
Artículo

[Dani Fibla](#) · 9 ago, 2021 Lectura de 2 min

[Open Exchange](#)

REST API que consume SQL y devuelve JSON

¡Hola desarrolladores!

A veces, cuando desarrollamos una maqueta o una Prueba de Concepto (PoC), necesitamos una interfaz sencilla que proporcione datos en IRIS en JSON contra consultas SQL.

Y recientemente contribuí con un sencillo módulo que hace exactamente eso:

acepta una sentencia SQL y devuelve JSON.

¿Cómo se instala? Solo llama:

```
zpm "install sql-rest"
```

Si lo instalas en un namespace X, configurará un endpoint /sql en tu sistema, que aceptará solicitudes POST con una sentencia SQL y te devolverá el resultado para los datos disponibles en el namespace X.

¿Cómo probarlo? Por supuesto, necesitarás algunos datos de prueba que pueden ser instalados, por ejemplo, con:

```
USER>zpm "install dataset-countries"
```

Este es el flujo de prueba completo:

```
$ docker run --rm --name my-iris -d --publish 9091:1972 --publish 9092:52773 intersys  
temsdc/iris-community:2020.4.0.524.0-zpm
```

```
55f047cf0513a835bc31102c42d7ed02135ec195749d24aalfa4905a1c64d89c
```

```
$ docker exec -it my-iris iris session IRIS
```

```
Node: 55f047cf0513, Instance: IRIS
```

```
USER>zpm "install sql-rest"
```

```
USER>zpm "install sql-rest"
```

```
[swagger-ui] Reload START
```

```
[swagger-ui] Reload SUCCESS
```

```
[swagger-ui] Module object refreshed.
```

```
[swagger-ui] Validate START
[swagger-ui] Validate SUCCESS
[swagger-ui] Compile START
[swagger-ui] Compile SUCCESS
[swagger-ui] Activate START
[swagger-ui] Configure START
[swagger-ui] Configure SUCCESS
[swagger-ui] Activate SUCCESS
[sql-rest] Reload START
[sql-rest] Reload SUCCESS
[sql-rest] Module object refreshed.
[sql-rest] Validate START
[sql-rest] Validate SUCCESS
[sql-rest] Compile START
[sql-rest] Compile SUCCESS
[sql-rest] Activate START
[sql-rest] Configure START
[sql-rest] Configure SUCCESS
[sql-rest] Activate SUCCESS
USER>zpm "install dataset-countries"

[dataset-countries] Reload START
[dataset-countries] Reload SUCCESS
[dataset-countries] Module object refreshed.
[dataset-countries] Validate START
[dataset-countries] Validate SUCCESS
[dataset-countries] Compile START
[dataset-countries] Compile SUCCESS
[dataset-countries] Activate START
[dataset-countries] Configure START
```

```
[dataset-countries] Configure SUCCESS
```

```
[dataset-countries] Activate SUCCESS
```

```
USER>zn "%SYS"
```

```
%SYS>Do ##class(Security.Users).UnExpireUserPasswords("*")
```

```
%SYS>halt
```

```
$ curl -u _SYSTEM:SYS -X POST "http://localhost:9092/sql/query" -H "accept: application/json" -H "Content-Type: application/json" -d "SELECT * FROM dc_data.Country"
```

```
<
{
  "children":[
    {"ID":1,"code":"AGO","name":"Angola","continent":"Africa","region":"Central Africa","surface_area":1246700,"independence_year":1975,"population":12878000,"life_expectancy":38.3000,"gnp":6648,"gnp_old":7984,"local_name":"Angola","government_form":"Republic","head_of_state":"Jose Eduardo dos Santos","capital":56,"code2":"AO"}
  ]
}
```

...

Y eso es todo. Espero que os resulte útil.

[#InterSystems Package Manager \(IPM\) #SQL #InterSystems IRIS](#)

[Ir a la aplicación en InterSystems Open Exchange](#)

URL de fuente: <https://es.community.intersystems.com/post/rest-api-que-consume-sql-y-devuelve-json>