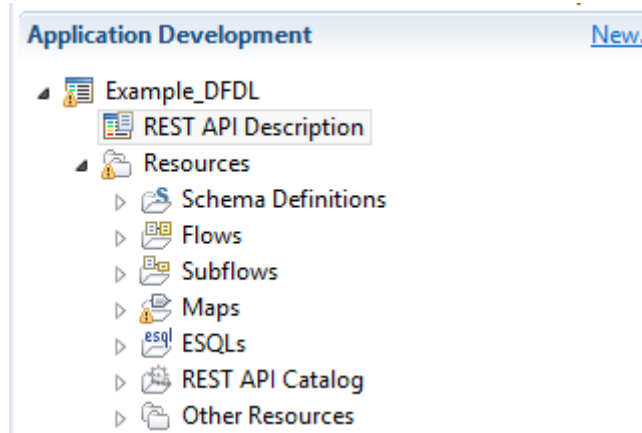


Código fuente: [Example_DFDL.zip](#)



▼ Header

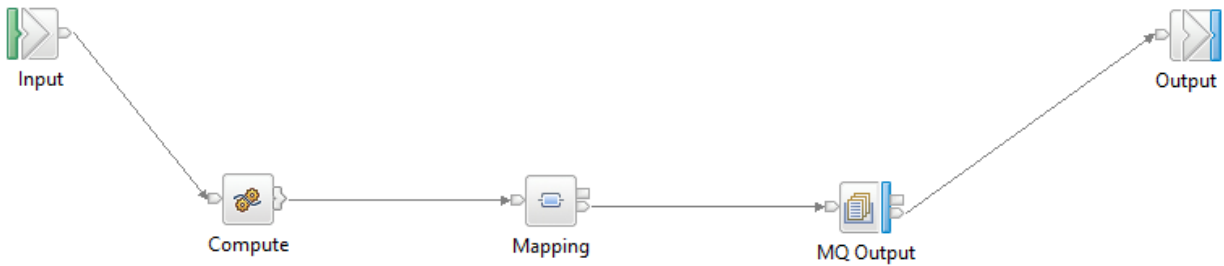
REST API base URL Title De

You can access the operations in the REST API by pointing your web browser to the following URL, where <hostname> is the host name and <http://<hostname>:<port_number>/example_dfdl/v1

▼ Resources

▼ /dfdl

GET	getDfdl	Retrieve dfdl			
Name	Parameter type	Data type	Format	Required	Description
<input type="text" value="param1"/>	<input type="text" value="query"/> ▼	<input type="text" value="string"/> ▼	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
<input type="text" value="param2"/>	<input type="text" value="query"/> ▼	<input type="text" value="string"/> ▼	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
Response status	Description	Array			
<input type="text" value="200"/>	<input type="text" value="The operation was successful."/>	<input type="checkbox"/>			



Example_DFDL getDfdl.subflow getDfdl_Compute.esql

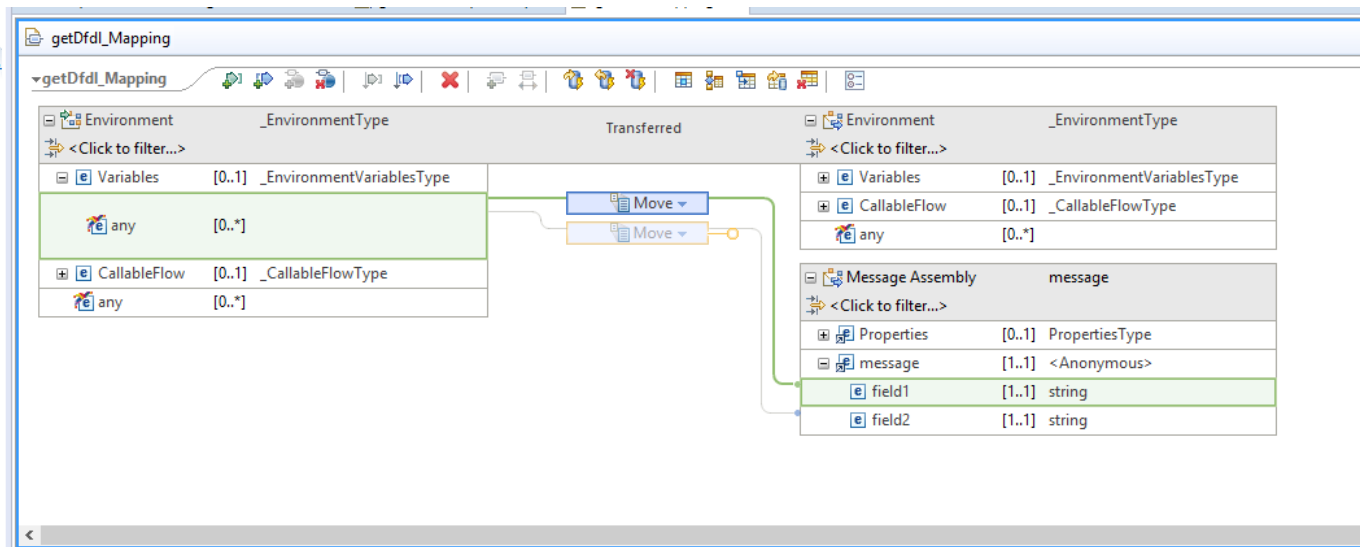
```

1 CREATE COMPUTE MODULE getDfdl_Compute
2   CREATE FUNCTION Main() RETURNS BOOLEAN
3   BEGIN
4
5     SET Environment.Variables.param1 = InputLocalEnvironment.REST.Input.Parameters.param1;
6     SET Environment.Variables.param2 = InputLocalEnvironment.REST.Input.Parameters.param2;
7
8     --SET OutputRoot.JSON.Data.mensaje.campo1 = Environment.Variables.param1;
9     --SET OutputRoot.JSON.Data.mensaje.campo2 = Environment.Variables.param2;
10
11    RETURN TRUE;
12  END;
13
14 END MODULE;
15
    
```

