How to invoke a GraphQL API with Oracle APEX

Nina Brötje APEX / Oracle, Datenbank 16.05.2024 0 Kommentare

Technical information

- Oracle APEX Version 23.2.4
- Pokemon API on https://stepzen.com/blog/free-public-graphql-apis

Introduction

GraphQL is a query language and runtime environment for APIs. Unlike REST, GraphQL allows clients to request only the spec overfetching and underfetching, thus improving data transfer efficiency. With its type system and query structures, GraphQL c way to exchange data between client and server.

Many people already know that a consumption of a GraphQL API is possible via the **apex_web_service** package in PL/SQL. H do this declaratively in Oracle APEX.

Easy way with PL/SQL

With PL/SQL it is very easy to consume a GraphQL API using the function **apex_web_service.make_rest_request**. You cons the GraphQL endpoint, specifying the GraphQL query or mutation in the request body. You then handle the response returne format, within your APEX application, parsing and processing the data as needed. This method allows you to interact with Gr APEX application's PL/SQL code.

```plsql
declare
    l_query         clob;
    l_url           varchar2(50)    := 'https://beta.pokeapi.co/graphql/v1beta';
    l_body          clob            := '{ "query": "{ pokemon_v2_pokemon(limit: 10) { height id name order pokemon_sp
    l_response      clob;
    l_status_code   number;
begin

    apex_web_service.g_request_headers(1).name :=   'Content-Type';
    apex_web_service.g_request_headers(1).value :=  'application/json';

    l_response := apex_web_service.make_rest_request(
        p_url           => l_url,
        p_http_method   => 'POST',
        p_body          => l_body
    );

    l_status_code := apex_web_service.g_status_code;

    if l_status_code = 200 then
        apex_collection.create_or_truncate_collection(
            p_collection_name => 'POKEMON_COLLECTION'
        );

        apex_collection.add_member(
            p_collection_name => 'POKEMON_COLLECTION',
            p_clob001         => l_response
        );
    else
        -- e.g. error handling
        null;
    end if;

end;
```

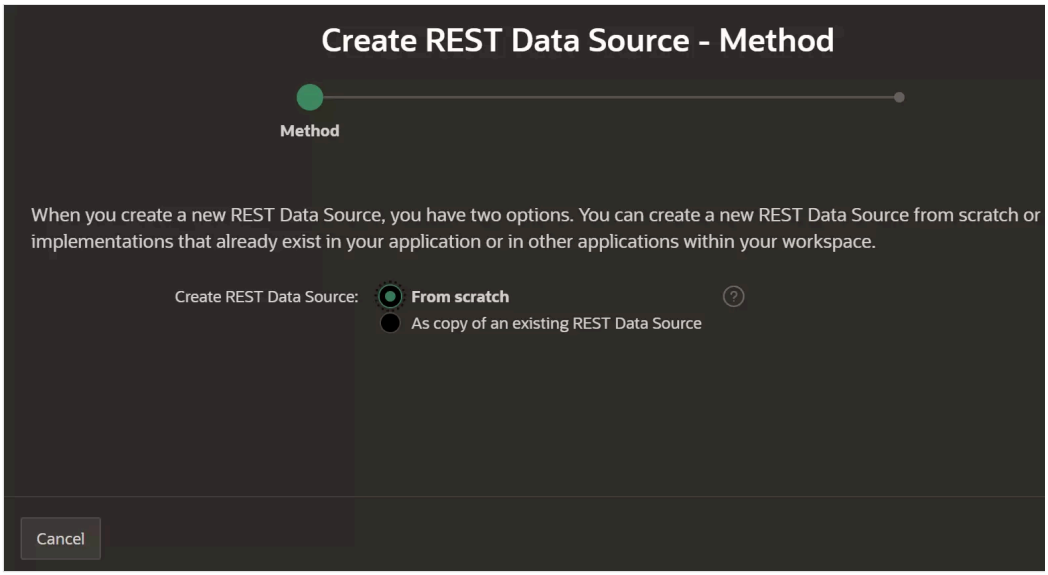
```plsql
select
    jt.id,
    jt.name,
    jt.height,
    jt."order",
    jt.species_id
from
    apex_collections c
    cross join json_table(
        clob001,
        '$.data.pokemon_v2_pokemon[*]'
        columns(
            id          number          path "id",
            name        varchar2(500)   path "name",
            height      number          path height,
            "order"     number          path "order",
            species_id  number          path pokemon_species_id
        )
    ) as jt
where
    collection_name = 'POKEMON_COLLECTION';
```

