**CALLING STORED PROCEDURES**

# Introduction:

**Stored Procedure** : A stored procedure is a prepared SQL code that you can save, so the code can be reused repeatedly.

So, if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.

You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.

# Creating Stored Procedure:

You can use any database to connect in my case I am using MYSQL Database.

**Create a Procedure** **in MySQL** :

Ex:

Use sales;

delimiter //

create procedure customers()

begin

select \* from customer ;

end //

delimiter ;

# Creating a Data Service :

Create a new Integration Project.

(For Creating a new Integration Project refer to the basic creation of Integration Project.)

1. **Right Click on the Project and select new Data Service Configs.**

Graphical user interface, application

Description automatically generated

1. **Give the Module Name and click on Finish.**

Graphical user interface, application

Description automatically generated

1. **Right click on the dataservice.**
2. **Select new and click on Data Service.**

Graphical user interface, application

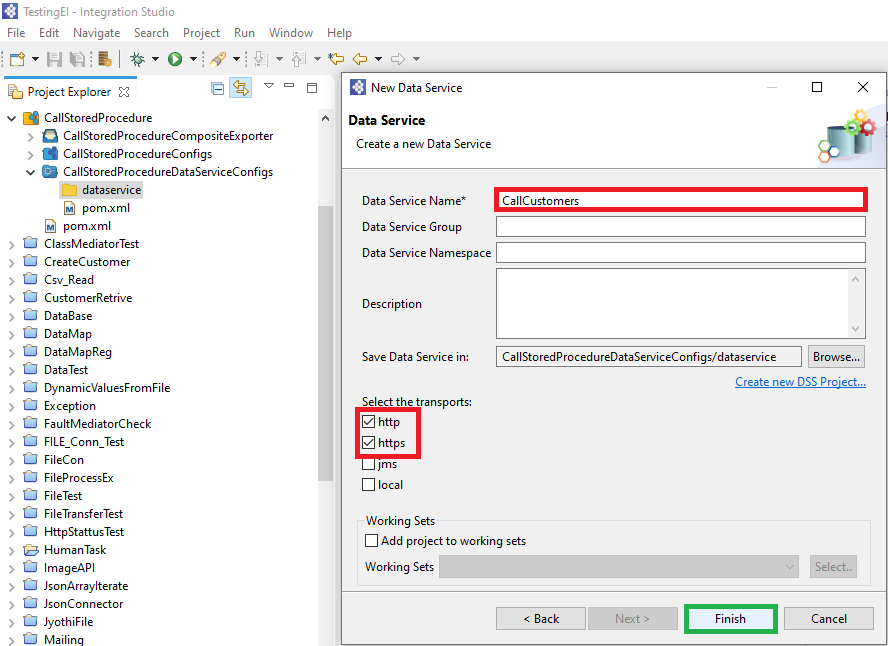
Description automatically generated

1. **Select Create new Data Service and click on Next.**

Graphical user interface, text, application

Description automatically generated

1. **Give name to Data Service, Select the http, https transports and click on finish.**