getDfdl_Mapping

<Click to filter...>

Environment	_EnvironmentType	Transferred	Environment	_EnvironmentType										
Variables	[0..1] _EnvironmentVariablesType	<div style="display: flex; justify-content: center; gap: 10px;"> <div style="border: 1px solid #ccc; padding: 2px;">Move</div> <div style="border: 1px solid #ccc; padding: 2px;">Move</div> </div>	Variables	[0..1] _EnvironmentVariablesType										
any	[0..*]		CallableFlow	[0..1] _CallableFlowType										
CallableFlow	[0..1] _CallableFlowType		any	[0..*]										
any	[0..*]		<div style="border-bottom: 1px solid #ccc; padding-bottom: 5px;"> <Click to filter...> </div> <table border="1" style="width: 100%; border-collapse: collapse; font-size: 0.8em;"> <thead> <tr> <th style="width: 30%;">Message Assembly</th> <th style="width: 70%;">message</th> </tr> </thead> <tbody> <tr> <td>Properties</td> <td>[0..1] PropertiesType</td> </tr> <tr> <td>message</td> <td>[1..1] <Anonymous></td> </tr> <tr> <td>field1</td> <td>[1..1] string</td> </tr> <tr> <td>field2</td> <td>[1..1] string</td> </tr> </tbody> </table>		Message Assembly	message	Properties	[0..1] PropertiesType	message	[1..1] <Anonymous>	field1	[1..1] string	field2	[1..1] string
Message Assembly	message													
Properties	[0..1] PropertiesType													
message	[1..1] <Anonymous>													
field1	[1..1] string													
field2	[1..1] string													



Properties Problems Outline Tasks Deployment Log

Transform - Move

Cardinality Input array indexes:

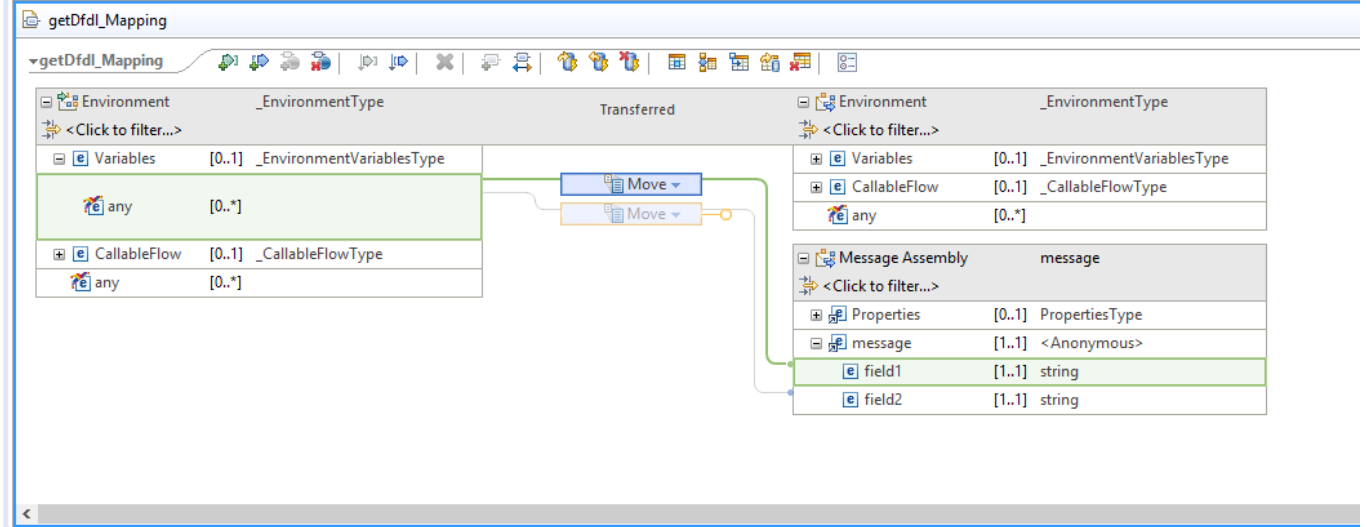
Environment / Variables / any [1]

