**DSS-Data Service Server**

The **Data Service Call Mediator** is used to invoke data service operations. It automatically creates a payload and sets up the necessary headers to invoke the data service. Also, it improves the performance by directly calling the data service (without HTTP transport).

# [Creating a Datasource](DSS-DataServiceServer.docx)

Follow the instructions given below to create a new Datasource connection in WSO2 Integration Studio.

## Instructions

Follow the steps given below to create the datasource file:

1. Select the already created [**Datasource Config module**](https://apim.docs.wso2.com/en/4.2.0/integrate/develop/create-integration-project/#datasource-project) in the project navigator, right-click, and go to **New -> Datasource**.

A screenshot of a computer

Description automatically generated with medium confidence

The **New Datasource** window will open as shown below.

A screenshot of a computer

Description automatically generated with medium confidence

1. Select your [**datasource config module**](https://apim.docs.wso2.com/en/4.2.0/integrate/develop/create-integration-project/#datasource-project) as the **Container**, add the file name for your datasource, and click **Finish**.

A datasource file will now be created in your datasource config module. Shown below is the sample configuration that is created. You can now update the values in this configuration.

<**datasource**>

<**name**>MySQLConnection</**name**>

<**description**>MySQL Connection</**description**>

<**jndiConfig** useDataSourceFactory="false">

<**name**>MysqlConJNDI1</**name**>

</**jndiConfig**>

<**definition** type="RDBMS">

<**configuration**>

<**driverClassName**>com.mysql.jdbc.Driver</**driverClassName**>

<**url**>jdbc:mysql://localhost:3306/mysqldb</**url**>

<**username**>username</**username**>

<**password**>password</**password**>

</**configuration**>

</**definition**>

</**datasource**>

# [Creating a Data Service](DSS-DataServiceServer.docx)

Follow the instructions given below to create a new data service artifact.

**Tip**

You can also use a sample template to create your data service.

1. Open the **Getting Started** view of WSO2 Integration Studio (**Menu -> Help -> Getting Started**).
2. In the Getting Started view, go to the **Data Service** tab and select the **REST Data Service** example.

## Instructions[**¶**](https://apim.docs.wso2.com/en/latest/integrate/develop/creating-artifacts/data-services/creating-data-services/#instructions)

## Create the data service artifact[**¶**](https://apim.docs.wso2.com/en/latest/integrate/develop/creating-artifacts/data-services/creating-data-services/#create-the-data-service-artifact)

Follow the steps given below to create the data service file:

1. Right-click the **Data Service Config** module in the project explorer and go to **New -> Data Service**.

A screenshot of a computer

Description automatically generated with medium confidence

1. In the **New Data Service** wizard that opens, select **Create New Data Service** and click **Next**.

[A screenshot of a computer

Description automatically generated with medium confidence](https://apim.docs.wso2.com/en/4.2.0/assets/img/integrate/tutorials/data_services/119130577/119130578.png)

1. Enter a name for the data service and click **Finish**.

A data service file (DBS file) will now be created in your data service project as show below.

A screenshot of a computer

Description automatically generated with medium confidence

### **Adding a datasource**[**¶**](https://apim.docs.wso2.com/en/latest/integrate/develop/creating-artifacts/data-services/creating-data-services/#adding-a-datasource)

You can configure the datasource connection details using this section.

1. Click **Data Sources** to expand the section.

A screenshot of a computer

Description automatically generated with medium confidence

1. Click **Add New** to open the **Create Datasource** page.

A screenshot of a computer

Description automatically generated with medium confidence

1. Enter the datasource connection details.
2. Click **Test Connection** to expand the section.

A screenshot of a computer

Description automatically generated with medium confidence

1. Click the **Test Connection** button to verify the connectivity between the MySQL datasource and the data service.
2. Save the data service.

### **Creating a query**[**¶**](https://apim.docs.wso2.com/en/latest/integrate/develop/creating-artifacts/data-services/creating-data-services/#creating-a-query)

You can configure the main query details using this section.

