

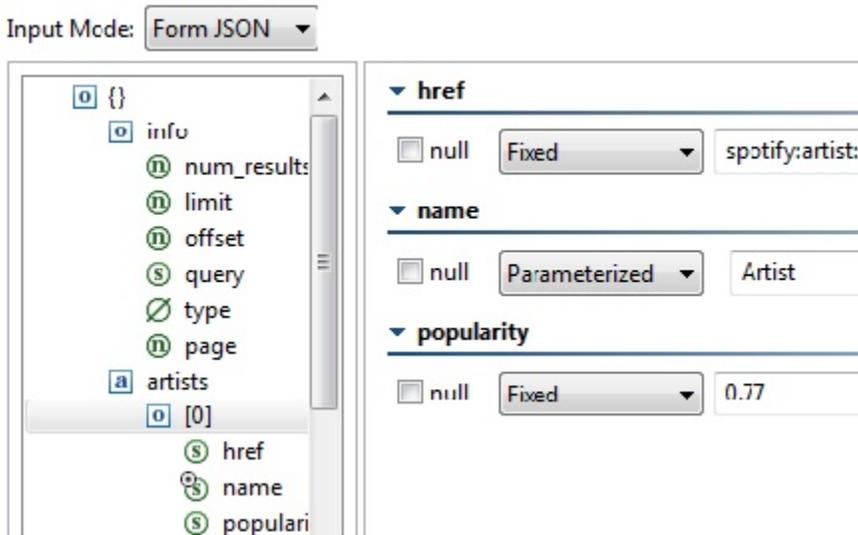
Form JSON View Options

This topic covers the Form JSON view, which provides a tree view of the JSON message.

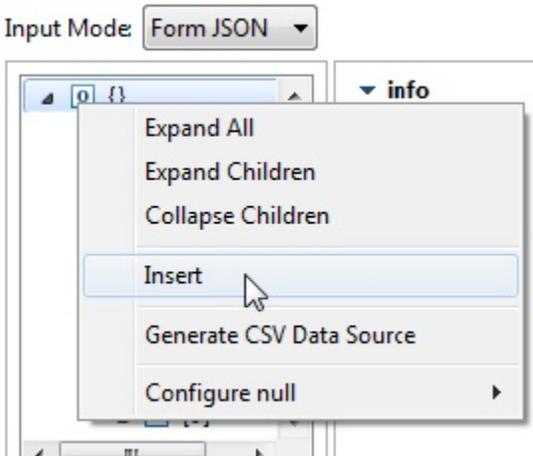
The Form JSON view allows you to graphically construct and review JSON messages. It is a JSON-focused version of the Form Input view (described in [Form Input View Options](#))—just as the Form XML view (described in [Form XML View Options](#)) is an XML-focused version of the Form Input view.

Building the Tree

If you paste a JSON message into the Literal view, then switch to Form JSON view, the Form JSON view will automatically be populated with the provided values and show a graphical representation of the message.



Additionally, you can build out the tree to reflect the appropriate message structure (i.e., by inserting, removing, copying, and pasting nodes). You can copy and paste into arrays and objects.

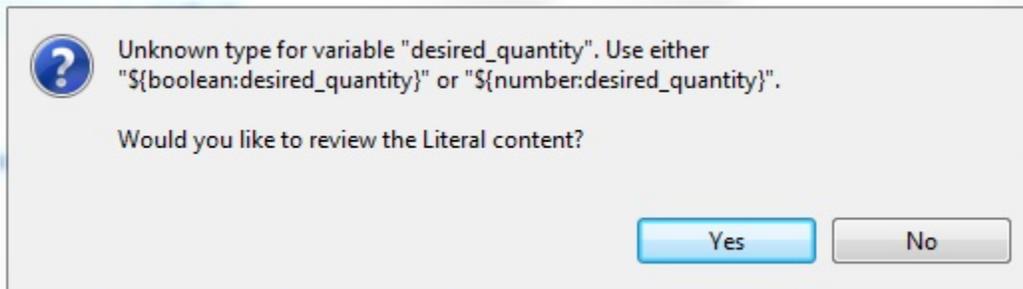


Parameterizing Number or Boolean Fields in a Literal Message

With all message formats, including JSON, you can use the `${}` syntax to access data source values from data sources or data banks—for example, you could use `$(title)` to reference the title column from a data source or use `$(books:publisher)` to reference the publisher column from the books sheet of an Excel data source.