Variables

Condition

Sort

Order



Properties Problems Outline Tasks Deployment Log

Transform - Move

Cardinality Input Variables:

Name	Edit...
param1	

Condition

Sort

Order

The image shows two screenshots of the IBM App Connect Designer interface, illustrating the configuration of a DFDL mapping.

Top Screenshot: Shows the 'getDfdl_Mapping' workspace. On the left, the source data structure is an 'Environment' of type '_EnvironmentType' containing 'Variables' (cardinality [0..1], type '_EnvironmentVariablesType') and 'any' (cardinality [0..*]). On the right, the target data structure is a 'Message Assembly' of type 'message' containing 'Properties' (cardinality [0..1], type 'PropertiesType'), 'message' (cardinality [1..1], type '<Anonymous>'), 'field1' (cardinality [1..1], type 'string'), and 'field2' (cardinality [1..1], type 'string'). A 'Move' connector is placed between the 'any' element and the 'field2' element. The connector's 'Input array indexes' property is highlighted with a red box and set to '2'.

Bottom Screenshot: Shows the 'Transform - Move' properties panel. The 'Input Variables' section is expanded, showing a table with the following data:

Name	Edit...
param2	

The 'Name' field in the table is highlighted with a red box.

The screenshot displays the IBM App Connect 11 development environment. At the top, a flow diagram shows a sequence of nodes: **Input**, **Compute**, **Mapping**, and **MQ Output**. The **MQ Output** node is highlighted with a red box, and its label **MQOutput : MQ Output** is also boxed. Below the flow diagram, the **MQ Output Node Properties - MQ Output** panel is open. The **Basic** tab is selected, showing the **Queue name** field with the value **QL.IN.AS400**, which is highlighted with a red box. Below this, the **MQ Explorer - Content** panel is visible. On the left, a tree view of queues includes **QL.IN.AS400**, highlighted with a red box. The main area shows a **Message browser** for **Queue Manager Name: QM_01** and **Queue Name: QL.IN.AS400**. A **Message 5 - Properties** dialog is open, showing the **Data** tab. The **Message data** field contains the value **123 ,456**, highlighted with a red box. The **Message data bytes** section shows a hex dump of the message data.

GET ▼ http://aprendeibm:7080/example_dfdl/v1/dfdl?param1=123¶m2=456

Params ● Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

	KEY	VALUE
<input checked="" type="checkbox"/>	param1	123
<input checked="" type="checkbox"/>	param2	456

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize XML ▼

1 123 ,456

schema01.xsd

Test Parse Model Test Serialize Model Hide properties Show all sections Focus on selected Show quick outline Create logical instance

Messages

A message is a global element that models an entire document of data.

Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
message		1	1		
sequence		1	1		
field1	string	1	1	a	a
field2	string	1	1	a	a

[Add a Local Element](#)

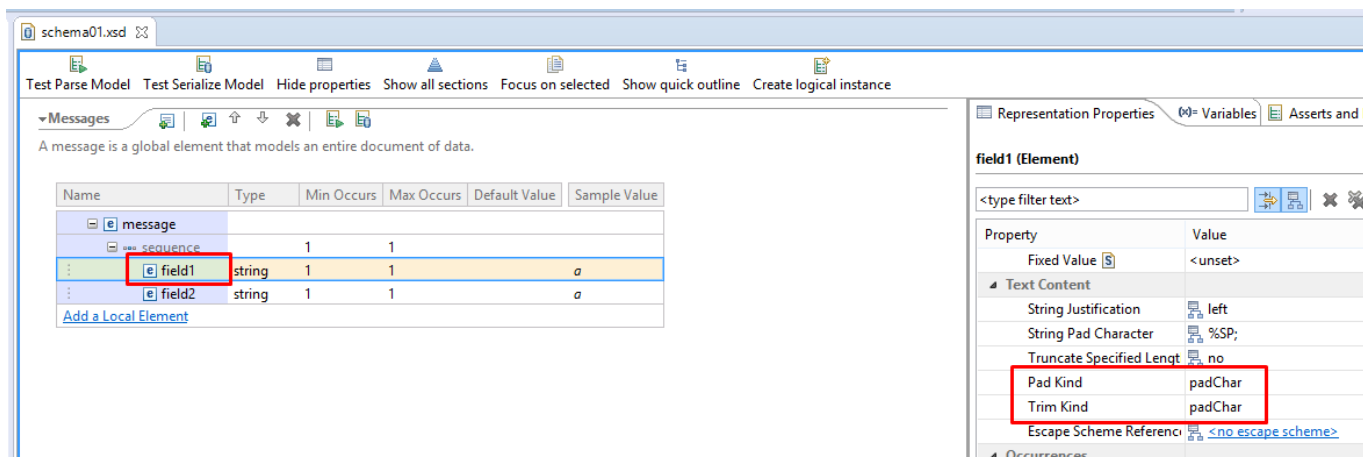
Representation Properties (0)= Variables Assets and Discriminators

