e.g. Interactive
-- Grid fetches multiple rows by primary key. The Plug-In developer can use this
-- to execute optimized HTTP requests in order to implement these row fetches.
primary_key_values wwv_flow_exec_api.t_context,
--
-- indicates whether this is the initial request within the APEX page view for this
-- web source module.
initial_request   boolean,
--
-- First Row the APEX component is interested in. Translates to the "offset" parameter in an
-- "offset / limit" pagination scheme.
first_row         number,
--
-- Maximum Rows the APEX component is interested in. Translates to the "limit" parameter in an
-- "offset / limit" pagination scheme.
max_rows         number,
--
-- If a fixed page size as been configured in the Web Source Module attributes, this will be
-- passed in here.
fixed_page_size   pls_integer,
--
-- Passed in as TRUE, if the APEX component requested all rows, e.g. to compute an aggregation
-- like PIVOT or GROUP BY views.
fetch_all_rows    boolean,
--
-- "context" information to pass from one HTTP request to the next when fetching multiple pages. If the
-- HTTP request to the REST service returns some context information which is to be added to the request
-- for the next page, then the plug-in developer must return that information as "request_context". APEX
-- will pass this information as "t_web_source_fetch_params.request_context" for the next request.
request_context   varchar2(32767) );

```

これらの渡された情報を元にREST APIを発行して、p_resultに検索結果を返します。定義は **t_web_source_fetch_result** です。

```

-- response structure for the "Fetch Rows" procedure. Main purpose is to pass the response JSON
-- or XML back to the APEX engine. JSON or XML parsing is done by the APEX-Engine, and not by the
-- Plug-In developer.

```

```

type t_web_source_fetch_result is record(
--
-- response CLOB or JSON documents as array. If an invocation of the "Fetch" procedure

```

```

-- leads to multiple HTTP requests being made, all response CLOBs are returned using the array
responses          wwv_flow_t_clob,
--
-- indicate which row number the first row in the JSON response is. This is important
-- for pagination styles based on page size and page number. Example:
-- * APEX requests rows 81 to 120
-- * REST Service works with page size 50, so pages #2 and #3 need to be fetched
-- * the "response_first_row" result attribute needs to be set as "51", so the APEX engine knows that
--   30 rows have to be skipped.
-- * by default we assume that the JSON result starts at exactly the requested first row
(t_web_source.first_row)
response_first_row  number,
--
-- if we can determine the row count contained in the JSON responses, pass that back to the engine here. Unlike
-- the "total_row_count" this represents only the amount of rows contained in the JSON / XML documents
-- passed back as the "responses" array.
response_row_count  number,
--
-- if we can determine that there are more rows to fetch, pass that info back to the EXEC engine here.
has_more_rows       boolean          default true,
--
-- some REST services always return the *total number of result rows* as part of every response (page). This
-- attribute allows the plug-in developer to pass this information back to the APEX engine. For instance,
-- APEX reports can use this information to allow "X to Y of Z" pagination schemes. If the REST
-- services does not provide this information, this result attribute should remain NULL.
total_row_count     number,
--
-- "context" information to pass from one HTTP request to the next when fetching multiple pages. If the
-- HTTP request to the REST service returns some context information which is to be added to the request
-- for the next page, then the plug-in developer must return that information as "request_context". APEX
-- will pass this information as "t_web_source_fetch_params.request_context" for the next request.
request_context     varchar2(32767),
--
-- return filters which are *not* passed to the external REST Service. The EXEC engine will execute
-- those with the local SQL parsing query. Initialize with t_web_source.filters, i.e. perform local
-- filtering for all filters.
local_filters       wwv_flow_exec_api.t_filters,
--
-- pass received Web Source Module parameters back to the EXEC engine; e.g. to set page items.
out_parameters      t_web_source_parameters,
--
status_code         number,
reason_phrase       varchar2(32767));

```

検索結果はresponsesに、JSONまたはXMLのデータとして設定します。

RESTソースDMLプロシージャ

以下の定義を持つプロシージャを実装します。INSERT、DELETE、UPDATE文の処理を実装します。

```

procedure dml_web_source (
    p_plugin          in          apex_plugin.t_plugin,

```



```

p_web_source in          apex_plugin.t_web_source,
p_params     in          apex_plugin.t_web_source_dml_params,
p_result     in out nocopy apex_plugin.t_web_source_dml_result )

```

p_paramsとして、INSERT/DELETE/UPDATE文のどれかと、そのパラメータが渡されます。定義は **t_web_source_dml_params** です。