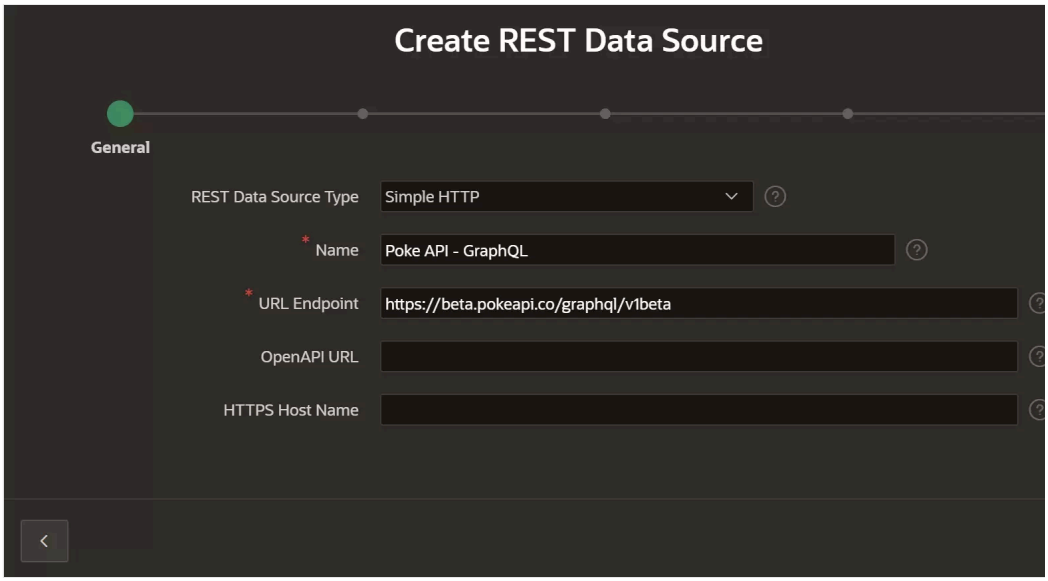
But can we not also do this declaratively in APEX? This has not yet been documented so far, but there is a very simple trick th