sequence

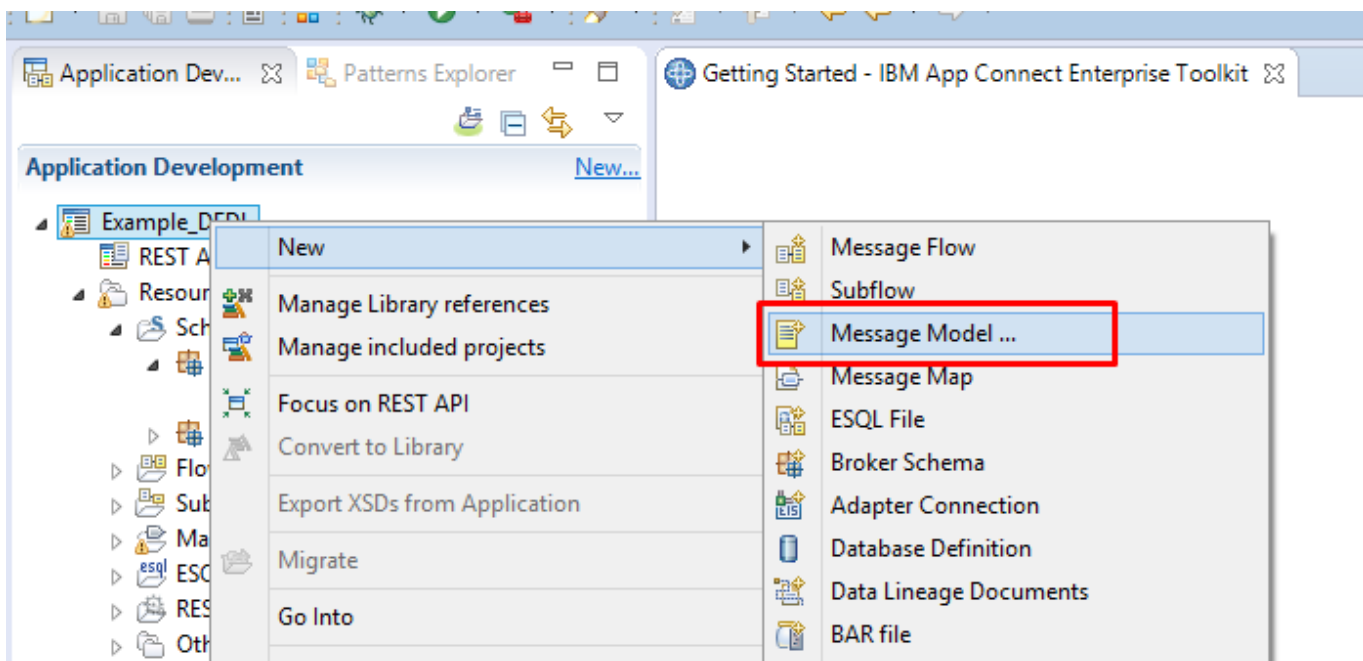
<type filter text>

Property	Value
Initiated Content	no
Sequence Kind	ordered
Occurrences	
Min Occurs	1
Max Occurs	1
Alignment	
Delimiters	
Separator	,
Initiator	<no initiator>
Terminator	<no terminator>
Output New Line	%CR,%LF

Delimiters		
Separator	,	
Separator Suppressor	never	
Separator Position	infix	
Initiator	<no initiator>	



Crear el XSD con la definición a trabajar la trama posicional



Create a new message model file

Select the message model type or format

XML

- SOAP XML XML data for use in Web Services.
- Other XML All other XML data.