1. Click **Queries** to expand the section.

[A close-up of a message

Description automatically generated with low confidence](https://apim.docs.wso2.com/en/4.2.0/assets/img/integrate/tutorials/data_services/query_expanded.png)

1. Click **Add New** to open the **Add Query** page.

[A screenshot of a computer

Description automatically generated with low confidence](https://apim.docs.wso2.com/en/4.2.0/assets/img/integrate/tutorials/data_services/add_query.png)

1. Enter the following query details.

|  |  |
| --- | --- |
| Parameter | Description |
| Query ID | Give a unique name to Identify the Query. |
| Datasource | All the datasources created for this data service are listed. Select the required datasource from the list. |
| SQL Query | You can enter the SQL query in this text box. |

#### Input mapping[¶](https://apim.docs.wso2.com/en/latest/integrate/develop/creating-artifacts/data-services/creating-data-services/#input-mapping)

You can configure input parameters for the query using this section.

1. Click **Input Mappings** to expand the section.

[A picture containing text, font, screenshot

Description automatically generated](https://apim.docs.wso2.com/en/4.2.0/assets/img/integrate/tutorials/data_services/input_mapping_expanded.png)

1. There are two ways to create the mapping:
   * You can click **Generate** to automatically generate the input mappings from the SQL query.
   * If you want to add a new input mapping:
     1. Click **Add New** to open the **Add Input Mapping** page.

[A screenshot of a computer

Description automatically generated with medium confidence](https://apim.docs.wso2.com/en/4.2.0/assets/img/integrate/tutorials/data_services/add_input_mappings.png)

* + 1. Enter the following input mapping details:

|  |  |
| --- | --- |
| Parameter | Description |
| Mapping Name | Give a name for the mapping. |
| Parameter Type | The parameter type. |
| SQL Type | The SQL type. |

* + 1. Save the input mapping.

Shown below is an example query with input mapping:

[A screenshot of a computer

Description automatically generated with medium confidence](https://apim.docs.wso2.com/en/4.2.0/assets/img/integrate/tutorials/data_services/input_mappings.png)

#### Result (Output Mappings)[¶](https://apim.docs.wso2.com/en/latest/integrate/develop/creating-artifacts/data-services/creating-data-services/#result-output-mappings)

You can configure output result parameters for the query using this section.

1. Click **Result (Output Mappings)** to expand the section.

[A screenshot of a computer

Description automatically generated](https://apim.docs.wso2.com/en/4.2.0/assets/img/integrate/tutorials/data_services/out_mapping_expanded.png)

1. Enter the following details:

|  |  |
| --- | --- |
| Property | Description |
| Grouped by Element | Employees |

1. There are two ways to create the output mapping:
   * You can click **Generate** to automatically generate the output mappings from the SQL query.
   * Alternatively, you can manually add the mappings:
     1. Click **Add New** to open the **Add Output Mapping** page.

[A screenshot of a computer

Description automatically generated with medium confidence](https://apim.docs.wso2.com/en/4.2.0/assets/img/integrate/tutorials/data_services/add_output_mappings.png)

* + 1. Enter the following output element details.

|  |  |
| --- | --- |
| Property | Description |
| Datasource Type | column |
| Output Field Name | EmployeeNumber |
| Datasource Column Name | EmployeeNumber |
| Schema Type | String |

* + 1. Save the element.
    2. Follow the same steps to create the remaining output elements.

Shown below is an example query with output mappings:

A screenshot of a computer

Description automatically generated with medium confidence

#### Advanced properties[¶](https://apim.docs.wso2.com/en/latest/integrate/develop/creating-artifacts/data-services/creating-data-services/#advanced-properties)

Click **Advanced Properties** to expand the section and add the required parameter values.

A screenshot of a computer

Description automatically generated with medium confidence

The data service should now have the query element added.

### **Adding a SOAP operation**[**¶**](https://apim.docs.wso2.com/en/latest/integrate/develop/creating-artifacts/data-services/creating-data-services/#adding-a-soap-operation)

Use this section to configure a SOAP operation for invoking the data service.

1. Click **Operations** to expand the section.

[A close-up of a message

Description automatically generated with low confidence](https://apim.docs.wso2.com/en/4.2.0/assets/img/integrate/tutorials/data_services/new-operataion.png)

1. Click **Add New** to add a SOAP Operation for your data service.

[A screenshot of a computer

Description automatically generated with medium confidence](https://apim.docs.wso2.com/en/4.2.0/assets/img/integrate/tutorials/data_services/add-operation.png)

1. Enter the following information:

|  |  |
| --- | --- |
| Parameter | Description |
| Operation Name | Give a name to the SOAP Operation. |
| Query ID | Select the Query from the listed queries. |
| Operation Parameters | Click **Add New** to add new parameters to the operation. |

### **Adding a Resource**[**¶**](https://apim.docs.wso2.com/en/latest/integrate/develop/creating-artifacts/data-services/creating-data-services/#adding-a-resource)

Use this section to configure a REST resource for invoking the data service.