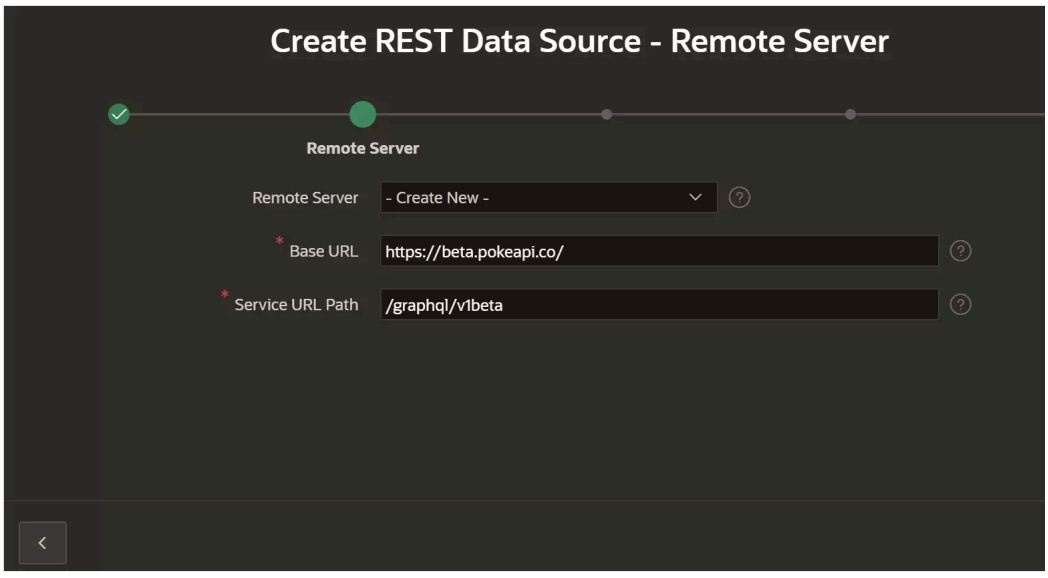
Do it declaratively in the APEX backend

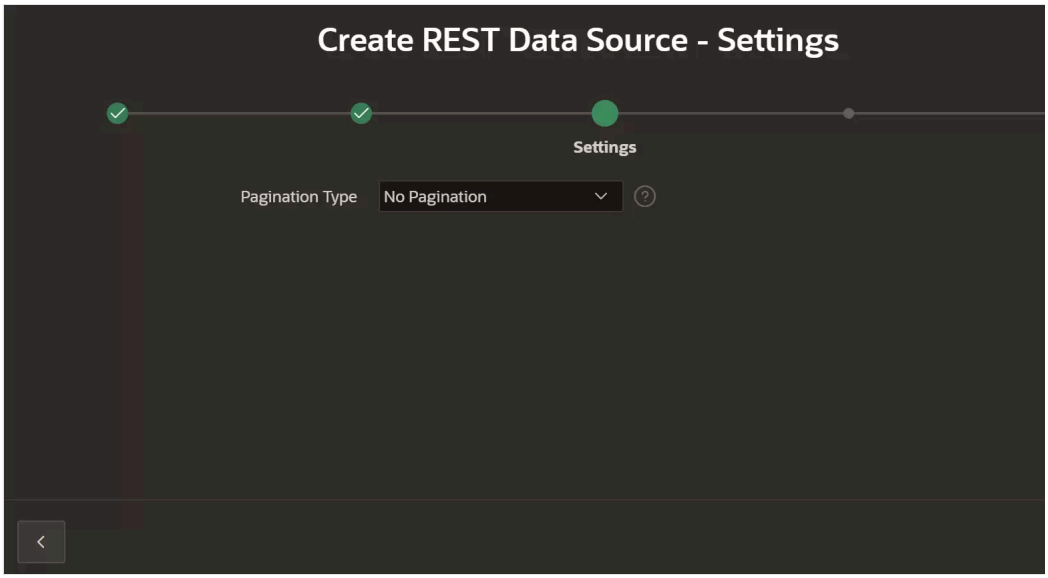
To be able to use a GraphQL API in Oracle APEX as a REST data source, we have to trick it a little bit.

The first steps are analogue to a simple REST API.


Create REST Data Source - Method
Method
When you create a new REST Data Source, you have two options. You can create a new REST Data Source from scratch or implementations that already exist in your application or in other applications within your workspace.
Create REST Data Source:
From scratch
As copy of an existing REST Data Source
Cancel


Create REST Data Source
General
REST Data Source Type
Simple HTTP
* Name
Poke API - GraphQL
* URL Endpoint
https://beta.pokeapi.co/graphql/v1beta
OpenAPI URL
HTTPS Host Name


Create REST Data Source - Remote Server
Remote Server
Remote Server
- Create New -
* Base URL
https://beta.pokeapi.co/
* Service URL Path
/graphql/v1beta

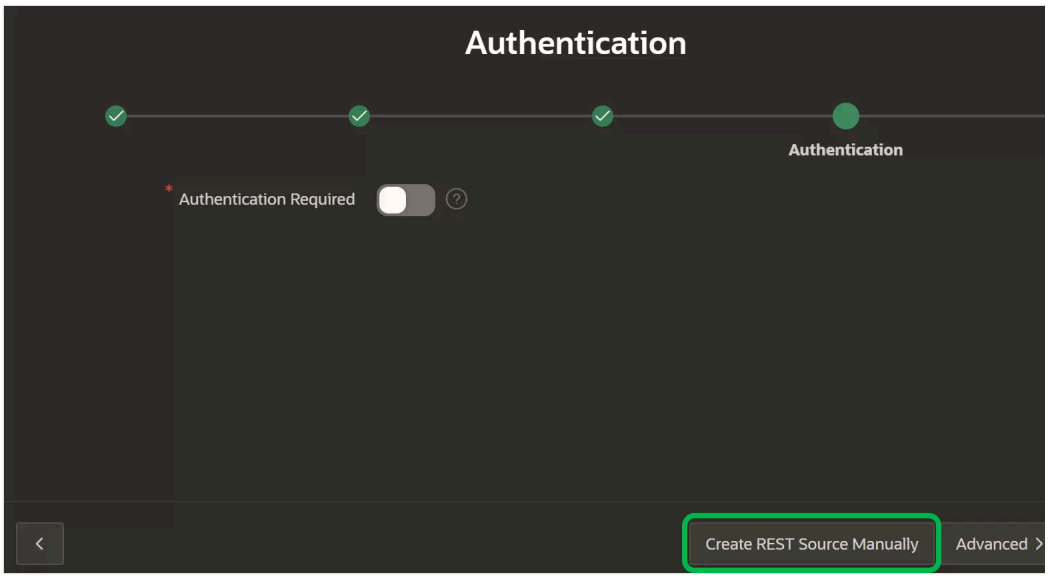


On the *Authentication* page of the wizard, it is now important **not** to click on *Discover*.

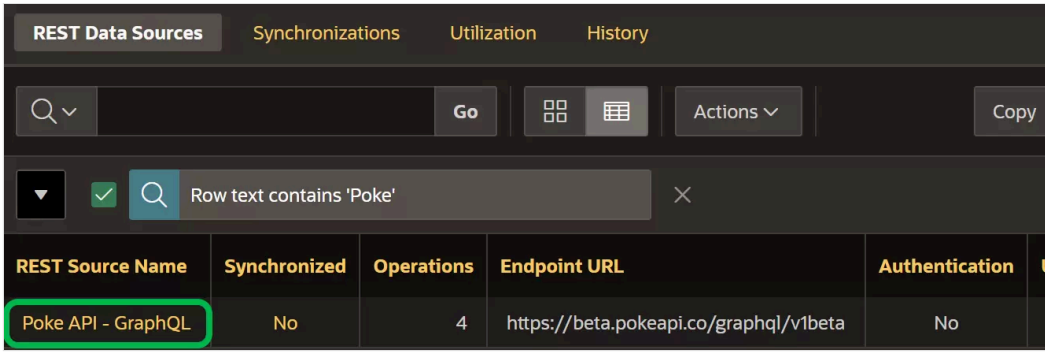
In this case, our new "REST Data Source" would be tested before it is created, which can lead to various errors ("GET query mi
Unknown", error of the API itself, etc.).