Text and binary

- CSV text Comma Separated Values data, a delimited text format commonly used as an export format by spreadsheets and databases.
- Record-oriented text Text data formats where delimited fields are grouped into records.
- COBOL Data for COBOL programs
- C Data for C programs
- Other text or binary All other text or binary data formats.

Enterprise Information Systems

- SAP Data from SAP systems including IDoc and BAPI
- Siebel Data from Siebel systems
- PeopleSoft Data from PeopleSoft
- JD Edwards Data from JD Edwards systems

Other

- CORBA IDL Data from CORBA
- Database record Records from relational databases
- MIME Data for extended email format
- IBM supplied Predefined data format

New Message Model


Other text or binary

Choose how you would like to create your text data or binary data message model.

IBM App Connect Enterprise requires a message model in order to parse, serialize and validate custom data. A message model also speeds up development of your integration applications by enabling ESC assist and graphical maps.

- Create a DFDL schema file using this wizard to guide you
- Create an empty DFDL schema file, I will model my data using the DFDL schema editor
- Import or replace the IBM supplied DFDL schema property defaults for other text or binary.

The first option is suitable if you have a text format that consists of a number of records or segments (header, repeating body, optional trailer). The records can have either fixed-length or variable-length records and fields can have initiators.



The diagram shows a message structure with three sections: Header, Body, and Trailer. The Header section contains a 'CREATED' field with a date '08/22/2011 1:45:23 PM' and a 'COUNT' field with a value of '5', followed by a carriage return and line feed sequence '<CRLF>'. The Body section contains a 'START' field with an 'ID' field (value '1'), a 'STATUS' field (value 'SENT'), and a 'QTY' field (value '300'), followed by a carriage return and line feed sequence '<CRLF>'. The Trailer section contains an 'EDITED' field with a date '08/22/2011 2:15:05 PM' and a carriage return and line feed sequence '<CRLF>'. A yellow sticky note is overlaid on the diagram, showing the following DFDL code:

```
CREATED{DATE=08/22/2011 1:45:23 PM|COUNT=5}␣  
START{ID=1|STATUS=SENT|QTY=300}␣  
START{ID=2|STATUS=RETAINED|QTY=12}␣  
START{ID=3|STATUS=PENDING|QTY=1456}␣  
START{ID=4|STATUS=SENT|QTY=0941}␣  
START{ID=5|STATUS=RETURNED|QTY=0600}␣  
EDITED{DATE=08/22/2011 2:15:05 PM}␣
```

