

DRS-Database Rest Services

Quick Configuration



DRS – Database Rest Services - Quick Configuration

This document presents a sequence for creating the necessary records to have a database ready to use after installation. The meaning of the fields present in the pages is described in the SDD document.


This sequence assumes the the DRS application is started using the line:

```
java -Dspring.config.location=<setup folder>\sdd.properties  
-DconfigDir=<setup folder> -jar DRS.war
```

where **<setup folder>** is the folder that contains the drs.properties and all properties to access the databases defined in the DRS application. The drs.properties can be copied from GitHub and adapted to meet the current environment.

The first page sent is the login page. In this case, the only user that can be used is the **ADMIN**, and the password is **admin**, inserted as part of the installation.

The environment is displayed on this page and this environment must be defined after the login.



DRS - Database Rest Services Login -

User Name Password

If the username and password are informed correctly, the menu page is displayed. The corresponding environment is displayed in the upper left corner of the page and the current user is displayed in the upper right corner.

**DRS-Database Rest Services**User: ADMIN

[Home](#) [Admin ▼](#) [Services ▼](#) [Support ▼](#) [Help ▼](#) [Logoff](#)

The only information present in the database is the record related to the user ADMIN. Selecting the menu **Admin** and the option **Users**, the **Users** query page is displayed with only one record.

DRS – Database Rest Services - Quick Configuration

Users									
Userid: <input type="text"/> Search Add New User									
	User Id	Name	Service User	Oper User	Admin User	Super User	Last Access	Last IP	Status
	ADMIN	ADMINISTRATOR	Yes	Yes	Yes	Yes	26/08/2021 11:47:36	192.168.0.199	Active
(1 of 1) 1									

The next step is to create a server. In the **Admin** menu, select **Servers**. The server query page is displayed and is empty.

Servers							
Refresh Add New Server							
	Server Name	Server Type	IP	Max.Pool	Status	Last Update	Updated By
No records found.							
(1 of 1)							

By pressing the **Add New Server** button, the server page is sent. The **Server Name** can be the network name or an alias.

Server Name

Server Name

Server Type ☐ MCP ☒ SQL ☐ External ☐ Application Server

IP Address

Max Pool Size

Status * ☒ Active ☐ Suspended

Update Date

Updated BY

[Back](#) [New](#) [Delete](#) [Save](#)

In this case, the server svr01 has an IP address of 192.168.0.199. The server type selected is SQL, which means the connection will use a JDBC connection and a relational database.

The next step is related to the database to be accessed. The configuration is made by selecting the menu **Admin** and the **Databases** option. The databases query page is displayed.

DRS – Database Rest Services - Quick Configuration

The screenshot shows a web interface titled "Databases" with a red header. Below the header, there are input fields for "ServerName:" and "Database:", followed by "Search" and "Add New Database" buttons. A table with columns "Database Name", "Server Name", "DB Type", "Status", "Last Update", and "Updated By" is shown. The table contains one row with the text "No records found." Below the table, there is a red bar with "(1 of 1)" and navigation icons. A small Excel icon is visible in the bottom right corner.

By pressing the Add New Database, the database page is shown. To define a database, it is necessary to define the **Database Name** and the **Server Name** where this database is present.

If the database type is **DMSII MCPSQL**, the database name must have the same name defined in the **MCPSQL/CONFIG** file. The file **mcpsql.properties** must be present in the **SWDIR_SETUP** folder. GitHub has a sample of this file (**mcpsql.properties**).

If the database type is **DMSII JDBC**, the database name must have the name used in the configuration file stored in the folder pointed by the **SWDIR_SETUP** environment variable. If the database name is **DBTST**, the file **DBTST.properties** must be present in the **SWDIR_SETUP** folder. GitHub has a sample of this file named **DMSII_JDBC.properties**.

The database type selected in this sequence is **SQL**, which means that the properties file with the configuration must be present in the **SWDIR_SETUP** directory with the name **ORACLE1.properties**. The sample properties file for this database type can be accessed in GitHub, with the name **JDBC.properties**.

The screenshot shows a web interface titled "Database" with a red header. Below the header, there are input fields for "Database Name *" (ORACLE1) and "Server Name *" (SVR01). Below these, there are radio buttons for "Database Type *" (DMSII JDBC, DMSII MCPSQL, SQL). The "SQL" radio button is selected. Below the radio buttons, there are radio buttons for "Status *" (Active, Suspended). The "Active" radio button is selected. Below the status radio buttons, there are input fields for "Update Date" and "Updated BY". At the bottom, there are buttons for "Back", "New", "Delete", and "Save".

After the database creation, the first service can be defined. Select **Services** on the **Services** menu and the query page will be displayed with no records.

DRS – Database Rest Services - Quick Configuration

Services										
Service Filter:				Query		New Service				
Service Name	Description	ServerName	Database Name	Service Type	Avg.Alert	Max.Alert	Rows Per Req.	Role Name	Sec. Enable	
No records found.										

By pressing the **New Service** button, the service detail page is shown.

Service

Service Name *

service01

Server Name *

SVR01

Database Name

ORACLE1

Description

Test service - Oracle database

Service Type

☒ SQL ☐ URL

Avg.Response Time Alert

0.0000

Max.Response Time Alert

0.0000

Rows Per Request

0

Action *

select * from client

Role Name

Select One

Return Data Only *

☐ Yes ☒ No

Parameter Type *

☐ Header ☒ Body

Security Enabled *

☐ Yes ☒ No

Stored Statistics *

☒ Yes ☐ No

Status *

☒ Active ☐ Suspended

Update Date

Back

New

Delete

Save

Parameters

After creating the record, the service is ready to be used. You can run Postman to request the service **service01**.

DRS – Database Rest Services - Quick Configuration

GET ▼ http://192.168.0.199:8095/Drs/services?service=service01 Send ▼

Params ● Authorization Headers (9) Body ● Pre-request Script Tests Settings

Query Params

	KEY	VALUE
<input checked="" type="checkbox"/>	service	service01
	Key	Value

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▼ ≡

```
1  {
2    "records": 5,
3    "responseTime": 0.0661857,
4    "time": "26-08-2021 12:13:19.803",
5    "items": [
6      {
7        "CREDIT_LIMIT": 10000,
8        "CITY": "NEW YORK",
9        "COUNTRY": "USA",
10       "CLIENT_ID": 1,
11       "CLIENT_NAME": "GOOGLE",
12       "ADDRESS": "HARTFORD MAROON RD, 569",
```