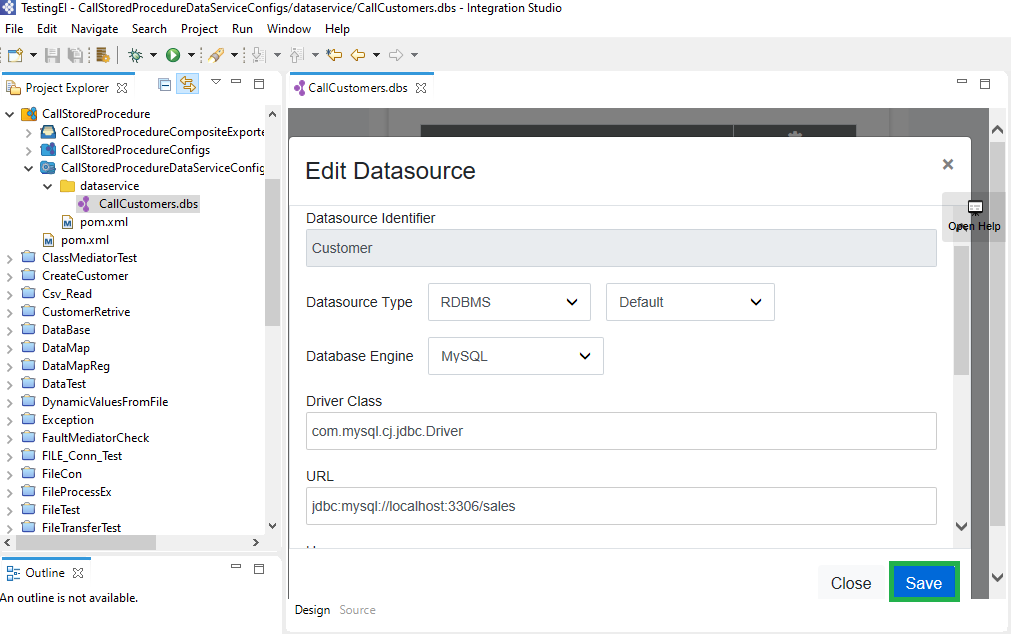
****

1. **Click on Config Data Source and click on Add New.**

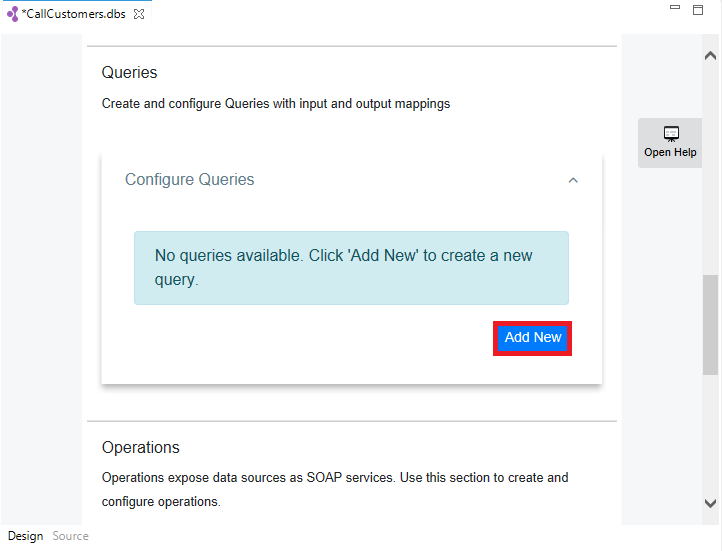
**Graphical user interface, text, application

Description automatically generated**

1. **Configure the Data Source and click on save.**

****

1. **Click on Queries and click on Add New.**

****

1. **Give the Query id.**
2. **Select the Data Source.**
3. **Write the SQL Query and Set the Result Out Put Mapping.**

**Note : While Setting the result output Mapping Group by Element Name is Mandatory and the Row name is optional.**

**To retrieve the multiple records, we need the root element so the Group By Element is Mandatory.**

**Graphical user interface, application

Description automatically generated**

**Sample Source Code For Query:**

<query id=*"getCustomers"* useConfig=*"Customer"*>

<sql>call customers()</sql>

<result rowName=*"Customer"* element=*"CustomerAll"*>

<element name=*"Customer\_id"* xsdType=*"string"* column=*"Customer\_id"* />

<element name=*"firstName"* xsdType=*"string"* column=*"firstName"* />

<element name=*"lastName"* xsdType=*"string"* column=*"lastName"* />

<element name=*"dOB"* xsdType=*"string"* column=*"dOB"* />

<element name=*"title"* xsdType=*"string"* column=*"title"* />

<element name=*"marital\_Status"* xsdType=*"string"* column=*"marital\_Status"* />

<element name=*"occupation"* xsdType=*"string"* column=*"occupation"* />

<element name=*"gender"* xsdType=*"string"* column=*"gender"* />

</result>

</query>

**Graphical user interface, application

Description automatically generated**

1. **Click on the Resources and click on Add New.**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

1. **Give the resource path, resource method, query id and click on save.**

**Graphical user interface, application

Description automatically generated**

**The Configuration for the Data Service is done.**

1. **Now add the Data Service Artifact in the CAR file and Run the Server.**

**Copy the URL and Paste it in the SOAP UI.**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

1. **Now give the URL in the new SOAP endpoint and click on OK.**
2. **Navigate to the resource path and run the End Point.**