New Message Model

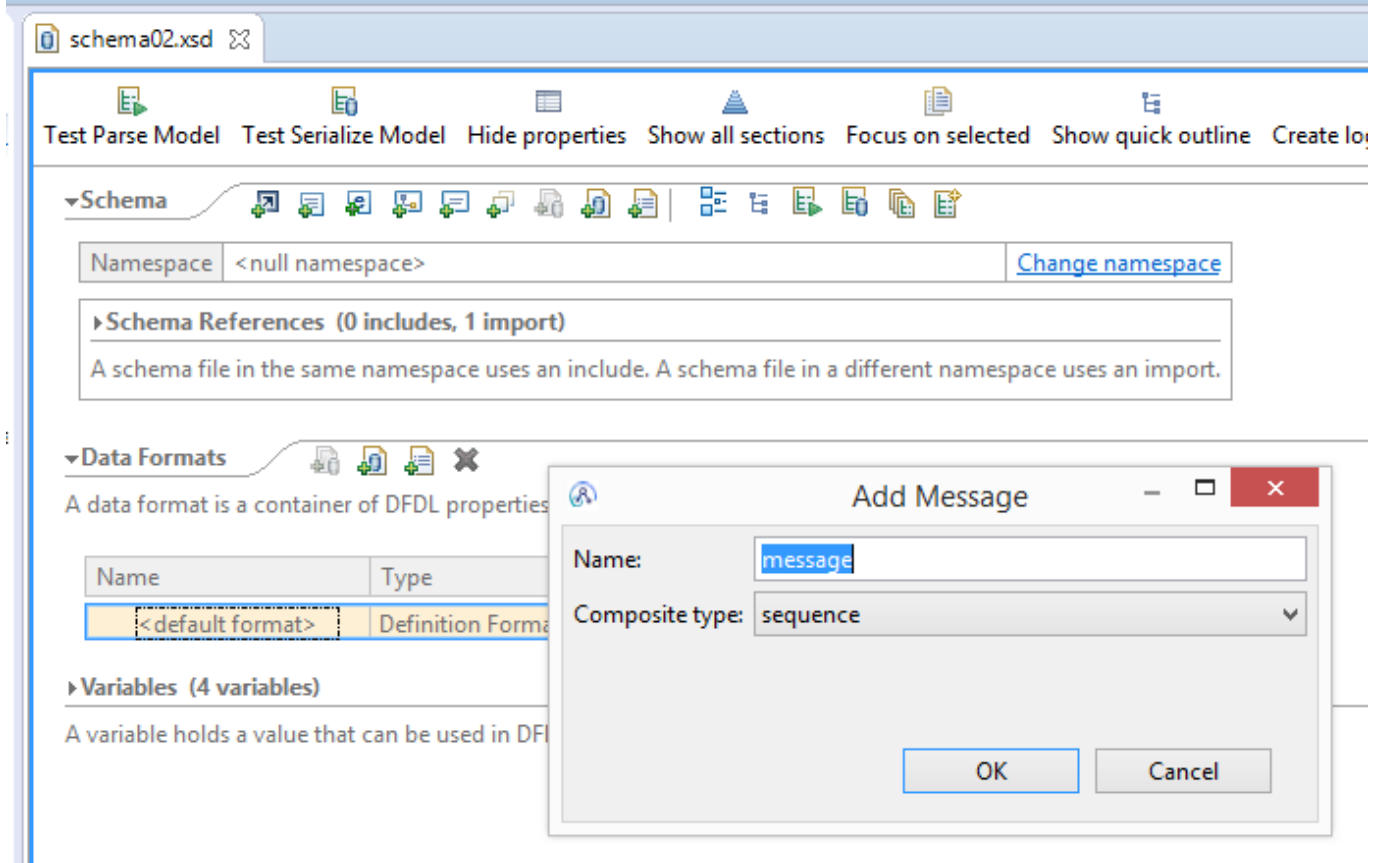
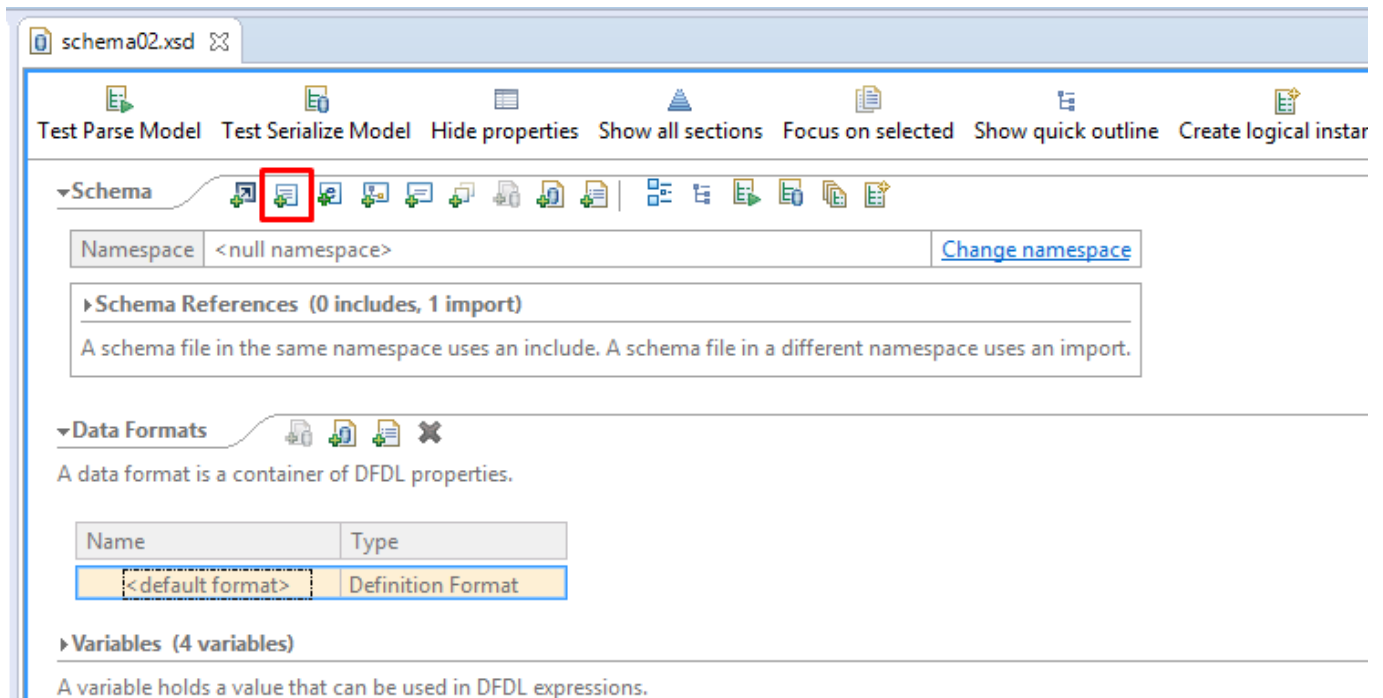
Create an empty Data Format Description Language (DFDL) Schema