1. Click **Resources** to expand the section.

[A close-up of a message

Description automatically generated with low confidence](https://apim.docs.wso2.com/en/4.2.0/assets/img/integrate/tutorials/data_services/resource_expanded.png)

1. Click **Add New** to add a new resource.

[A screenshot of a computer

Description automatically generated with medium confidence](https://apim.docs.wso2.com/en/4.2.0/assets/img/integrate/tutorials/data_services/create_resource.png)

1. Give the following details to create the REST resource.

|  |  |
| --- | --- |
| Parameter | Description |
| Resource Path | Give the HTTP REST resource path you need to expose. |
| Query ID | Select the Query ID from the drop down list that you need to expose as a REST resource. |

1. Click **Save** to add the resource to the data service.

The data service should now have the resource added.

### **Generate Data Service from a Datasource**[**¶**](https://apim.docs.wso2.com/en/latest/integrate/develop/creating-artifacts/data-services/creating-data-services/#generate-data-service-from-a-datasource)

Follow the steps given below to automatically create a data service using a given datasource structure. When generating a data service, the server takes its table structure according to the structure specified in the datasource and automatically creates the SELECT, INSERT, UPDATE, and DELETE operations.

1. Create a datasource project and add a datasource in the current workspace. You can refer [Creating a Datasource](https://apim.docs.wso2.com/en/4.2.0/integrate/develop/creating-artifacts/data-services/creating-datasources) for more information.
2. In the **New Data Service** wizard that opens, select **Generate Data Service from Datasource** and click **Next**. [A screenshot of a computer

   Description automatically generated](https://apim.docs.wso2.com/en/4.2.0/assets/img/integrate/tutorials/data_services/generate_dataservice.png)
3. From the wizard, select the datasource that you have configured in step 1.

[A screenshot of a computer

Description automatically generated](https://apim.docs.wso2.com/en/4.2.0/assets/img/integrate/tutorials/data_services/generate_dataservice_select_datasource.png)

1. Select the driver to connect to the datasource. You need to browse and upload a driver from your file system.

[A screenshot of a computer

Description automatically generated](https://apim.docs.wso2.com/en/4.2.0/assets/img/integrate/tutorials/data_services/select_driver_file_system.png)

Then click **Fetch Table** to list down all avaialble tables in the selected datasource.

1. From the list of tables, select the tables and the REST resource methods that you want in the generated data service.

**Note**

* 1. The **POST** REST method is enabled only when the database is not in read-only mode.
  2. The **PUT** and **DELETE** REST methods are enabled only when a primary key is defined on the table.

[A screenshot of a computer

Description automatically generated](https://apim.docs.wso2.com/en/4.2.0/assets/img/integrate/tutorials/data_services/select_tables.png)

1. You can select a service generation mode from the following two options:
   1. Single Service: Creates a single data service for resources of all tables. If this option is selected, you need to provide a name for the Data Service you are creating.
   2. Multiple Services: Creates a service per table, which will contain isolated resources for each table.
2. Click **Finish** to generate the services and add to the dataservices project.

# [Data Service Call Mediator](DSS-DataServiceServer.docx)

**Info**

* You need to first have a [Data Service Project](https://apim.docs.wso2.com/en/4.2.0/integrate/develop/creating-artifacts/data-services/creating-data-services) to use the Data Service Call mediator.
* The Data Service Call mediator is a [content-aware](https://apim.docs.wso2.com/en/4.2.0/reference/mediators/about-mediators/#classification-of-mediators) mediator.