```

-----
-- parameters record for the "DML" procedure. Contains basically column meta data
-- and column values.
-----

type t_web_source_dml_params is record(
    --
    continue_on_error          boolean default false,
    lost_update_detection      wwv_flow_exec_api.t_lost_update_detection,
    --
    allowed_operations_column varchar2(255),
    --
    -- APEX_EXEC context object, containing row meta data and rows/columns values. Plug-In developers
    -- can use APEX_EXEC.NEXT_ROW and GET_XXX functions in order to access individual rows, columns
    -- and their values. APEX_EXEC.SET_VALUE functions allow to overwrite column values with RETURNING
    -- values received from the REST Service.
    -- This approach
    -- * requires a few additions to WWV_FLOW_EXEC and WWV_FLOW_EXEC API in order to deal with
    --   such a "Plugin-Values-Only" context
    -- * does not require us to expose the t_columns_values and t_row_statuses record structures,
    --   currently defined in wwv_flow_exec.
    insert_values_context      wwv_flow_exec_api.t_context,
    update_values_context      wwv_flow_exec_api.t_context,
    delete_values_context     wwv_flow_exec_api.t_context );

```

結果はp_resultで、定義は**t_web_source_dml_result**です。

```

-----
-- response structure for the "DML" procedure. Main purpose is to pass RETURNING values
-- and "Success" or "Error" status for each row back to the EXEC Engine
-----

type t_web_source_dml_result is record(
    --
    -- true if DML processing failed for at least one of the rows
    has_errors          boolean          default false,
    --
    -- status and RETURNING values for each row.
    insert_values_context wwv_flow_exec_api.t_context,
    update_values_context wwv_flow_exec_api.t_context,
    delete_values_context wwv_flow_exec_api.t_context,
    --
    -- pass received Web Source Module parameters back to the EXEC engine; e.g. to set page items.
    out_parameters      t_web_source_parameters );

```

RESTデータ・ソースが参照のみの場合、実装不要だと思われます。

RESTソース実行プロシージャ

以下の定義を持つプロシージャを実装します。APEX_EXEC.EXECUTE_WEB_SOURCEにて呼び出されたときに実行されるプロシージャです。データベースのストアド・プロシージャを呼び出す実装になります。

```
procedure execute_web_source(
    p_plugin      in          apex_plugin.t_plugin,
    p_web_source in          apex_plugin.t_web_source,
    p_result      in out nocopy apex_plugin.t_web_source_execute_result )
```

p_resultとして返す実行結果は、**t_web_source_execute_result**として定義されています。

```
-----
-- response structure for the "Execute" procedure.
-----
```

```
type t_web_source_execute_result is record(
    --
    -- pass received Web Source Module parameters back to the EXEC engine; e.g. to set page items.
    out_parameters      t_web_source_parameters );
```

実際には**t_web_source_parameters**として返される値が定義されています。

```
-----
-- record type for the Web Source parameter. This type is used to pass
-- information about Web Source Parameters to the Plug-In implementation.
-----
```

```
type t_web_source_parameter is record(
    name                varchar2(32767), -- reduce this to 255 for the Plug-In interface
    param_type          t_web_source_param_type,
    direction           t_web_source_param_dir,
    value               varchar2(32767),
    --
    -- this is for OUT parameters mapped to the Web Source Response body. That could
    -- be longer than 32k, so we need to be able to carry a CLOB
    response_body_value clob,
    --
    -- Only relevant for URL Query String parameters (c_web_src_param_query). If TRUE, then one Query String
    parameter
    -- will be generated for each member of the colon-separated string.
    is_array            boolean,
    --
    -- Only relevant for URL Query String, Cookie and HTTP Header Parameters. If TRUE, then the parameter will be
    -- completely omitted when its value is NULL.
    omit_when_null      boolean );
```

