# [Exposing a Datasource as a Data Service](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service)

In this tutorial, we will run through the process of service enabling an **RDBMS** as a data service.  For more information on the datasource types supported by WSO2 EI, see the [What's Next](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-What'sNext) section.

**Before you begin!**

Follow the steps given below to set up a MySQL database for this tutorial.

1. Install the MySQL server.
2. Download the product installer from [here](http://wso2.com/integration/), and run the installer.  
   Let's call the installation location of your product the **<EI\_HOME>** directory. This is located in a place specific to your OS as shown below:

| **OS** | **Home directory** |
| --- | --- |
| Mac OS | /Library/WSO2/EnterpriseIntegrator/6.6.0 |
| Windows | C:\Program Files\WSO2\EnterpriseIntegrator\6.6.0\ |
| Ubuntu | /usr/lib/wso2/EnterpriseIntegrator/6.6.0 |
| CentOS | /usr/lib64/EnterpriseIntegrator/6.6.0 |

1. Download the JDBC driver for MySQL from [here](http://dev.mysql.com/downloads/connector/j/). Unzip it, get the <MySQL\_HOME>/mysql-connector-java-8.0.16.jar JAR, and place it in the <EI\_HOME>/lib directory.

If the driver class does not exist in the relevant folders when you create the datasource, you will get an exception, such as 'Cannot load JDBC driver class com.mysql.jdbc.Driver'.

1. Create a database named Employees.

|  |
| --- |
| CREATE DATABASE Employees; |

1. Create the Employee table inside the Employees database:

|  |
| --- |
| USE Employees;    CREATE TABLE Employees (EmployeeNumber **int**(11) NOT NULL, FirstName varchar(255) NOT NULL, LastName varchar(255) DEFAULT NULL, Email varchar(255) DEFAULT NULL, Salary varchar(255)); |

Let's get started

* [Creating the data service](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-Creatingthedataservice)
  + [Connecting to the datasource](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-Connectingtothedatasource)
  + [Creating a query to Get data](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-CreatingaquerytoGetdata)
  + [Creating a query to Post data](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-CreatingaquerytoPostdata)
  + [Creating a query to Update data](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-CreatingaquerytoUpdatedata)
  + [Create SOAP operations to invoke queries](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-CreateSOAPoperationstoinvokequeries)
  + [Create REST resources to invoke queries](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-CreateRESTresourcestoinvokequeries)
* [Invoking your data service using SOAP](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-InvokingyourdataserviceusingSOAP)
  + [Post new data](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-Postnewdata)
  + [Get data](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-Getdata)
  + [Update data](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-Updatedata)
* [Invoking your data service using REST](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-InvokingyourdataserviceusingREST)
  + [Post new data](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-Postnewdata.1)
  + [Get data](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-Getdata.1)
  + [Update data](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-Updatedata.1)

**Creating the data service**

Follow the steps given below.

1. Start the WSO2 ESB profile.
   * [On MacOS/Linux/CentOS](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ba002576187447b8ae425b539a0f6095)
   * [On Windows](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#f15bcd27a1b64c7686e862428e7ca62d)

Open a terminal and execute the following command:

|  |
| --- |
| sudo wso2ei-6.6.0-integrator |

1. Open the ESB profile's Management Console using <https://localhost:9443/carbon>, and log in using admin as the username and the password.
2. Click **Data Service → Create**, to start creating a data service.
3. Enter the following name for the data service.

|  |  |
| --- | --- |
| **Data Service Name** | RDBMSDataService |

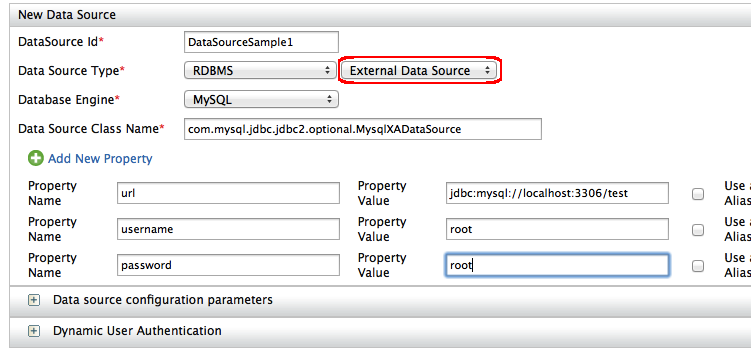
1. Click **Next** to enter the datasource connection details.