## **Syntax**[**¶**](https://apim.docs.wso2.com/en/latest/reference/mediators/dss-mediator/#syntax)

<dataServiceCall serviceName="data-service-name">

<source [type="inline" | "body"]/>

<operations [type="single" | "batch" | "request-box"] >

<operation name="operation-name">

<param name="param-name" value="param-value"/>

</operation>

</operations>

<target [type="body" | "property"] name="target-property-name"/>

</dataServiceCall>

## **Configuration**[**¶**](https://apim.docs.wso2.com/en/latest/reference/mediators/dss-mediator/#configuration)

The **Source Configuration** properties of the Data Service Call Mediator are as follows:

| Parameter Name | Description |
| --- | --- |
| **Type** | The type defines the source for the payload that is required for the data service call. By default, the source type is set to ‘body’. The available values are as follows:   * **INLINE** - The payload should be configured within the mediator configuration. * **BODY** - The body of the original message is passed as the payload to the data service. |

The **Operation Configurations** for the Data Source Call mediator are as follows:

| Parameter Name | Description |
| --- | --- |
| **name** | Defines the name of the operation that is to be invoked |
| **Params Configuration** | The possible values for this parameter are as follows:   * **Name**: Defines the name of the parameter. * **Evaluator**: Only required for json param expressions (json). * **Value/Expression**: Value of the parameter. If the expression is configured, the parameter value is determined during message mediation by evaluating an expression. The expression should be specified for the Expression parameter. |

The **Target Configuration** properties of the Data Service Call mediator are as follows:

| Parameter Name | Description |
| --- | --- |
| **Type** | By setting the target type, the response payload of the data service call can be stored in the body or a property. By default, the target type is set to ‘body’. The available values are as follows:   * **BODY**: The response payload is stored in the message body. * **PROPERTY**: The response payload is stored in the defined property. |
| **Name** | Specifies the property name. You can define dynamic property names when the target type is defined as a property:  <property name="{get-property('propertyName')}" />  <property name="{$ctx:propertyName}" />  <property name="{json-eval(propertyName)}" /> |

## **Examples**[**¶**](https://apim.docs.wso2.com/en/latest/reference/mediators/dss-mediator/#examples)

Use the following datasource to try out the Data Service Call mediator. Create a new data service configuration and then copy the following content to define the DSSCallMediatorTest data service:

**Sample data service to invoke using the Data Service Call mediator**

<**data** disableLegacyBoxcarringMode="true" enableBatchRequests="true" enableBoxcarring="true" name="DSSCallMediatorTest" transports="http https local">

<**config** enableOData="false" id="01">

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

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

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

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

</**config**>

<**query** id="getEmployeeByNumberQuery" useConfig="01">

<**sql**>select EmployeeNumber, FirstName, LastName, Email, Salary from Employees where EmployeeNumber=:EmployeeNumber</**sql**>

<**result** element="Entries" rowName="Entry">

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

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

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

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

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

</**result**>

<**param** name="EmployeeNumber" sqlType="STRING"/>

</**query**>

<**query** id="addEmployeeQuery" useConfig="01">

<**sql**>insert into Employees (EmployeeNumber, FirstName, LastName, Email, Salary) values(:EmployeeNumber,:FirstName,:LastName,:Email,:Salary)</**sql**>

<**param** name="EmployeeNumber" sqlType="STRING"/>

<**param** name="FirstName" sqlType="STRING"/>

<**param** name="LastName" sqlType="STRING"/>

<**param** name="Email" sqlType="STRING"/>

<**param** name="Salary" sqlType="STRING"/>

</**query**>

<**operation** name="addEmployee" returnRequestStatus="true">

<**call-query** href="addEmployeeQuery">

<**with-param** name="EmployeeNumber" query-param="EmployeeNumber"/>

<**with-param** name="FirstName" query-param="FirstName"/>

<**with-param** name="LastName" query-param="LastName"/>

<**with-param** name="Email" query-param="Email"/>

<**with-param** name="Salary" query-param="Salary"/>

</**call-query**>

</**operation**>

<**operation** name="getEmployeeByNumber">

<**call-query** href="getEmployeeByNumberQuery">

<**with-param** name="EmployeeNumber" query-param="EmployeeNumber"/>

</**call-query**>

</**operation**>

<**resource** method="GET" path="Employee/{EmployeeNumber}">

<**call-query** href="getEmployeeByNumberQuery">

<**with-param** name="EmployeeNumber" query-param="EmployeeNumber"/>

</**call-query**>

</**resource**>

<**resource** method="POST" path="/Employee">

<**call-query** href="addEmployeeQuery">

<**with-param** name="EmployeeNumber" query-param="EmployeeNumber"/>

<**with-param** name="FirstName" query-param="FirstName"/>

<**with-param** name="LastName" query-param="LastName"/>

<**with-param** name="Email" query-param="Email"/>

<**with-param** name="Salary" query-param="Salary"/>

</**call-query**>

</**resource**>

</**data**>

### **Example 1: Inline single request operation**[**¶**](https://apim.docs.wso2.com/en/latest/reference/mediators/dss-mediator/#example-1-inline-single-request-operation)

In this example, an inline single request is configured and sent to the DSSCallMediatorTest service.