There are several reasons for this: A GraphQL API is always addressed with a POST method, whereby the request body contai
query. This is also used to fetch rows. Such a POST method can be created under *Advanced*. However, this is defined by defau
"fetch rows". APEX creates a GET for the "fetch rows" by default, but this is not supported by the GraphQL API.

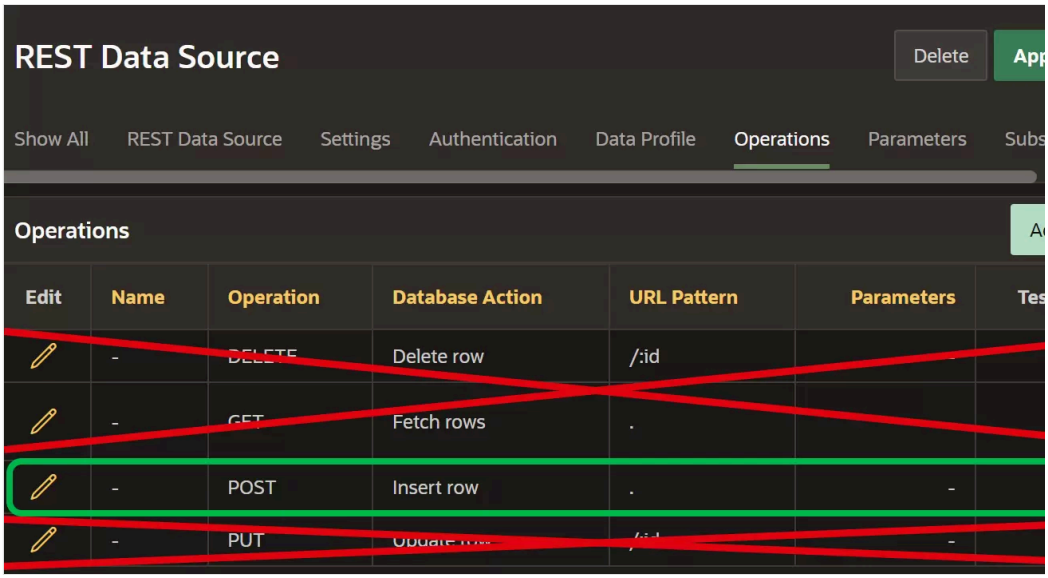
We click on *Create REST Source Manually* instead to prevent this.



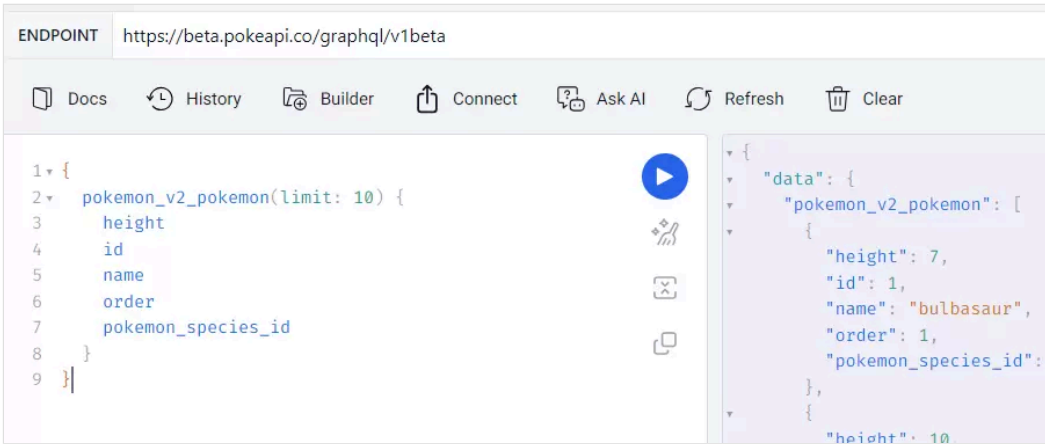
Now our GraphQL Data Source has been created. But we need to make a few more changes.



As we have created our GraphQL Data Source manually, all four HTTP methods are created automatically. We do not need DE
consume a GraphQL API. These are dropped. However, our POST method must be modified.