```
type t_web_source_parameters is table of t_web_source_parameter index by pls_integer;
```

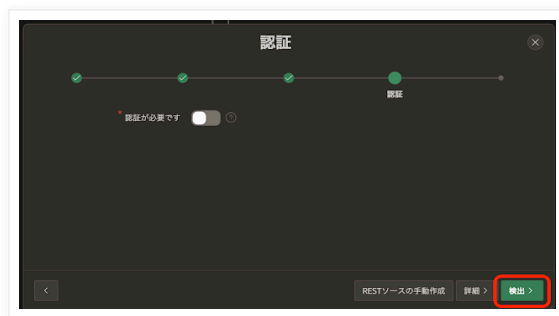
ストアド・プロシージャの呼び出しを行わない、参照のみのRESTデータ・ソースであれば実装は不要でしょう。

RESTソース検出プロシージャ

以下の定義を持つプロシージャを実装します。RESTデータ・ソースの検出を行ったときに呼び出されるプロシージャです。

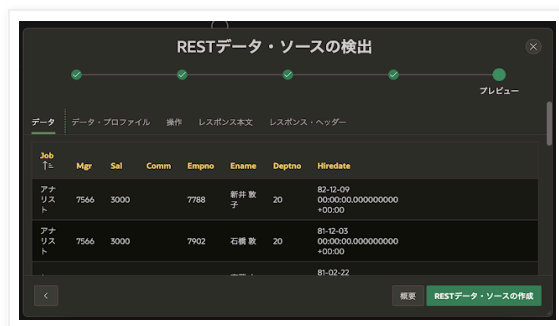
```
procedure discover_web_source (  
    p_plugin          in          apex_plugin.t_plugin,  
    p_web_source      in          apex_plugin.t_web_source,  
    p_param           in          apex_plugin.t_web_source_discover_params,  
    p_result          in out nocopy apex_plugin.t_web_source_discover_result)
```

検出が呼び出される画面です。



そして、検出された結果として、以下の画面から確認できるようなデータをp_resultとして返します。

データ



データ・プロフィール



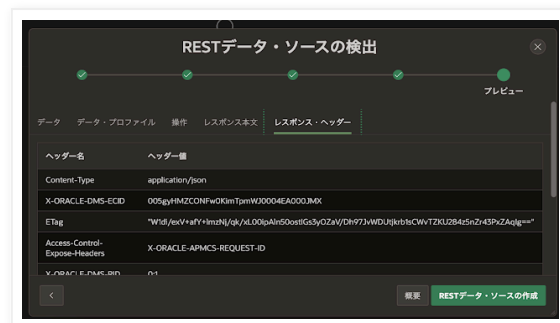
操作



レスポンス本文



レスポンス・ヘッダー



検出するにあたって渡されるパラメータp_paramの定義はt_web_discover_paramsです。今のところ使用されていません。p_web_sourceの情報を検出に使用します。

```
-- parameters record for the "Discovery" procedure. Not needed yet.
```

```
type t_web_source_discover_params is record(
    -- if a file (e.g. Response Sample) has been uploaded by the developer, use this for
    -- discovery.
    dummy          pls_integer );
```

検出結果をp_resultに返します。定義はt_web_source_discover_resultです。

```
-- discovery response structure; based on the Data Profile record types.
```

```
type t_web_source_discover_result is record(
    --
    -- discovery results for the service response: Data Format, JSON/XML selectors, data profile
    -- columns.
    data_format          t_data_format,
    row_selector         varchar2(32767),
    is_single_row        boolean,
```

```

profile_columns          t_web_source_columns,
xml_namespaces           varchar2(4000),
--
-- if the REST service returns pages of a fixed size, discovery can return the page size here
fixed_page_size          number,
--
-- The Plugin Developer can return a sample of the REST Response and have APEX discovering the
-- data profile columns and data types.
sample_response           clob,
--
-- discovered web source operations and plugin attributes, if applicable.
operations                t_web_source_operations,
parameters               t_web_source_parameters,
--
row_search_param_name     varchar2(255),
--
attribute_01              varchar2(32767),
attribute_02              varchar2(32767),
attribute_03              varchar2(32767),
attribute_04              varchar2(32767),
attribute_05              varchar2(32767),
attribute_06              varchar2(32767),
attribute_07              varchar2(32767),
attribute_08              varchar2(32767),
attribute_09              varchar2(32767),
attribute_10              varchar2(32767),
attribute_11              varchar2(32767),
attribute_12              varchar2(32767),
attribute_13              varchar2(32767),
attribute_14              varchar2(32767),
attribute_15              varchar2(32767),
--
-- return response headers and HTTP status code to the APEX engine
response_headers          wwv_flow_webservices_api.header_table,
status_code               number,
reason_phrase              varchar2(32767) );

```

data_formatの項目があります。この型は**t_data_format**で、XMLかJSONかを指定します。

```

=====
-- W E B      S O U R C E      M O D U L E S
=====

subtype t_data_format          is pls_integer range 1..2;

c_format_xml                   constant t_data_format := 1;
c_format_json                  constant t_data_format := 2;

```

概ね、検出後の画面の表示に含まれている情報を返しています。

終わりに

海外では**OData** (Open Data Protol)を扱うプラグインを実装できないだろうか、といった話も出ていました。RESTデータ・ソースにはプラグイン以外にも、データ・ソースの同期化（ローカル・データベースへのキャッシュと同じ）といった新機能も追加されているため、簡単に作れるとは言えないですが、RESTデータ・ソースのプラグインは作る価値があるだろうと感じます。

完

Yuji N. 時刻: 17:45

共有

◀

ホーム

▶

[ウェブ バージョンを表示](#)

自己紹介

Yuji N.

日本オラクル株式会社に勤務していて、Oracle APEXのGroundbreaker Advocateを拝命しました。
こちらの記事につきましては、免責事項の参照をお願いいたします。

[詳細プロフィールを表示](#)

Powered by Blogger.