
Article

[Evgeny Shvarov](#) · Mar 3, 2021 2m read

[Open Exchange](#)

REST API That consumes SQL and Returns JSON

Hi folks!

Sometimes when we develop a mockup or PoC there is a need for a simple interface that will provide data in IRIS in JSON against SQL queries.

And recently I contributed a simple module that does exactly that:

accepts SQL string and returns the JSON.

How to install? Just call:

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

If you install it in a namespace X it will setup a /sql endpoint to your system that will accept POST requests with SQL string and will return the result for you for the data available in the namespace X.

How to test? Of course, you'd need some test data, that could be installed e.g. with:

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

Here is the full test flow:

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

```
55f047cf0513a835bc31102c42d7ed02135ec195749d24aa1fa4905a1c64d89c
```

```
$ 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":"Republ
    ic","head_of_state":"Jose Eduardo dos Santos","capital":56,"code2":"AO"}
  ]
}
```

```
...
```

That's it. Hope you'll find it useful.

[#InterSystems Package Manager \(IPM\)](#) [#SQL](#) [#InterSystems IRIS](#)
[Check the related application on InterSystems Open Exchange](#)

Source URL:<https://community.intersystems.com/post/rest-api-consumes-sql-and-returns-json>