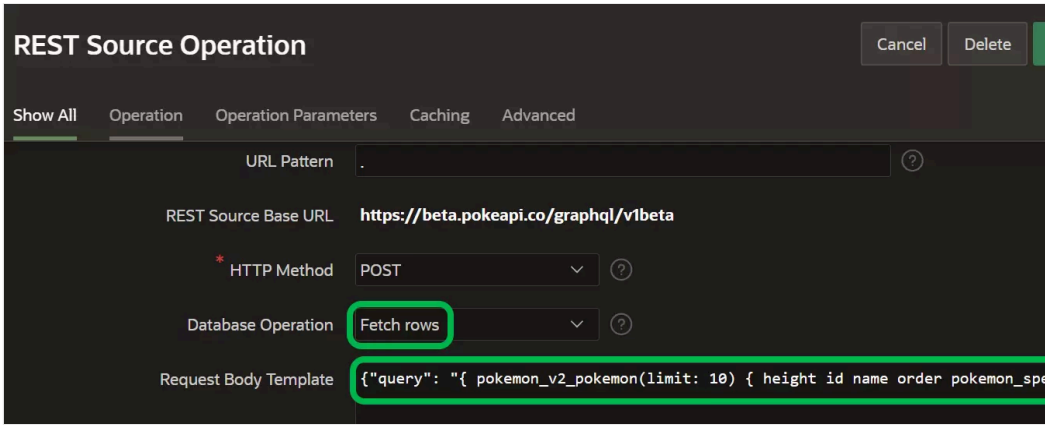


To do this, it is important to know what our GraphQL query and its JSON response look like (see the following illustration).

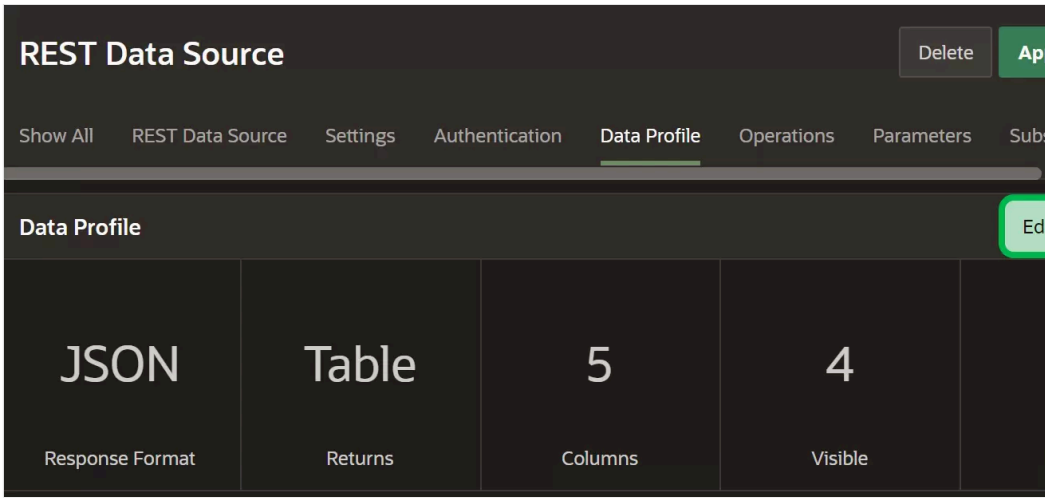


In our case, we want to use the POST method to fetch data. So, we set the database operation to "fetch rows".

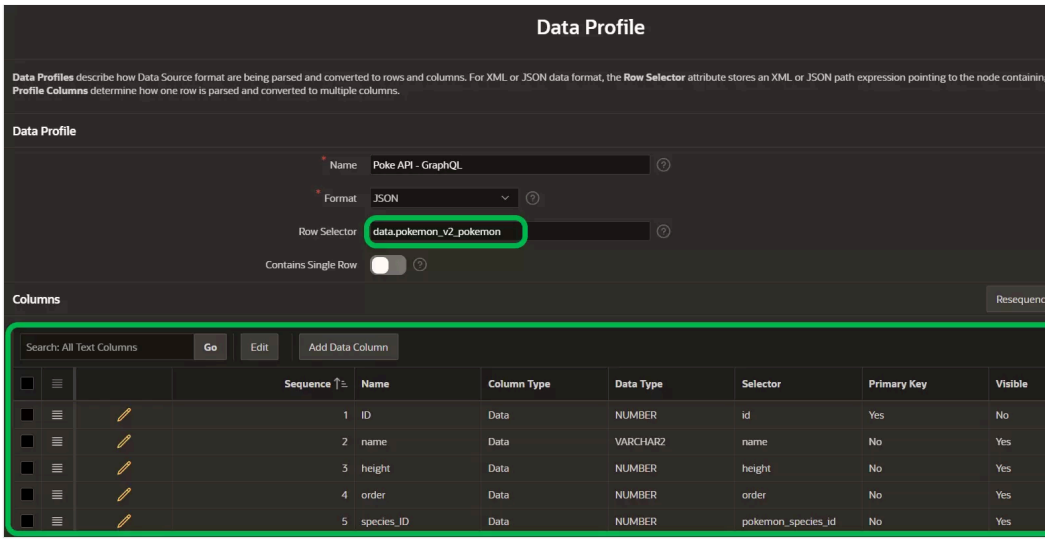
In the request body template, we insert our JSON body that is to be sent to the API. Attention: Paragraphs can lead to errors h



