

LAB- Using Database Endpoint

In this lab you will understand

- a) how to use a database endpoint to fire a select query
- b) how to fire a query with parameters

Before you proceed, please make sure that you have installed mysql database latest version and also any sql client.

Make sure that you give password as root since all my examples use root/root as username/password

Execute **muletrainingdb.sql** to make sure that database, tables are created and data is populated.

Note: In this exercise, I will be using mysql database. If you want to use any other database, you should be able to use. But my muletrainingdb.sql may not execute because I wrote it specific to mysql.

Also, regarding maven dependencies to be added for other databases, please google.

STEP 1

1) Create a mule project with name **01-databaseendpoint-start**.

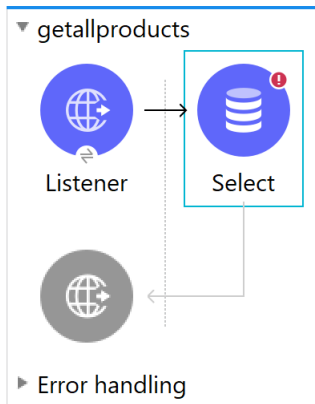
create a file with name databaseendpoint.xml.

On the right side mule palette, add database module

Create another file with name globals.xml.

From now onwards, when ever you are asked to create any global element, make sure that you do it in globals.xml

2) Drag and drop a http listener and database select component as shown below.



Make sure that you configure Http Listener for <http://localhost:8081/db> (configure global element in globals.xml)

2) Create database connector configuration for mysql as shown below :

Database Config

Default configuration

General Advanced Notes Help

Name: Database_Config

Connection: MySQL Connection

General Transactions Advanced

Required Libraries

MySQL JDBC Driver Please add the required driver. Configure...

Connection

Host: localhost

Port: 3306

User: root

Password: ☐ Show password

Database: muletrainingdb

3) Click on Configure button in database configuration and select “Add Maven dependency”

Give the version as 8.0.16 if your mysql version is 8. If your are using Mysql 5,

take the default

Open pom.xml and verify that u have the following dependency: **(U can select 8.0.16 if your mysql server version is 8)**

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.17</version>
</dependency>
```

Click on Global Elements Tab of xml , edit Database config and Click on Test Connection and verify that connection is succesfull.

Write the query as below :

Select * from product

Drag a Logger after select and configure it to log the payload.

6)Now we want to convert output of “Select ” component to Json.

Drag TransformMessage component after “Select ” component. Configure it to transform to json

If we give request to <http://localhost:8081/db> and observe the json.

If we give request to <http://localhost:8081/db?brandname=Apple> , We want all the products whose brand_name is Apple.

Modify the query to use the query parameter brandname as show below :

```
select * from product where brand_name=:brandname
```

configure input parameters as shown below :

```
{  
    brandname: attributes.queryParams.brandname as String  
}
```

Now deploy and test by giving request to following URL:

<http://localhost:8081/db?brandname=Apple>

STEP 2 (Optional) :

Can you modify the above flow to use choice router such that if there is a query parameter brandname, it should display products by name from database . else, it should display all products from database