**Graphical user interface, text, application

Description automatically generated**

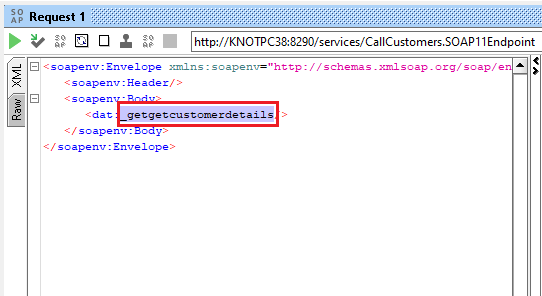
**Now we got the Result.**

**Graphical user interface, text, application

Description automatically generated**

# Converting the SOAP to REST API

* Create a New REST API. (For Creating the REST API please refer to the basic document to create a REST API.)
* ADD a Log Mediator
* Add the Header Mediator.
* SET :
* Header Name = Action.
* Value as mentioned in the <dat/> tag.

****

**Graphical user interface, application

Description automatically generated**

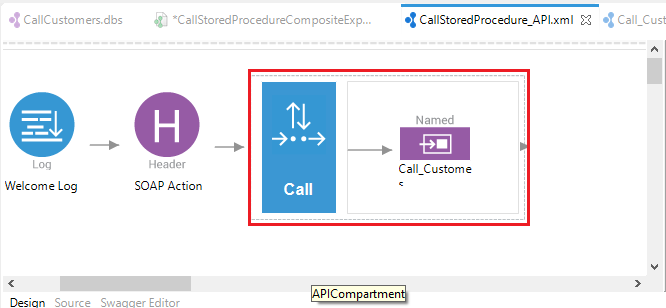
# Define End Point :

* Right click on endpoints and click to create new endpoint.
* Create a new Address End Point.
* Set :
* Format = SOAP 1.1
* Paste the URI.

Graphical user interface, application, email

Description automatically generated

* Drag the Call Mediator and inside the mediator place the defined endpoint.