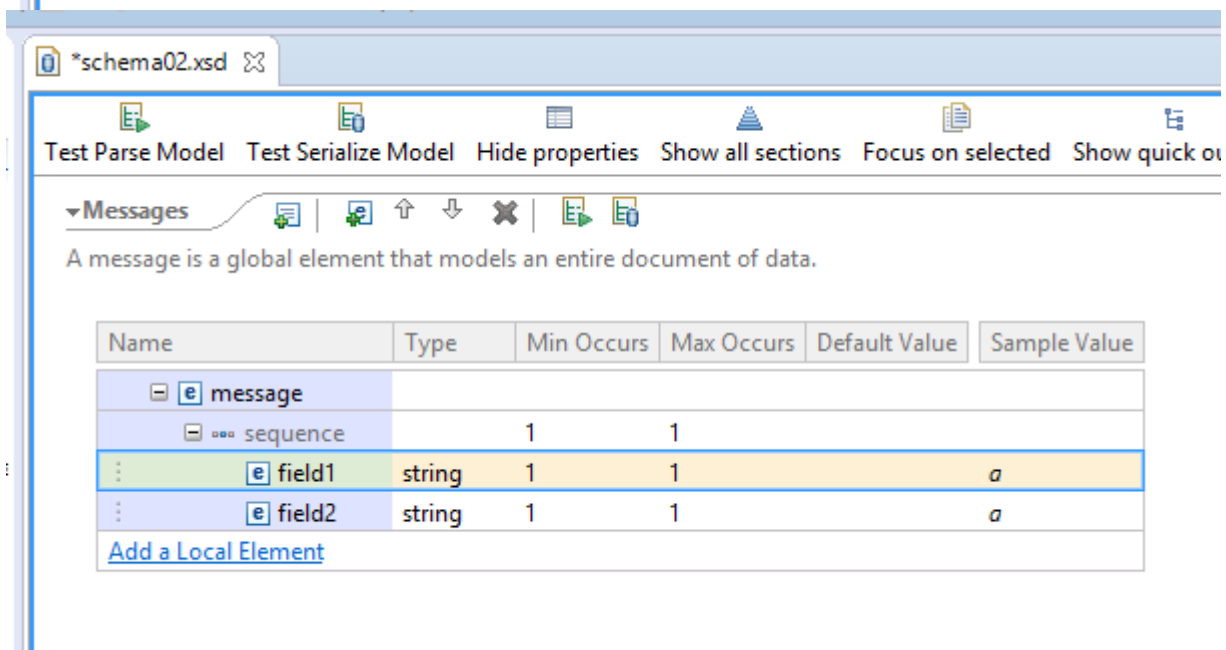
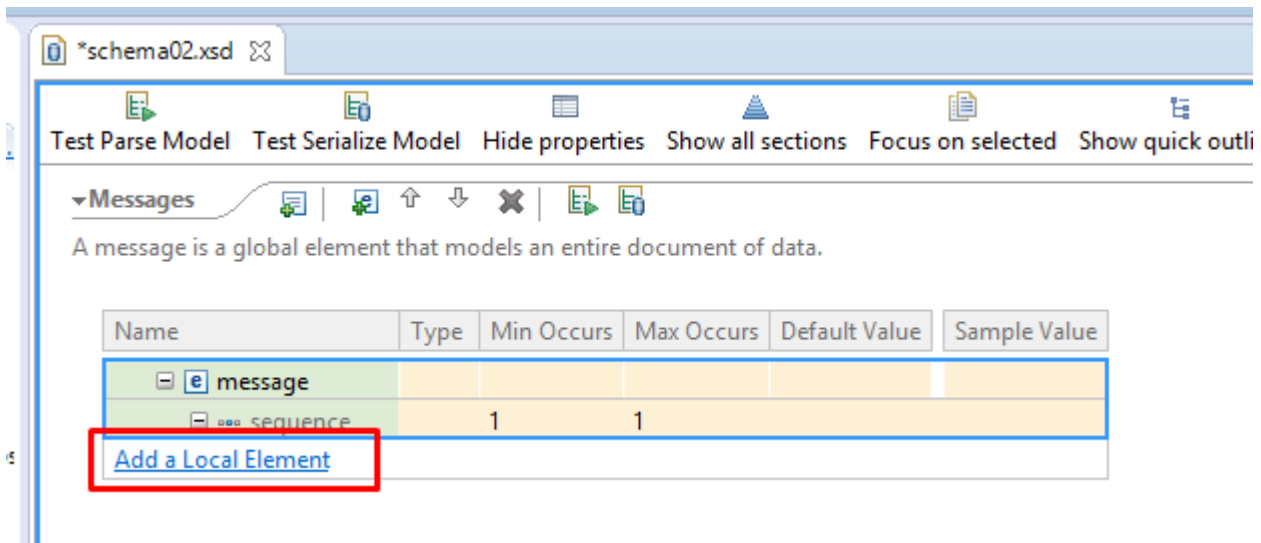
Specify the location and name of the DFDL schema, and specify the name of the message.