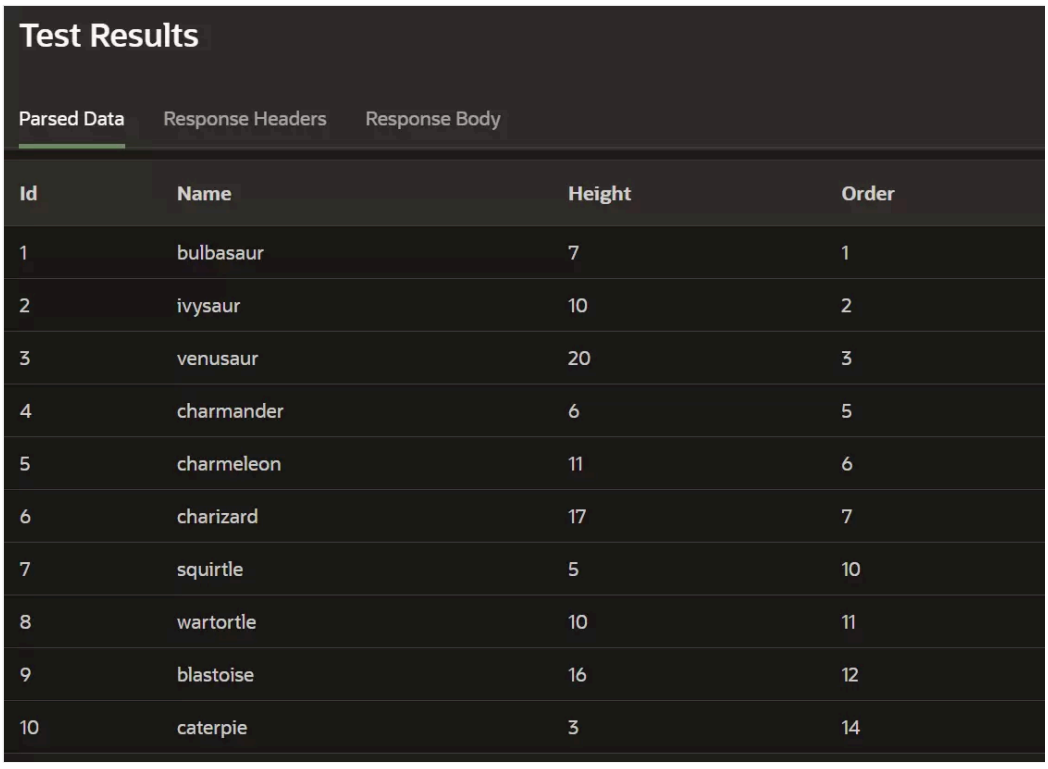
Next, we modify the data profile.



The row selector must be adapted according to the expected JSON response. The columns created by APEX by default are th columns are added.



Now let's test it.



Yay! Successful!

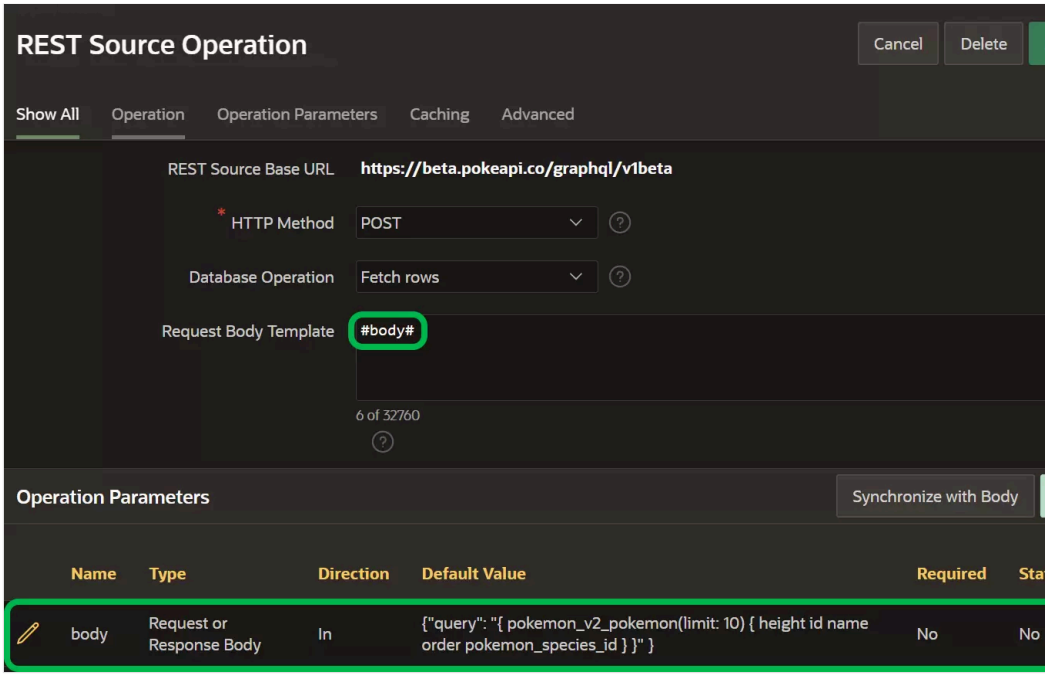
Now we can use this new GraphQL Data Source using its POST method for our APEX components.