**Synapse Configuration**

<**proxy** xmlns="http://ws.apache.org/ns/synapse"

name="dssCallMediatorInlineSingleRequestProxy"

transports="http https"

startOnLoad="true">

<**description**/>

<**target**>

<**inSequence**>

<**dataServiceCall** serviceName="DSSCallMediatorTest">

<**source** type="inline"/>

<**operations** type="single">

<**operation** name="addEmployee">

<**param** name="employeeNumber" value="111"/>

<**param** name="firstname" value="Peter"/>

<**param** name="lastName" value="Parker"/>

<**param** name="email" value="peter@wso2.com"/>

<**param** name="salary" value="1000"/>

</**operation**>

</**operations**>

<**target** type="body"/>

</**dataServiceCall**>

<**respond**/>

</**inSequence**>

</**target**>

</**proxy**>

**Sample Request**

Invoke the dssCallMediatorInlineSingleRequestProxy proxy service:

http://localhost:8290/services/dssCallMediatorInlineSingleRequestProxy

**Response**

<**axis2ns3:REQUEST\_STATUS** xmlns:axis2ns3="http://ws.wso2.org/dataservice">SUCCESSFUL</**axis2ns3:REQUEST\_STATUS**>

### **Example 2: Inline batch request operation**[**¶**](https://apim.docs.wso2.com/en/latest/reference/mediators/dss-mediator/#example-2-inline-batch-request-operation)

In this example, an inline batch request is configured and sent to the DSSCallMediatorTest service.

**Synapse Configuration**

<**proxy** xmlns="http://ws.apache.org/ns/synapse"

name="dssCallMediatorInlineBatchRequestProxy"

transports="http https"

startOnLoad="true">

<**description**/>

<**target**>

<**inSequence**>

<**dataServiceCall** serviceName="DSSCallMediatorTest">

<**source** type="inline"/>

<**operations** type="batch">

<**operation** name="addEmployee">

<**param** name="employeeNumber" value="222"/>

<**param** name="firstname" value="John"/>

<**param** name="lastName" value="Doe"/>

<**param** name="email" value="john@wso2.com"/>

<**param** name="salary" value="2000"/>

</**operation**>

<**operation** name="addEmployee">

<**param** name="employeeNumber" value="333"/>

<**param** name="firstname" value="Joel"/>

<**param** name="lastName" value="Miller"/>

<**param** name="email" value="joel@wso2.com"/>

<**param** name="salary" value="3000"/>

</**operation**>

</**operations**>

<**target** type="body"/>

</**dataServiceCall**>

<**respond**/>

</**inSequence**>

</**target**>

</**proxy**>

**Sample Request**

Invoke the dssCallMediatorInlineBatchRequestProxy proxy service:

http://localhost:8290/services/dssCallMediatorInlineBatchRequestProxy

**Response**

<**axis2ns3:REQUEST\_STATUS** xmlns:axis2ns3="http://ws.wso2.org/dataservice">SUCCESSFUL</**axis2ns3:REQUEST\_STATUS**>

### **Example 3: Inline request box operation**[**¶**](https://apim.docs.wso2.com/en/latest/reference/mediators/dss-mediator/#example-3-inline-request-box-operation)

In this example, an inline batch request is configured and sent to the DSSCallMediatorTest service.