Specifically for JSON, a special notation is used for parameterizing a number or boolean field within a JSON message: `$(number:<value>)` or `$(boolean:<value>)`. For example, to parameterize a number field with the column Count, you would use `$(number:Count)`.

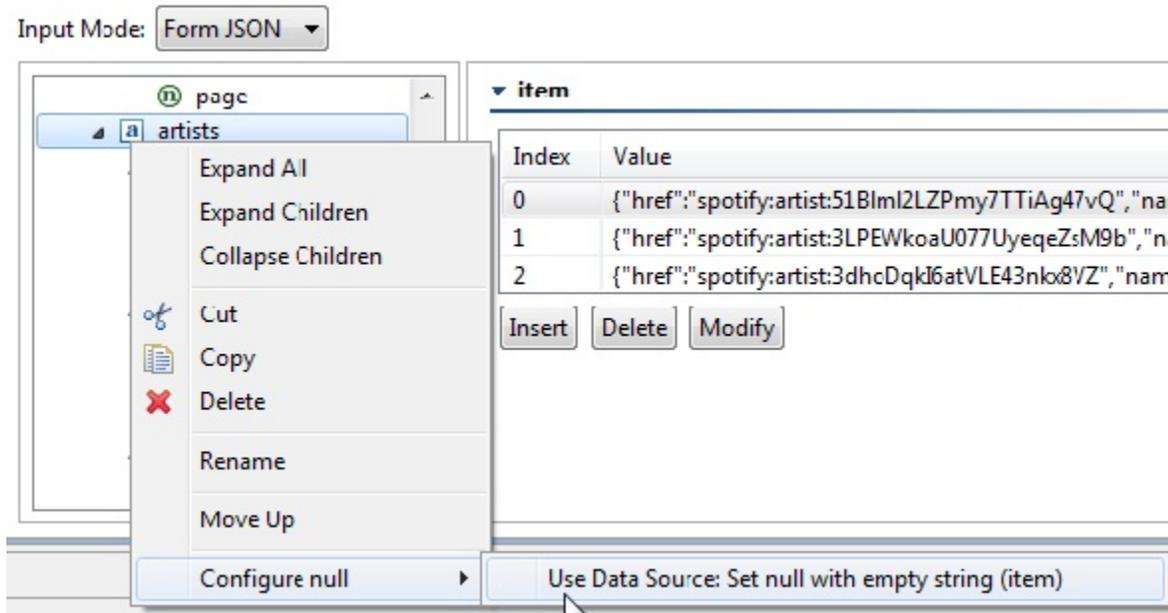
If you forget to use this format for number or boolean fields and switch from Literal view to Form JSON view, you will see a warning dialog such as the following:



Editing Values

You can specify values using the fixed, parameterized, auto, and script options available across the various tree views.

Note that Form Input options which are applicable to JSON are available from this view (with relevant terminology changes, such as null vs nil). This includes populating the entire tree with values in a data source, generating a CSV data source template, replacing specific elements with data source values, etc. For details, see [Form Input View Options](#).



Viewing the Literal JSON

Switching to the Literal view will apply the contents of Form JSON to the Literal view. Right-clicking JSON messages in the Literal view allows you to automatically "beautify" the JSON.

Input Mode:

Text File Data Source

MIME Type:

```
1 {
2   "products" : [
3     {
4       "product_id" : "${product_id}",
5       "desired_quantity" : "${desired_quantity}"
6     }
7   ],
8   "stores" : [
9     {
10      "store_id" : "${store_id}"
11    }
12  ],
13  "nearby" : "${nearby}",
14  "radius" : ${number:radius},
15  "multichannel_options" :
16    {
17      "multichannel_option" : "${multichannel_option}"
18    }
19 }
```