Utilise the dynamic magic of GraphQL

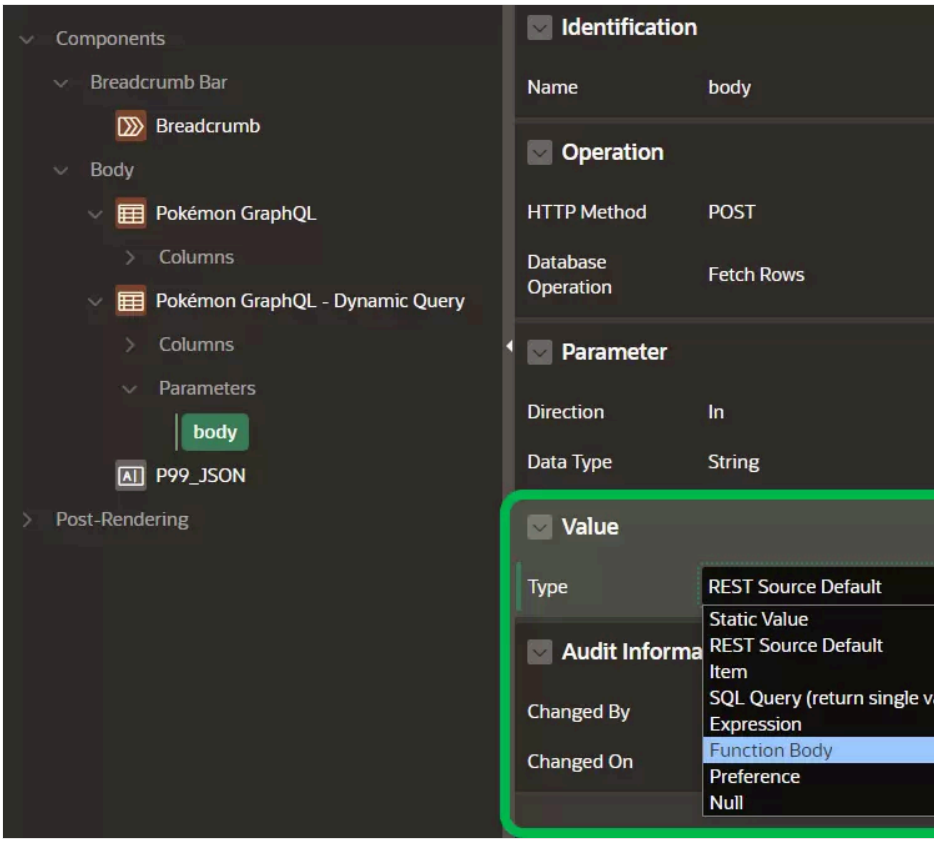
We are using GraphQL, which actually stands for dynamism. So, the example shown above is really for showing that the conn declaratively. But can we also make the whole thing dynamic? Of course we can!