**Synapse Configuration**

<**proxy** xmlns="http://ws.apache.org/ns/synapse"

name="dssCallMediatorInlineRequestBoxProxy"

transports="http https"

startOnLoad="true">

<**description**/>

<**target**>

<**inSequence**>

<**dataServiceCall** serviceName="DSSCallMediatorTest">

<**source** type="inline"/>

<**operations** type="request-box">

<**operation** name="addEmployee">

<**param** name="employeeNumber" value="444"/>

<**param** name="firstname" value="Ellie"/>

<**param** name="lastName" value="Dina"/>

<**param** name="email" value="dina@wso2.com"/>

<**param** name="salary" value="4000"/>

</**operation**>

<**operation** name="getEmployeeByNumber">

<**param** name="employeeNumber" value="444"/>

</**operation**>

</**operations**>

<**target** type="body"/>

</**dataServiceCall**>

<**respond**/>

</**inSequence**>

</**target**>

</**proxy**>

**Sample Request**

Invoke the dssCallMediatorInlineRequestBoxProxy proxy service:

http://localhost:8290/services/dssCallMediatorInlineRequestBoxProxy

**Response**

<**axis2ns4:DATA\_SERVICE\_REQUEST\_BOX\_RESPONSE** xmlns:axis2ns4="http://ws.wso2.org/dataservice"><**Entries** xmlns="http://ws.wso2.org/dataservice"><**Entry**><**EmployeeNumber**>444</**EmployeeNumber**><**FirstName**>Ellie</**FirstName**><**LastName**>Dina</**LastName**><**Email**>dina@wso2.com</**Email**><**Salary**>4000</**Salary**></**Entry**></**Entries**></**axis2ns4:DATA\_SERVICE\_REQUEST\_BOX\_RESPONSE**>

### **Example 4: Single request operation when the source type is set to body**[**¶**](https://apim.docs.wso2.com/en/latest/reference/mediators/dss-mediator/#example-4-single-request-operation-when-the-source-type-is-set-to-body)

In this example, an inline single request is configured and sent to the DSSCallMediatorTest service.

**Synapse Configuration**

<**proxy** xmlns="http://ws.apache.org/ns/synapse"

name="dssCallMediatorSourceTypeBodyProxy"

transports="http https"

startOnLoad="true">

<**description**/>

<**target**>

<**inSequence**>

<**dataServiceCall** serviceName="DSSCallMediatorTest">

<**source** type="body"/>

<**target** type="body"/>

</**dataServiceCall**>

<**respond**/>

</**inSequence**>

</**target**>

</**proxy**>

**Sample Request**

Invoke the dssCallMediatorSourceTypeBodyProxy proxy service with the given payload:

http://localhost:8290/services/dssCallMediatorSourceTypeBodyProxy

Copied to clipboard<**addEmployee**>

<**EmployeeNumber**>555</**EmployeeNumber**>

<**Firstname**>Peter</**Firstname**>

<**LastName**>Parker</**LastName**>

<**Email**>peter@wso2.com</**Email**>

<**Salary**>5000</**Salary**>

</**addEmployee**>

**Response**

<**axis2ns3:REQUEST\_STATUS** xmlns:axis2ns3="http://ws.wso2.org/dataservice">SUCCESSFUL</**axis2ns3:REQUEST\_STATUS**>

### **Example 5: Batch request operation when source type is set to body**[**¶**](https://apim.docs.wso2.com/en/latest/reference/mediators/dss-mediator/#example-5-batch-request-operation-when-source-type-is-set-to-body)

In this example, an inline batch request is configured and sent to the DSSCallMediatorTest service.

**Synapse Configuration**

<**proxy** xmlns="http://ws.apache.org/ns/synapse"

name="dssCallMediatorSourceTypeBodyProxy"

transports="http https"

startOnLoad="true">

<**description**/>

<**target**>

<**inSequence**>

<**dataServiceCall** serviceName="DSSCallMediatorTest">

<**source** type="body"/>

<**target** type="body"/>

</**dataServiceCall**>

<**respond**/>

</**inSequence**>

</**target**>

</**proxy**>

**Sample Request**

Invoke the dssCallMediatorSourceTypeBodyProxy proxy service with the given payload.