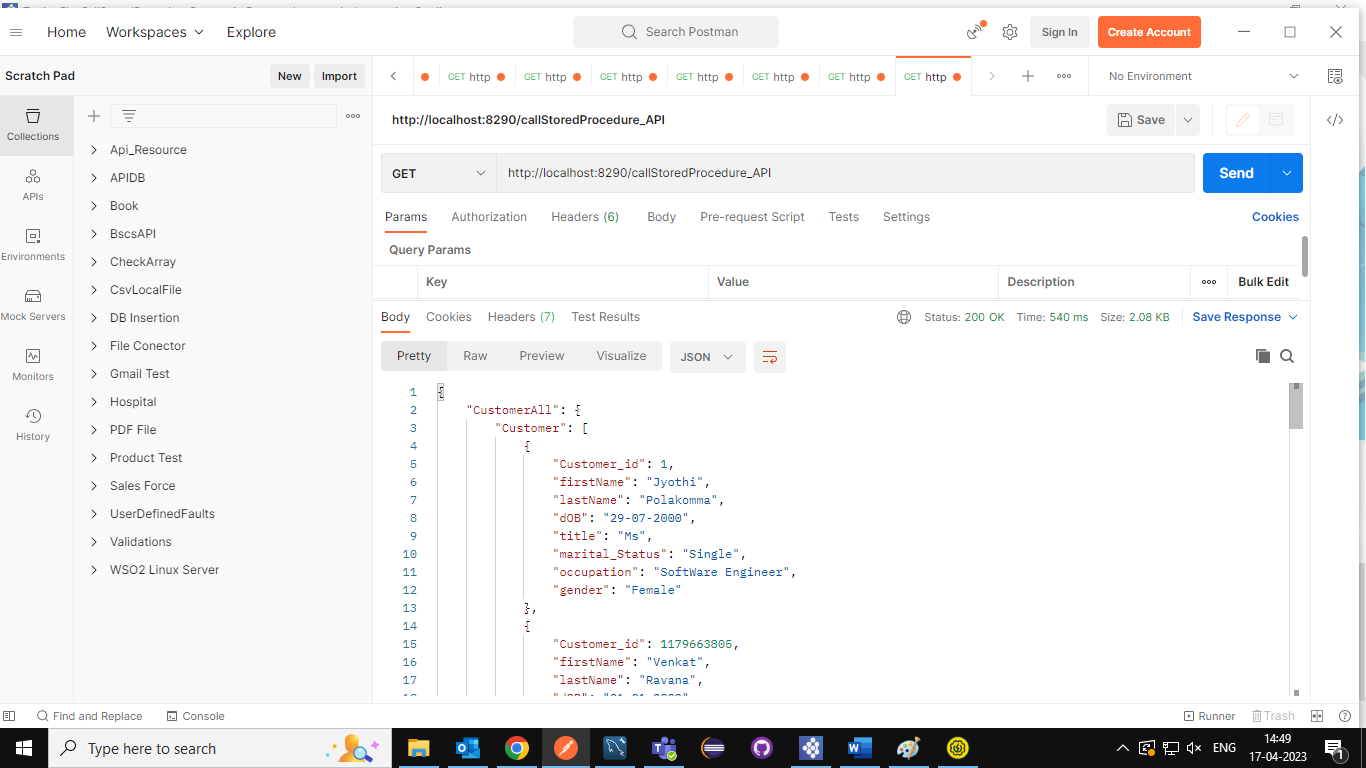


* The Result for the Data Service is in the XML Format.
* If you want to change the response in JSON format add a Property Mediator set Name = messageType and value = application/json.
* Add the Respond mediator to get the Response.

Graphical user interface, application

Description automatically generated

* Export the Artifacts and run the Server.
* Paste the URL in Post-Man and check the response.



# Sample Data Service Code:

<data transports=*"http https"* serviceGroup=*""* serviceNamespace=*""* name=*"CallCustomers"*>

<description />

<config id=*"Customer"*>

<property name=*"driverClassName"*>com.mysql.cj.jdbc.Driver</property>

<property name=*"url"*>jdbc:mysql://localhost:3306/sales</property>

<property name=*"username"*>root</property>

<property name=*"password"*>mysql</property>

</config>

<query id=*"getCustomers"* useConfig=*"Customer"*>

<sql>call customers()</sql>

<result element=*"CustomerAll"* rowName=*"Customer"*>

<element name=*"Customer\_id"* column=*"Customer\_id"* xsdType=*"string"* />

<element name=*"firstName"* column=*"firstName"* xsdType=*"string"* />

<element name=*"lastName"* column=*"lastName"* xsdType=*"string"* />

<element name=*"dOB"* column=*"dOB"* xsdType=*"string"* />

<element name=*"title"* column=*"title"* xsdType=*"string"* />

<element name=*"marital\_Status"* column=*"marital\_Status"* xsdType=*"string"* />

<element name=*"occupation"* column=*"occupation"* xsdType=*"string"* />

<element name=*"gender"* column=*"gender"* xsdType=*"string"* />

</result>

</query>

<resource method=*"GET"* path=*"getCustomerDetails"*>

<description />

<call-query href=*"getCustomers"* />

</resource>

</data>

# Sample REST API Source Code :

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<api context=*"/callStoredProcedure\_API"* name=*"CallStoredProcedure\_API"* xmlns=*"http://ws.apache.org/ns/synapse"*>

<resource methods=*"GET"*>

<inSequence>

<log description=*"Welcome Log "* level=*"full"*>

<property name=*"API Name =>"* value=*"CallStoredProcedure\_API"*/>

</log>

<header description=*"SOAP Action"* name=*"Action"* scope=*"default"* value=*"\_getgetcustomerdetails"*/>

<call description=*"Calling the Data Service"*>

<endpoint key=*"Call\_Customers"*/>

</call>

<property description=*"Change the Message Type"* name=*"messageType"* scope=*"axis2"* type=*"STRING"* value=*"application/json"*/>

<respond/>

</inSequence>

<outSequence/>

<faultSequence/>

</resource>

</api>