Application or Library:

Folder:

DFDL schema file name:





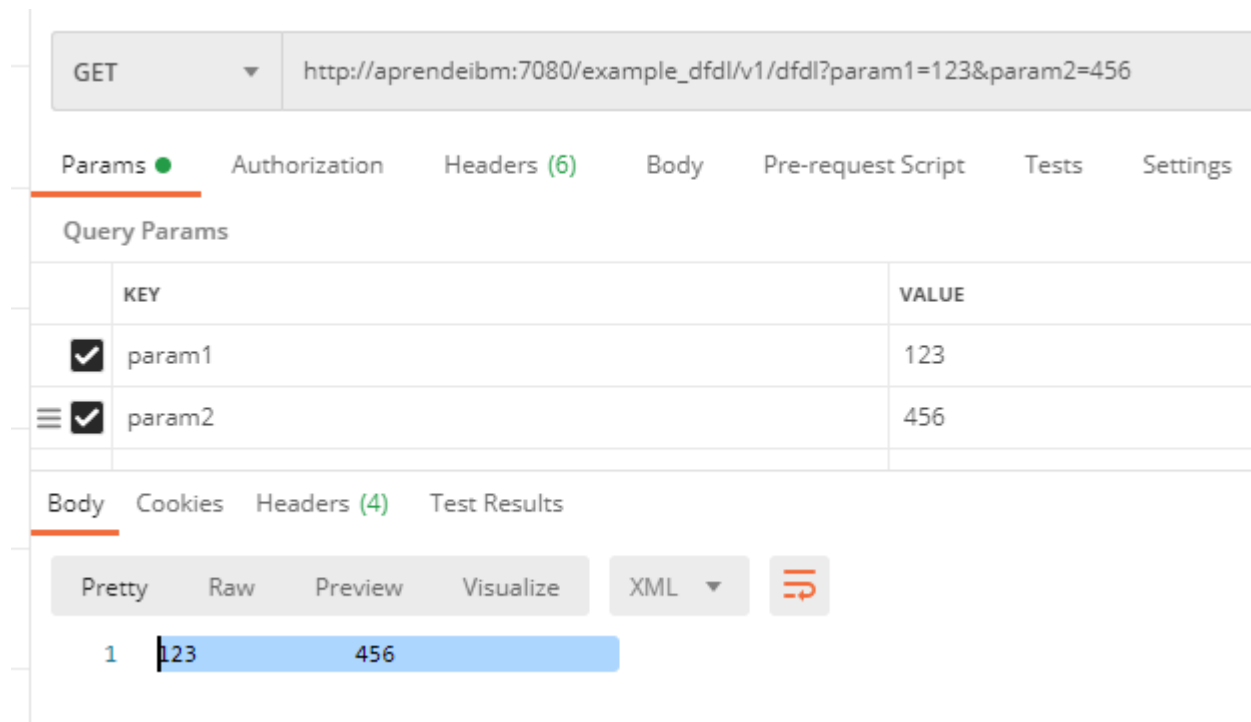
En caso que se requiera que el separador no sea coma y los valores esten pegados, setear

Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
message					
sequence		1	1		
field1	string	1	1		a
field2	string	1	1		a

The 'sequence' row is highlighted with a red box.

Property	Value
Sequence Kind	ordered
Occurrences	
Min Occurs	1
Max Occurs	1
Alignment	
Delimiters	
Separator	{ }
Separator Suppression Policy	never
Separator Position	infix
Initiator	<no initiator>
Terminator	<no terminator>

The 'Separator' property is highlighted with a red box.




GET `http://aprendeibm:7080/example_dfdl/v1/dfdl?param1=123¶m2=456`

Params ● Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

	KEY	VALUE
<input checked="" type="checkbox"/>	param1	123
<input checked="" type="checkbox"/>	param2	456

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize XML 

```
1 123 456
```

Julio Pari (IT Architect IBM)



Si te ha interesado este artículo y deseas un apoyo o asesoría en algún requerimiento, envíame un mensaje a: (info@juliopari.com) o sino a través de LinkedIn: <https://www.linkedin.com/in/juliopari/>