http://localhost:8290/services/dssCallMediatorSourceTypeBodyProxy

Copied to clipboard<**addEmployee\_batch\_req**>

<**addEmployee**>

<**EmployeeNumber**>666</**EmployeeNumber**>

<**Firstname**>Miles</**Firstname**>

<**LastName**>Jimmy</**LastName**>

<**Email**>jimmy@wso2.com</**Email**>

<**Salary**>2000</**Salary**>

</**addEmployee**>

<**addEmployee**>

<**EmployeeNumber**>777</**EmployeeNumber**>

<**Firstname**>Dia</**Firstname**>

<**LastName**>Jesse</**LastName**>

<**Email**>jesse@wso2.com</**Email**>

<**Salary**>1500</**Salary**>

</**addEmployee**>

</**addEmployee\_batch\_req**>

**Response**

<**axis2ns3:REQUEST\_STATUS** xmlns:axis2ns3="http://ws.wso2.org/dataservice">SUCCESSFUL</**axis2ns3:REQUEST\_STATUS**>

### **Example 6: Request box operation when source type is set to body**[**¶**](https://apim.docs.wso2.com/en/latest/reference/mediators/dss-mediator/#example-6-request-box-operation-when-source-type-is-set-to-body)

In this example, an inline request box request is configured and sent to the DSSCallMediatorTest service.

**Synapse Configuration**

<**proxy** xmlns="http://ws.apache.org/ns/synapse"

name="dssCallMediatorSourceTypeBodyProxy"

transports="http https"

startOnLoad="true">

<**description**/>

<**target**>

<**inSequence**>

<**dataServiceCall** serviceName="DSSCallMediatorTest">

<**source** type="body"/>

<**target** type="body"/>

</**dataServiceCall**>

<**respond**/>

</**inSequence**>

</**target**>

</**proxy**>

**Sample Request**

Invoke the dssCallMediatorSourceTypeBodyProxy proxy service with the given payload.

http://localhost:8290/services/dssCallMediatorSourceTypeBodyProxy

Copied to clipboard<<request\_box>

<**addEmployee**>

<**EmployeeNumber**>888</**EmployeeNumber**>

<**Firstname**>William</**Firstname**>

<**LastName**>Sakai</**LastName**>

<**Email**>sakai@wso2.com</**Email**>

<**Salary**>3000</**Salary**>

</**addEmployee**>

<**getEmployeeByNumber**>

<**EmployeeNumber**>888</**EmployeeNumber**>

</**getEmployeeByNumber**>

</**request\_box**>

**Response**

<**axis2ns1:DATA\_SERVICE\_REQUEST\_BOX\_RESPONSE** xmlns:axis2ns1="http://ws.wso2.org/dataservice">

<**Entries** xmlns="http://ws.wso2.org/dataservice">

<**Entry**>

<**EmployeeNumber**>888</**EmployeeNumber**>

<**FirstName**>William</**FirstName**>

<**LastName**>Sakai</**LastName**>

<**Email**>sakai@wso2.com</**Email**>

<**Salary**>3000</**Salary**>

</**Entry**>

</**Entries**>

</**axis2ns1:DATA\_SERVICE\_REQUEST\_BOX\_RESPONSE**>

### **Example 7: Inline single request operation when the target type is set to the property**[**¶**](https://apim.docs.wso2.com/en/latest/reference/mediators/dss-mediator/#example-7-inline-single-request-operation-when-the-target-type-is-set-to-the-property)

In this example, an inline single request is configured and sent to the DSSCallMediatorTest service and gets the response to a property.

**Synapse Configuration**

<**proxy** xmlns="http://ws.apache.org/ns/synapse"

name="testDSSResposeTarget"

transports="http https"

startOnLoad="true">

<**description**/>

<**target**>

<**inSequence**>

<**dataServiceCall** serviceName="DSSCallMediatorSample">

<**source** type="inline"/>

<**operations** type="single">

<**operation** name="addEmployee">

<**param** name="employeeNumber" value="111"/>

<**param** name="firstname" value="Peter"/>

<**param** name="lastName" value="Parker"/>

<**param** name="email" value="peter@wso2.com"/>

<**param** name="salary" value="1000"/>

</**operation**>

</**operations**>

<**target** type="property" name="responseValue"/>

</**dataServiceCall**>

<**log** level="custom">

<**property** name="responseValue" expression="$ctx:responseValue"/>

</**log**>

<**respond**/>

</**inSequence**>

</**target**>

</**proxy**>

**Sample Request**

Invoke the testDSSResposeTarget proxy service with the given payload.

http://localhost:8290/services/testDSSResposeTarget

**Response**

The following log will appear in the server console:

INFO {LogMediator} - {proxy:test} reponseValue = SUCCESSFUL