To do this, we change our POST method once again

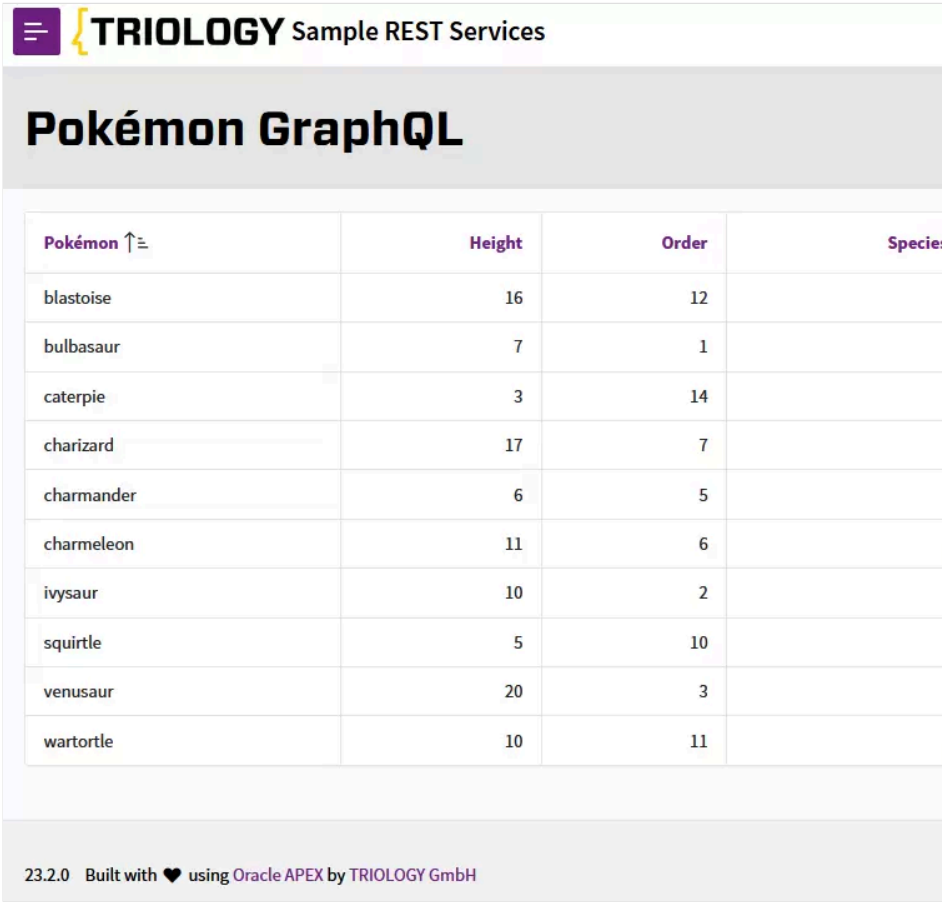
Set the *Request Body Template* to a substitution string, for example *#body#* and let it map to an operation parameter.



The corresponding parameter can then be used in the Page Designer and be set dynamically. It is important that the corresp JSON object that contains the correct GraphQL query.



And now we can use our GraphQL data source within our components.



Implement a GraphQL API with ORDS 23.3

As a small hint: Since ORDS 23.3 it is now also possible to implement GraphQL APIs with Oracle Database. Jeff Smith has writ this: https://www.thatjeffsmith.com/archive/2023/10/ords-23-3-is-now-available-or-hello-graphql/

Über den Autor

Nina Brötje

Kommentare

Keine Kommentare

Kommentar schreiben

Name *

E-Mail *

Kommentar *

Captcha*

Prüfen

* Diese Felder sind erforderlich

Absenden

Ähnliche Beiträge

How to import an Oracle APEX Application export file from a highe lower version

Maik Becker, Swathi Mary Ambati 08.03.2019

Recently we developed Oracle APEX application in our APEX 18.1 environment. Trying to share the tool with one of our cus they are still using 5.1.4 and won't be able to update within short time.

Mehrsprachige Datenbankanwendungen mit Oracle APEX

Swathi Mary Ambati 06.08.2018

Hallo Zusammen! Wenn du ein DBA/PLSQL-Guru bist, aber nicht viel Erfahrung in der Webentwicklung hast, dann ist Oracl besten Optionen. Egal ob Anfänger oder Fortgeschrittener, dieser Artikel gibt dir einen Einblick, wie eine mehrsprachige An erstellt werden kann. Hier zeigen wir dir, wie der in Oracle APEX integrierte Übersetzungsmechanismus funktioniert und wi Dieser Artikel basiert auf Oracle Application Express 18.1 Version.

Namenskonventionen in der Oracle Datenbank

Jan Niemann 31.07.2017

Ein wichtiger Teil von Code Conventions sind die Namenskonventionen. In den Namenskonventionen wird vereinbart, wie
Im Falle einer Datenbank handelt es sich bei den zu benennenden „Dingen“ um Schema-Objekte wie Tabellen, Sequenzen

Eine Chance für Azubis und Studis – die DOAG NextGen

Laura Fleischer 05.04.2023

Azubis! Studis! 2022 habe ich an der APEX Connect im Phantasialand teilgenommen und es war eine großartige Erfahrung.
Programm der Doag könnt ihr das auch!

Faceted Search with Kanban Board - #JoelKallmanDay

Maik Becker 11.10.2022

I decided to write this blogpost to give a better insight and explain, how easy it is to implement Faceted Search with differe
example with the Kanban Board.

Kategorien

Letzte Blogbeiträge

TRIOLOGY GmbH
Brabandtstraße 9-10
38100 Braunschweig
Deutschland

+49 531 23528-0

+49 531 23528-19

info@triology.de

Leistungen

Referenzen

Karriere

Unternehmen

Blog

Kontakt