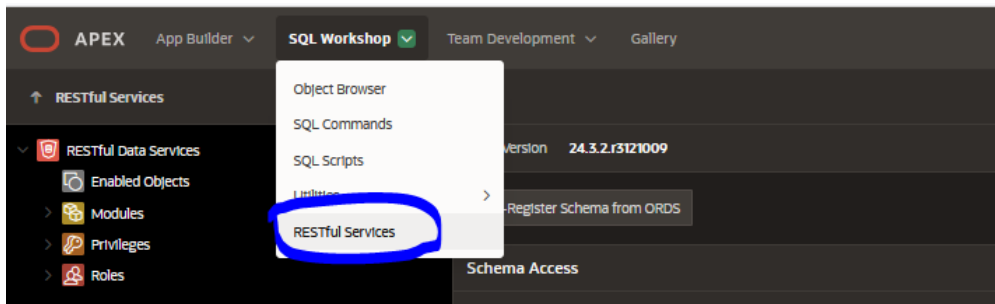


Keep Always Free Oracle Cloud APEX instance alive!

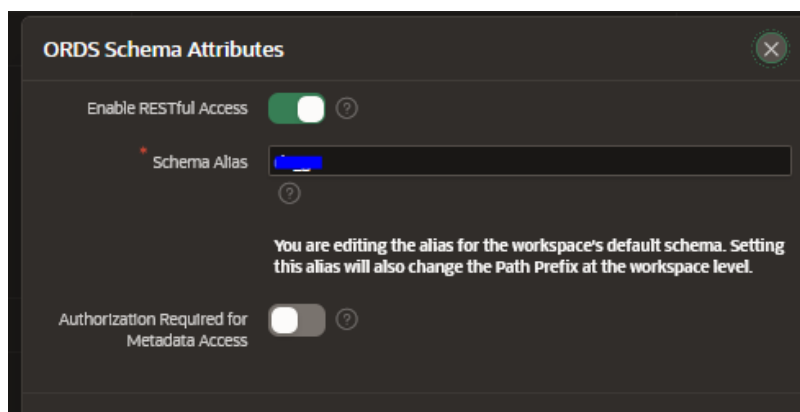
Wat meer details ...

REST-service

In het designer-deel van Apex vind je deze optie

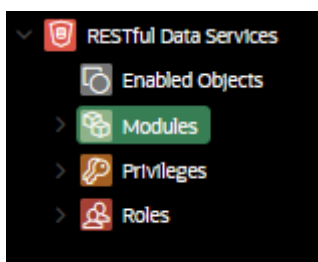


Click je dan op het hoogste niveau (RESTful Data Services) dan zie je een knop rechts in beeld om de RESTful Services te configureren.



"Enable RESTful Acces" moet aangevinkt worden om voor het schema de services aan te zetten. Onderin het window kun je aangeven of en wat voor autorisatie benodigd is om de REST-service te kunnen gebruiken.

Vervolgens kun je aan de slag en wanneer je dan op "Modules" klikt:



Dan zie je rechts in beeld een knop om een module toe te voegen.

The screenshot shows the APEX RESTful Services Module Definition interface. The left sidebar contains a tree view with 'RESTful Data Services' expanded, showing 'Enabled Objects', 'Modules', 'Privileges', and 'Roles'. The main panel is titled 'Module Definition' and contains the following fields:

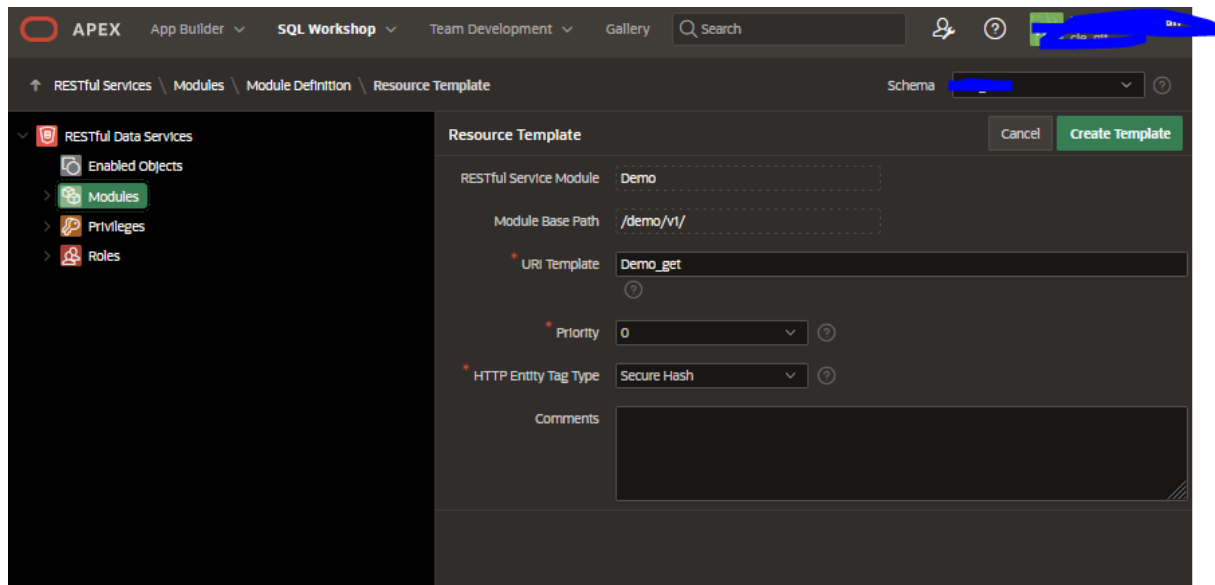
- Module Name:** Demo
- Base Path:** /demo/v1/
- Is Published:** ☒
- Pagination Size:** 25
- Origins Allowed:** (empty text field)
- Comments:** (empty text area)