**Connecting to the datasource**

Follow the steps given below.

1. Click **Add New Datasource** and enter the following details:

|  |  |
| --- | --- |
| **Datasource ID** | Datasource |
| **Datasource Type** | RDBMS |
| **Datasource Type (Default/External)** | Leave **Default** selected. |
| **Database Engine** | MySQL |
| **Driver Class** | com.mysql.jdbc.Driver |
| **URL** | jdbc:mysql://localhost:3306/Employees |
| **User Name** | Enter your MySQL server's username. |
| **Password** | Enter your MySQL server's password. If you have not assigned a password, keep this field empty. |

1. If you enter **External** instead of the Default datasource type, your datasource should be supported by an external provider class, such as  com.mysql.jdbc.jdbc2.optional.MysqlXADataSource. You can select the **External** option and enter the name and value of connection properties by clicking  **Add Property**. For example,  
     
   After an external datasource is created, it can be used as a usual datasource in queries. See the tutorial on [handling distributed transactions](https://docs.wso2.com/display/EI660/Handling+Distributed+Transactions) for more information on using external datasources.
2. Save the datasource.
3. Click **Next**, to start creating queries.

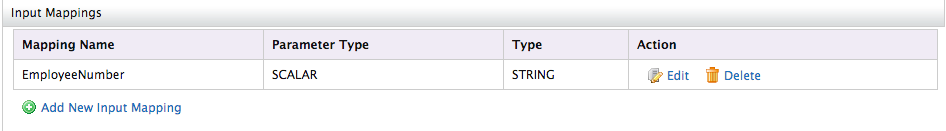
**Creating a query to Get data**

Let's create a query that can retrieve employee data, based on the employee number. That is, when the employee number is provided as an input, the data service should get the relevant employee details and present the result.

1. Click **Add New Query** to start creating a new query.
2. Enter the following details:

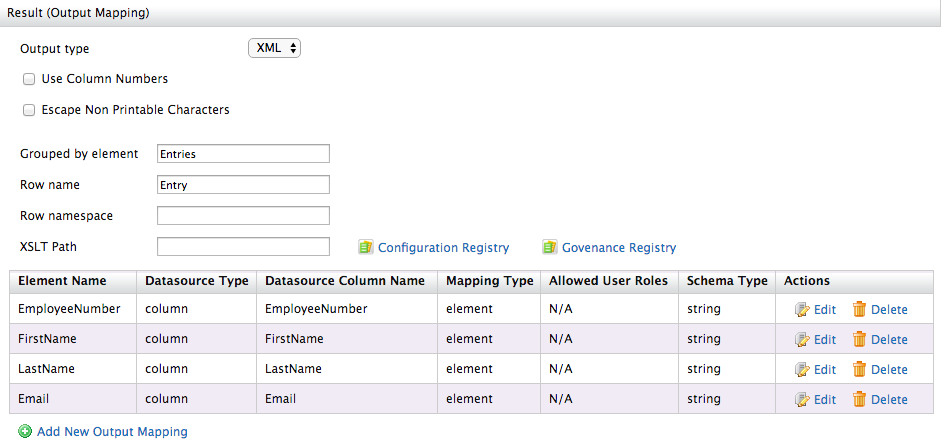
|  |  |
| --- | --- |
| **Query ID** | GetEmployeeDetails |
| **Datasource** | Datasource |
| **SQL** | |  | | --- | | select EmployeeNumber, FirstName, LastName, Email, Salary from Employees where EmployeeNumber=:EmployeeNumber | |

1. Click **Generate Input Mapping** to create the input mapping. The employee number is the input as shown below.



1. Click **Generate Response** to create the output mapping. This defines how the employee details retrieved from the datasource will be presented in the result. Note that, by default, the output type is XML.

If required, you can choose RDF, or JSON as the output type. See [Using JSON with Data Services](https://docs.wso2.com/display/EI660/Using+JSON+with+Data+Services) for more information on exposing data in JSON format.



1. Save the query.

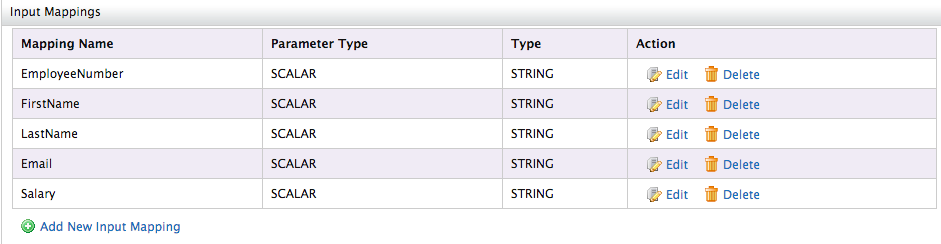
**Creating a query to Post data**

Let's create another query to post new employee data to the datasource.

1. Click **Add New Query** to start creating a new query.
2. Enter the following details:

|  |  |
| --- | --- |
| **Query ID** | AddEmployeeDetails |
| **Datasource** | Datasource |
| **SQL** | |  | | --- | | insert into Employees (EmployeeNumber, FirstName, LastName, Email, Salary) values(:EmployeeNumber,:FirstName,:LastName,:Email,:Salary) | |

1. Click **Generate Input Mapping** to create the input mapping. You need to specify values for the following elements when posting new employee data.



1. Save the query.

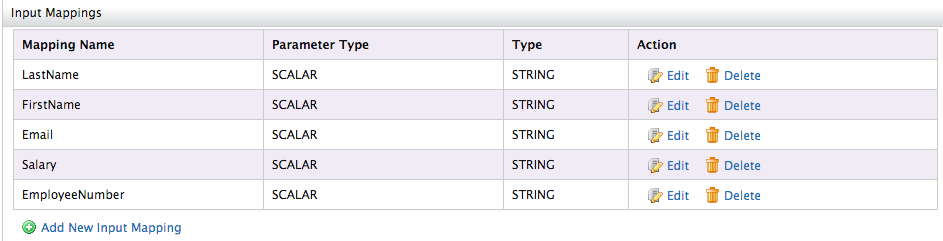
**Creating a query to Update data**

Now, let's create a query that can update an existing employee record in the datasource.

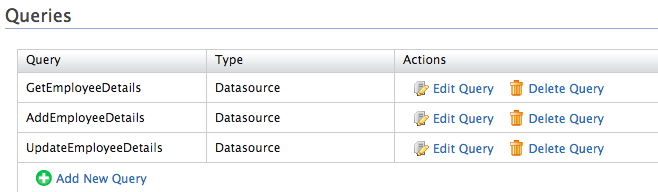
1. Click **Add New Query** to start creating a new query.
2. Enter the following details:

|  |  |
| --- | --- |
| **Query ID** | UpdateEmployeeDetails |
| **Datasource** | Datasource |
| **SQL** | |  | | --- | | update Employees set LastName=:LastName, FirstName=:FirstName, Email=:Email, Salary=:Salary where EmployeeNumber=:EmployeeNumber | |

1. Click **Generate Input Mapping** to create the input mapping. You need to specify values for the following elements when posting new employee data.



1. Save the query.

You should now have the following three queries created:  


       5. Click **Next**, to add Operations.

**Create SOAP operations to invoke queries**

Now, let's create SOAP operations to invoke the queries created above. Alternatively, you can create REST resources to invoke the queries. See the [next section](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-CreateRESTresourcestoinvokequeries) for instructions.

1. Click **Add New Operation** and enter the details as shown below.

|  |  |
| --- | --- |
| **Operation Name** | GetEmployeeOp |
| **Query ID** | GetEmployeeDetails |

1. Save the operation.
2. Click **Add New Operation** and enter the details as shown below.

|  |  |
| --- | --- |
| **Operation Name** | AddEmployeeOp |
| **Query ID** | AddEmployeeDetails |

1. Save the operation.
2. Click **Add New Operation** and enter the details as shown below.

|  |  |
| --- | --- |
| **Operation Name** | UpdateEmployeeOp |
| **Query ID** | UpdateEmployeeDetails |

1. Save the operation.

You can now invoke the data service query using SOAP.

**Create REST resources to invoke queries**

Now, let's create REST resources to invoke the queries created above. Alternatively, you can create SOAP operations to invoke the queries. See the [previous section](https://docs.wso2.com/display/EI660/Exposing+a+Datasource+as+a+Data+Service#ExposingaDatasourceasaDataService-CreateSOAPoperationstoinvokequeries), for instructions.

1. Click **Add New Resource** and enter the details as shown below.

|  |  |
| --- | --- |
| **Resource Path** | Employee/{EmployeeNumber} |
| **Resource Method** | GET |
| **Query ID** | GetEmployeeDetails |

1. Save the resource.
2. Click **Add New Resource** and enter the details as shown below.

|  |  |
| --- | --- |
| **Resource Path** | Employee |
| **Resource Method** | POST |
| **Query ID** | AddEmployeeDetails |

1. Save the resource.
2. Click **Add New Resource** and enter the details as shown below.

|  |  |
| --- | --- |
| **Resource Path** | Employee |
| **Resource Method** | PUT |
| **Query ID** | UpdateEmployeeDetails |

1. Save the resource.
2. Click **Finish** to complete creating the data service.

Once you have defined the operation, click **Finish** to complete the data service creation process. You will now be taken to the **Deployed Services** screen, which shows all the data services deployed on the server.  
You can now invoke the data service query using REST.

**Invoking your data service using SOAP**

You can try the data service you created by using the TryIt tool that is in your product by default.

1. Go to the **Deployed Services** screen and refresh the page.
2. Click the**Try this service** link for the **RDBMS**data service. The **TryIt** Tool will open with the data service.

**Post new data**

1. Select the **AddEmployeeOp** operation you created earlier.
2. Provide the employee details by copying the code given below.

|  |
| --- |
| <p:AddEmployeeOp xmlns:p="[http://ws.wso2.org/dataservice"](http://ws.wso2.org/dataservice)>        <!--Exactly 1 occurrence-->        <xs:EmployeeNumber xmlns:xs="[http://ws.wso2.org/dataservice"](http://ws.wso2.org/dataservice)>1</xs:EmployeeNumber>        <!--Exactly 1 occurrence-->        <xs:FirstName xmlns:xs="[http://ws.wso2.org/dataservice"](http://ws.wso2.org/dataservice)>John</xs:FirstName>        <!--Exactly 1 occurrence-->        <xs:LastName xmlns:xs="[http://ws.wso2.org/dataservice"](http://ws.wso2.org/dataservice)>Doe</xs:LastName>        <!--Exactly 1 occurrence-->        <xs:Email xmlns:xs="[http://ws.wso2.org/dataservice"](http://ws.wso2.org/dataservice)>JohnDoe@gmail.com</xs:Email>        <!--Exactly 1 occurrence-->        <xs:Salary xmlns:xs="[http://ws.wso2.org/dataservice"](http://ws.wso2.org/dataservice)>10000</xs:Salary>     </p:AddEmployeeOp> |

1. Click **Send**.

The data is now added to the database.

**Get data**

1. Select the GetEmployeeOp operation you created earlier. You need to provide the employee number as an input. Enter '1'.
2. Click **Send**to see the details of the employee you added previously:

|  |
| --- |
| <Entries xmlns="[http://ws.wso2.org/dataservice"](http://ws.wso2.org/dataservice)>     <Entry>        <EmployeeNumber>1</EmployeeNumber>        <FirstName>John</FirstName>        <LastName>Doe</LastName>        <Email>JohnDoe@gmail.com</Email>     </Entry>  </Entries> |

**Update data**

1. Select the **UpdateEmployeeOp** operation you created earlier.
2. Update the employee details by copying the code given below. Note that the salary value is changed to 20000.

|  |
| --- |
| <p:UpdateEmployeeOp xmlns:p="[http://ws.wso2.org/dataservice"](http://ws.wso2.org/dataservice)>        <!--Exactly 1 occurrence-->        <xs:LastName xmlns:xs="[http://ws.wso2.org/dataservice"](http://ws.wso2.org/dataservice)>Doe</xs:LastName>        <!--Exactly 1 occurrence-->        <xs:FirstName xmlns:xs="[http://ws.wso2.org/dataservice"](http://ws.wso2.org/dataservice)>John</xs:FirstName>        <!--Exactly 1 occurrence-->        <xs:Email xmlns:xs="[http://ws.wso2.org/dataservice"](http://ws.wso2.org/dataservice)>JohnDoe@gmail.com</xs:Email>        <!--Exactly 1 occurrence-->        <xs:Salary xmlns:xs="[http://ws.wso2.org/dataservice"](http://ws.wso2.org/dataservice)>20000</xs:Salary>        <!--Exactly 1 occurrence-->        <xs:EmployeeNumber xmlns:xs="[http://ws.wso2.org/dataservice"](http://ws.wso2.org/dataservice)>1</xs:EmployeeNumber>  </p:UpdateEmployeeOp> |

1. Click **Send**.

The data is now updated in the database.

**Invoking your data service using REST**

Let's take a look at the curl commands that are used to send the HTTP requests for each of the resources:

**Post new data**

1. Create a file called employee-payload.xml file, and define the XML payload for posting new data as shown below.

|  |
| --- |
| <\_postemployee>      <EmployeeNumber>3</EmployeeNumber>      <FirstName>Will</FirstName>      <LastName>Smith</LastName>      <Email>will@google.com</Email>      <Salary>15500.0</Salary>  </\_postemployee> |

1. Send the following HTTP request from the location where the employee-payload.xml file is stored:

|  |
| --- |
| curl -X POST -H 'Accept: application/xml'  -H 'Content-Type: application/xml' --data "@employee-payload.xml" http://localhost:8280/services/RDBMSDataService/employee |

**Get data**

The service can be invoked in REST-style via curl ([http://curl.haxx.se](http://curl.haxx.se/)). Shown below is the curl command to invoke the GET resource:

|  |
| --- |
| curl -X GET http://localhost:8280/services/RDBMSDataService.HTTPEndpoint/Employee/3 |

This generates a response as follows.

|  |
| --- |
| <Entries xmlns="[http://ws.wso2.org/dataservice"](http://ws.wso2.org/dataservice)><Entry><EmployeeNumber>3</EmployeeNumber><FirstName>Will</FirstName><LastName>Smith</LastName><Email>will@google.com</Email><Salary>15500.0</Salary></Entry><Entry><EmployeeNumber>3</EmployeeNumber><FirstName>Will</FirstName><LastName>Smith</LastName><Email>will@google.com</Email><Salary>15500.0</Salary></Entry><Entry><EmployeeNumber>3</EmployeeNumber><FirstName>Will</FirstName><LastName>Smith</LastName><Email>will@google.com</Email><Salary>15500.0</Salary></Entry></Entries> |

**Update data**

1. Create a file called employee-update-payload.xml file, and define the XML payload for updating an existing employee record as shown below.

|  |
| --- |
| <\_putemployee>      <EmployeeNumber>3</EmployeeNumber>      <LastName>Smith</LastName>      <FirstName>Will</FirstName>      <Email>will@google.com</Email>      <Salary>30000.0</Salary>  </\_putemployee> |

1. Send the following HTTP request from the location where the employee-update-payload.xml file is stored:

|  |
| --- |
| curl -X PUT -H 'Accept: application/xml'  -H 'Content-Type: application/xml' --data "@employee-update-payload.xml" http://localhost:8280/services/RDBMSDataService/employee |

What's Next

* Want to try out the datasource types supported by WSO2 EI? See the sections given below:
  + [Exposing a Google Spreadsheet as a Data Service](https://docs.wso2.com/pages/viewpage.action?pageId=85372020)
  + [Exposing CSV Data as a Data Service](https://docs.wso2.com/pages/viewpage.action?pageId=85372051)
  + [Exposing Excel Data as a Data Service](https://docs.wso2.com/pages/viewpage.action?pageId=85372066)
  + [Exposing RDF Data as a Data Service](https://docs.wso2.com/pages/viewpage.action?pageId=85372075)
  + [Exposing MongoDB as a Data Service](https://docs.wso2.com/pages/viewpage.action?pageId=85372088)
  + [Exposing a Web Resource as a Data Service](https://docs.wso2.com/pages/viewpage.action?pageId=85372095)
  + [Exposing a Carbon Datasource as a Data Service](https://docs.wso2.com/pages/viewpage.action?pageId=85372101)
  + [Exposing a Custom Datasource as a Data Service](https://docs.wso2.com/pages/viewpage.action?pageId=85372103)
  + [Exposing Cassandra as a Data Service](https://docs.wso2.com/pages/viewpage.action?pageId=85372119)
  + [Exposing a JNDI Datasource as a Data Service](https://docs.wso2.com/pages/viewpage.action?pageId=85372120)
* Try out the [Data Integration Samples](https://docs.wso2.com/pages/viewpage.action?pageId=85369285).