At the top right of the main panel are 'Cancel' and 'Create Module' buttons. A legend at the bottom states: 'Legend: Template fully protected by a privilege Template not protected by any privilege'.

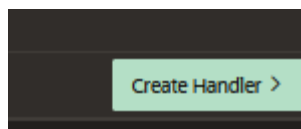
Wanneer je de naam van de module en het Base Path op deze manier invult en op Create Module klikt, dan is de eerste stap klaar. Je scherm ziet er dan ongeveer zo uit:

The screenshot shows the APEX RESTful Services Module Definition interface after successful creation. A green banner at the top reads 'Module Created'. The 'Module Definition' section now includes a 'Full URL' field with the value 'http://localhost:4242/demo/v1/'. The 'Apply Changes' button is now visible. Below the 'Module Definition' section is a 'Resource Templates' section with a search bar, 'Go' and 'Actions' buttons, and a 'Create Template >' button. A message at the bottom states: 'No Templates are currently defined for the selected module.'

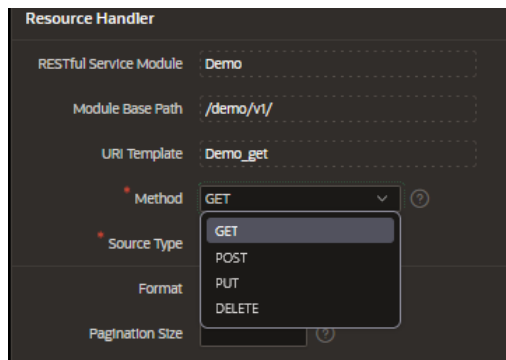
Vervolgens kun je dan een template toevoegen, hier definieer je uiteindelijk een POST-actie of een GET-actie. De eerste stap is dan:



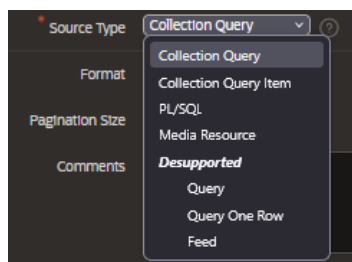
Nadat je op Create Template hebt geklikt kun je rechts in het scherm een Handler aanmaken via:



In de handler kun je dus die acties definiëren d.m.v. een dropdown:



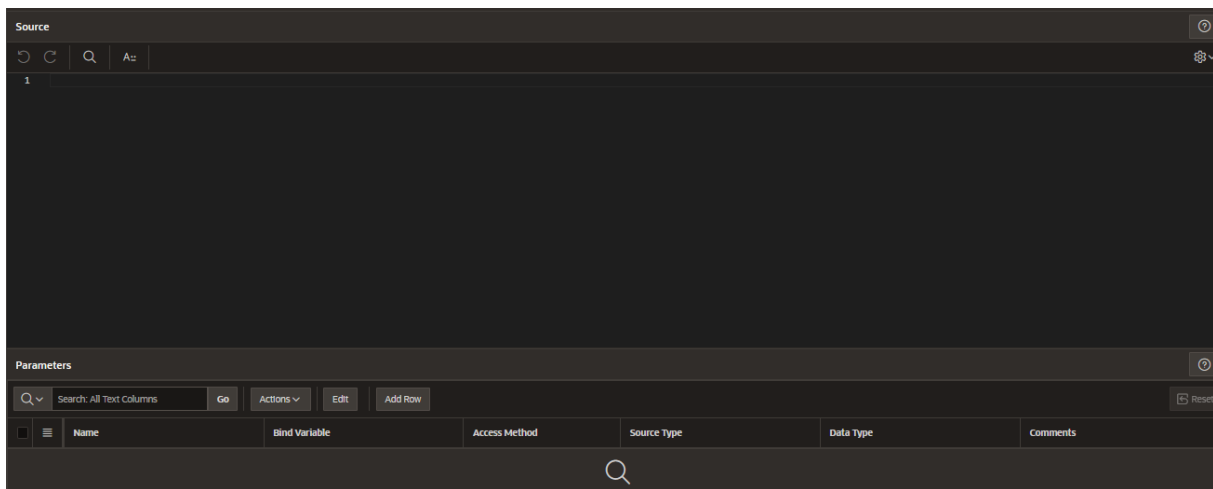
Ook het source Type is een dropdown:



(uitleg over de sourcetypes via <https://www.thatjeffsmith.com/archive/2019/02/ords-and-source-types/> of <https://docs.oracle.com/en/database/oracle/apex/22.1/htmldb/managing-REST-data-sources.html>)

Vervolgens kun je dan eventueel je PL/SQL-code in je source-scherm toevoegen en de verschillende parameters definiëren indien nodig.

Het source-deel kan maar zo'n 4k aan bytes aan, dus je zet hier een query neer of een call naar een procedure of functie (of combinatie van query en functie).



De parameters spreken voor zich. In de code kan dan bij het refereren aan deze parameters op de volgende wijze gebruikt worden:

```
name = demo_par_1      -> referentie : :demo_par_1
```

Dus wanneer je een procedure een parameter door moet geven doe je dat op deze manier:

```
proc_demo(i_rest_servicec_par_1 => :demo_par_1
          , i_par_dummy => '...'
          , i_enz ...
);
```

Je kunt in de afhandeling dan een logging aanmaken in een tabel waaruit achteraf af te leiden is wie de aanvraag heeft ingeschoten, hoe laat, met welke browser etc. Dit is in Oracle met `owa.init_cgi_env` af te handelen. (voorbeeld-script via <https://oracle-base.com/articles/misc/oracle-rest-data-services-ords-http-headers-and-ords-specific-bind-variables>)

Overzicht REST-services

Wellicht is deze query nog relevant, om een overzicht te krijgen van de ingevoerde REST-services:

```
select aa.*
  from ( select distinct a.name rest_naam, '/' || s.pattern || a.uri_prefix
        prefix, a.schema_id
          , b.uri_template service_naam
          , c.source_type
          , a.id
          , b.id templ_id
          , c.id handl_id
          , case when p.id is null
                  or b.uri_template = 'usage'
                then
                  '<<prefix-url>>../' || s.pattern || a.uri_prefix ||
b.uri_template
                else
                  null
                end theurl
          , case when b.uri_template in ( 'usage', 'open-api-catalog' )
                then
                  to_char( a.schema_id ) || '00001'
                else
                  to_char( a.schema_id ) || to_char( a.id)
                end lorder
        from user_ords_modules a
        join user_ords_templates b
          on b.module_id = a.id
        and b.schema_id = a.schema_id
        join user_ords_handlers c
          on b.id = c.template_id
        and c.schema_id = b.schema_id
        left join user_ords_parameters p
          on p.handler_id = c.id
        and p.schema_id = c.schema_id
        join user_ords_schemas s
          on s.id = a.schema_id
        ) aa
 order by 1,2, aa.lorder
;
```

Het resultaat is dat iets als:

	REST_NAAM	PREFIX	SCHEMA_ID	SERVICE_NAAM	SOURCE_TYPE	ID	TEMPL_ID	HANDL_ID	THEURL	LORDER
1	StartRegressieTesting_v1	/export/Test/Regressie/v1/	10009	usage	resource/lob	11804	12156	12157	<<prefix-url>>./export/Test/Regressie/v1/usage	1000900001
2	StartRegressieTesting_v1	/export/Test/Regressie/v1/	10009	Vraagaan	plsql/block	11804	11867	11884		1000911804
3	StartRegressieTesting_v1	/export/Test/Regressie/v1/	10009	Vraagaan	plsql/block	11804	11867	11884		1000911804

Dus de REST-service-naam, de gebruikte prefix, dat deel van de url die specifiek is voor die service, het schema_id waar de restservice beschikbaar is gesteld, de naam van de handler, het source-type welke gebruikt is en nog wat id's van de onderliggende gekoppelde gegevens.

Die service_naam "usage" geeft in mijn geval een html terug met daarin opgegeven hoe de REST-service voor die regressietesting te gebruiken, een soort swagger-file, maar dan niet in json-format, maar gewoon in plain-text.

De service_naam "Vraagaan" is dan de daadwerkelijke service.

Zo kun je dus een REST-service-naam op hoofdniveau opgeven en daaronder nog een aantal services (handlers) toevoegen.

in de "usage" of swagger (is met 1 click te genereren) kun je dan teruggeven via een tabelletje oid met uitleg van hoe en wat, per service.

Aanroepen REST-service vanuit Windows

In de Windows Task Scheduler (Taakplanner) definiëren we een taak die meermalen per week opgestart wordt. De taak zelf is het uitvoeren van een .bat script dat de volgende statements bevat:

