Talend User Components tElasticSearch*

Purpose

These components are dedicated to work with ElasticSearch in the Data Integration suite of Talend.
These components are working with the http API from ElasticSearch. This is the preferred way to write data into an index according the recommendation of ElasticSearch.

Component	Purpose
tElasticSearchIndexOutput	Write JSON documents into an ElasticSearch index.
tElasticSearchIndexErrors	Provides the errors from the last write request performed by tElasticSearchIndexOutput
tElasticSearchRequest	Send http requests e.g. read or search documents in ElasticSearch

Talend-Integration

You find this component in the studio palette under: ElasticSearch

Component tElasticSearchIndexOutput

This component writes JSON documents with batch post requests into an ElasticSearch index.

Basic settings

<table>
<tr><th>Property</th><th>Content</th></tr>
<tr><td>Schema</td><td>This component uses a fixed schema:
<table>
<tr><th>Column</th><th>Purpose</th></tr>
<tr><td>key</td><td>This value is the actual document id. With this id the document can be addressed within the index.</td></tr>
<tr><td>json</td><td>The json document to be indexed. This can be a String or a JsonNode (Jackson API)</td></tr>
<tr><td>delete</td><td>If this boolean value is true, the document with the given key will be deleted from the index. The column json can be null in this case.</td></tr>
</table></td></tr>
<tr><td>Server Nodes</td><td>ElasticSearch allows client-side load balancing. The client can determine which server should be used. To allow this setup here a list of ElasticSearch hosts comma separated.
The hosts are setup with hostname:port</td></tr>
<tr><td>Use encrypted conection</td><td>The connection will be established encrypted.</td></tr>
<tr><td>Use Authentication</td><td>Activate the authentication for the connection</td></tr>
<tr><td>User / Password</td><td>The user credentials if authentication is required</td></tr>
<tr><td>Index</td><td>The index the component has to write in</td></tr>
<tr><td>Object Type</td><td>The type of objects written into the index</td></tr>
</table>

The JSON documents can easily be created with the tJSONDoc* components.

Advanced settings

Property	Content
Batch Size	The number of documents send as batch to the index

Return values

Return value	Content
ERROR_MESSAGE	Error messages (without details about the records failed inserting)
NB_LINE	The amount of inserted, updated or deleted documents.
NB_LINE_INSERTED	The amount documents inserted
NB_LINE_DELETED	The amount documents deleted

Component tElasticSearchIndexErrors

This returns the errors occurred while indexing documents with tElasticSearchIndexOutput.

Basic settings

Property	Content
ElasticSearch component	Choose the tElasticSearchIndexOutput component which errors should be returned here
Schema	This component uses a fixed schema: Column: key — Purpose: The key of the affected document. Column: operation — Purpose: Which operation was performed: index or delete Column: failure_message — Purpose: The actual error

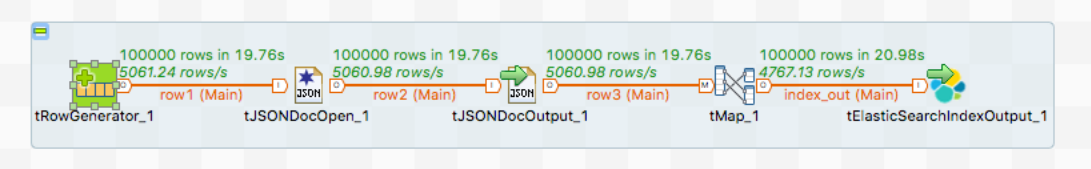
Return values

Return value	Content
ERROR_MESSAGE	Error messages of the operation of this component (not the index errors)
NB_LINE	Number index errors returned
COUNT_ERRORS	The number detected errors

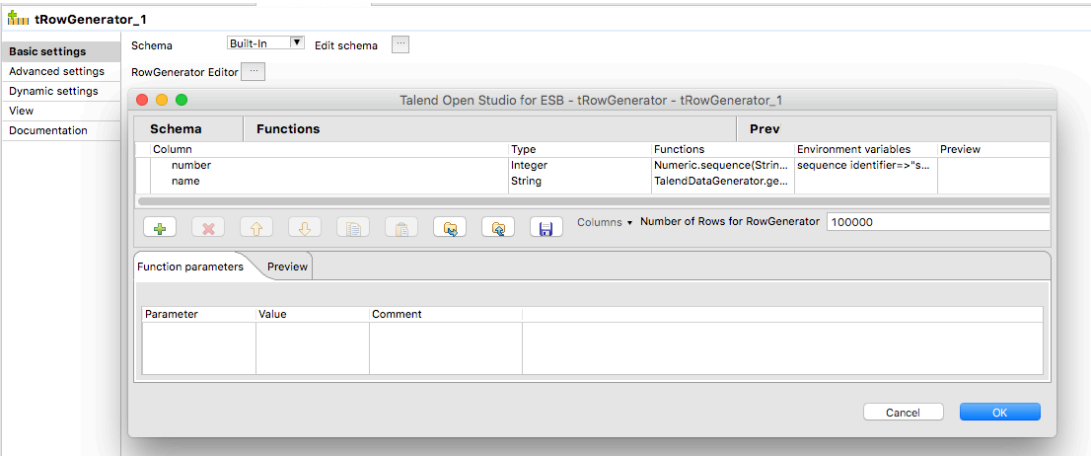
Example Use Cases

This is a load testing job. The source of the data is the row generator from Talend. Its values will be converted with the tJSONDocOpen + tJSONDocOutput component into a json document and after that send to the ElasticSearch index.

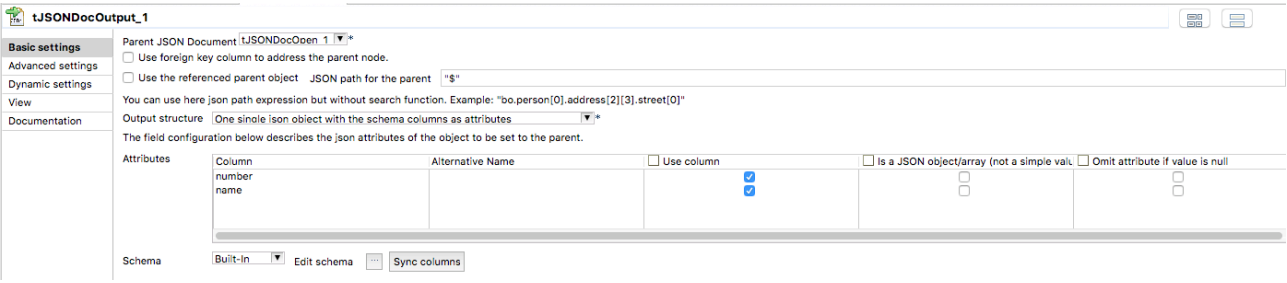
This is the example job design



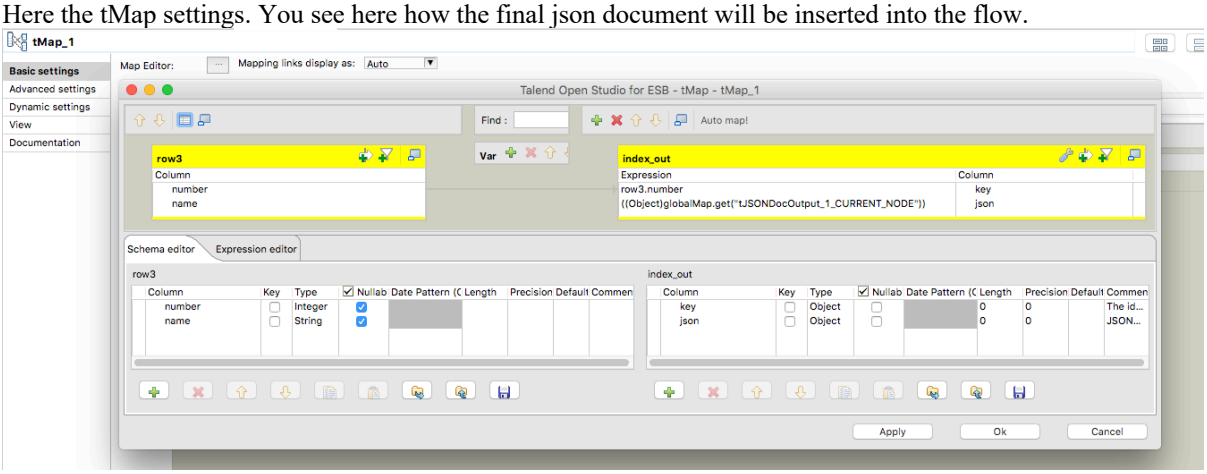
Here the settings of the row generator



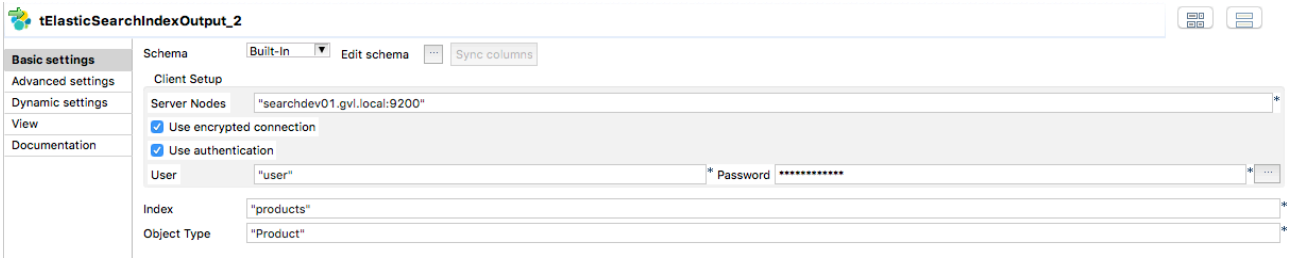
The json component settings. The tJSONDocOpen component simply creates the json document and does not do anything to the flow running through it.



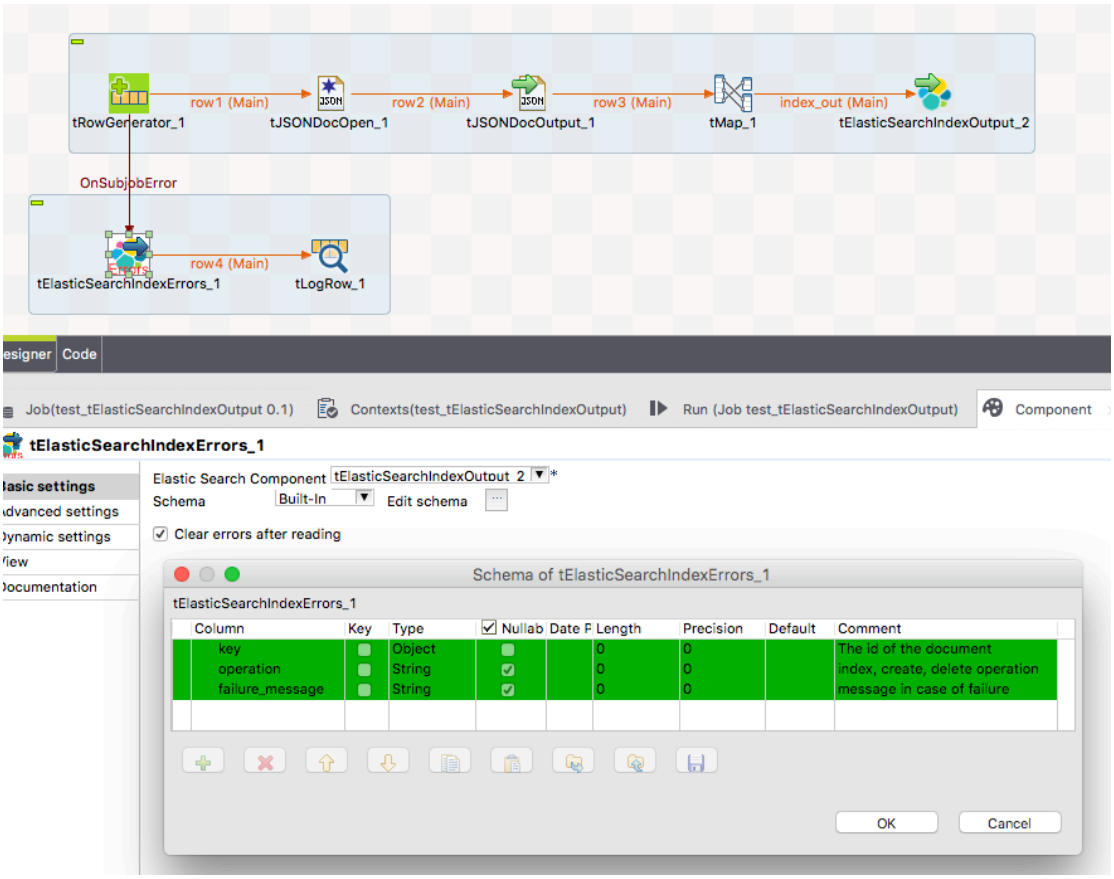
Here the tMap settings. You see here how the final json document will be inserted into the flow.



And finally, the setting of the tElasticSearchIndexOutput component:



Here another job design with the output of the errors occurred while indexing:



Component tElasticSearchRequest

This component can send arbitrary requests to ElasticSearch.

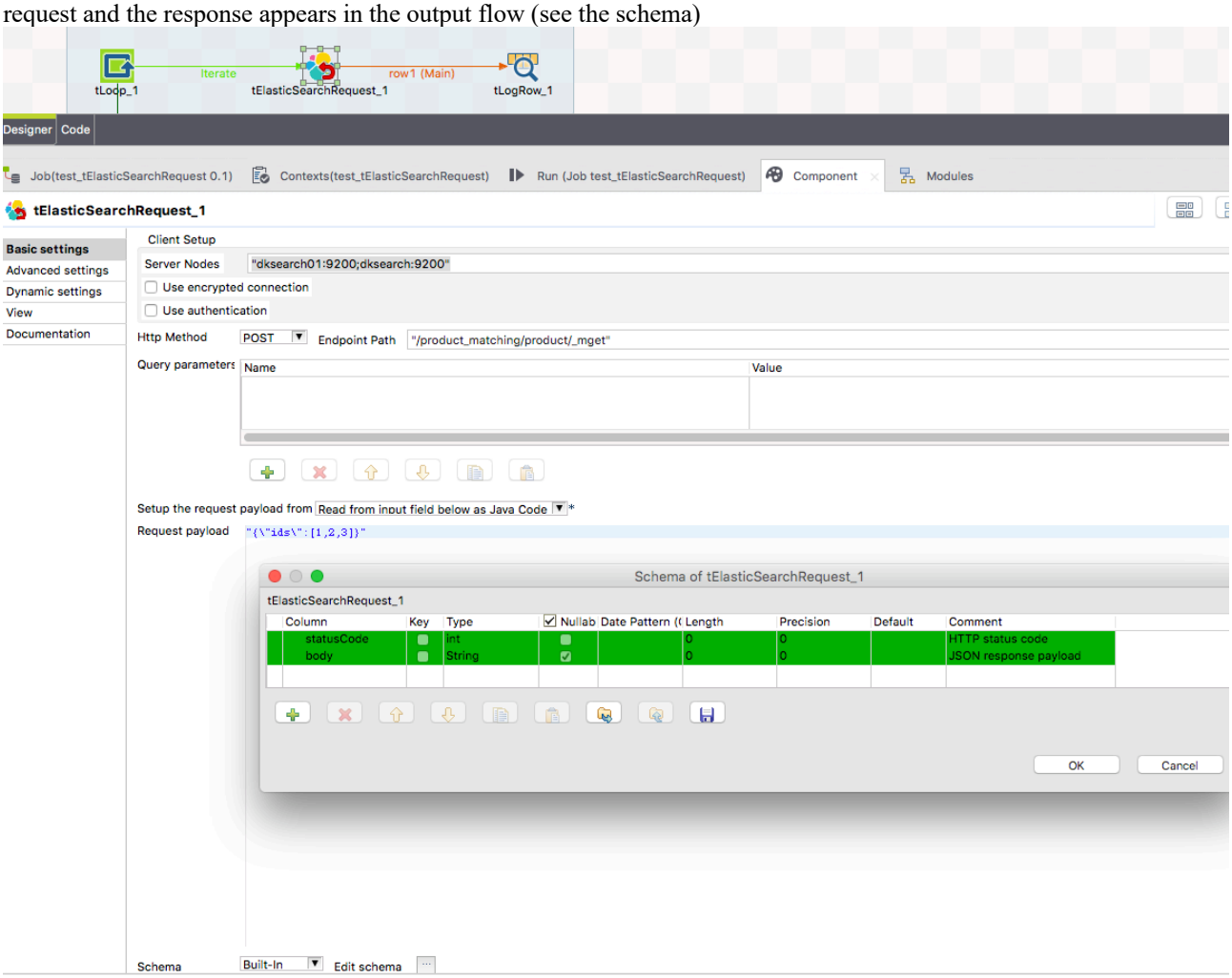
Basic settings

<table>
<tr><th>Property</th><th>Content</th></tr>
<tr><td>Server Nodes</td><td>ElasticSearch allows client-side load balancing. The client can determine which server should be used. To allow this setup here a list of ElasticSearch hosts comma separated.
The hosts are setup with hostname:port</td></tr>
<tr><td>Use encrypted conection</td><td>The connection will be established encrypted.</td></tr>
<tr><td>Use Authentication</td><td>Activate the authentication for the connection</td></tr>
<tr><td>User / Password</td><td>The user credentials if authentication is required</td></tr>
<tr><td>Http Method</td><td>GET, POST, PUT, DELETE</td></tr>
<tr><td>Endpoint path</td><td>The path to the desired service endpoint</td></tr>
<tr><td>Query parameters</td><td>A list of query parameters and their values</td></tr>
<tr><td>Setup request payload from</td><td>Read from input field as Java code: Write your payload for the request (e.g. a search request) as json as Java String (with e.g. code snippets like String operations etc)
Read from input flow column: Get the payload from an input flow column. In this mode the component needs an incoming flow.</td></tr>
<tr><td>Request Payload</td><td>The payload for the request in case of the first setup option above.</td></tr>
<tr><td>Request Payload input column</td><td>The payload will be read from an input column. This option appears for the second setup option above.
The input flow will be directed through the component without changes. Choose here the column containing the payload for the input flow.</td></tr>
<tr><td>Schema</td><td>This component uses a fixed schema:
<table>
<tr><th>Column</th><th>Purpose</th></tr>
<tr><td>statusCode</td><td>The response http status code</td></tr>
<tr><td>body</td><td>The response payload</td></tr>
</table>
</td></tr>
</table>

Example job with 2 different kind of preparing the payload for the request

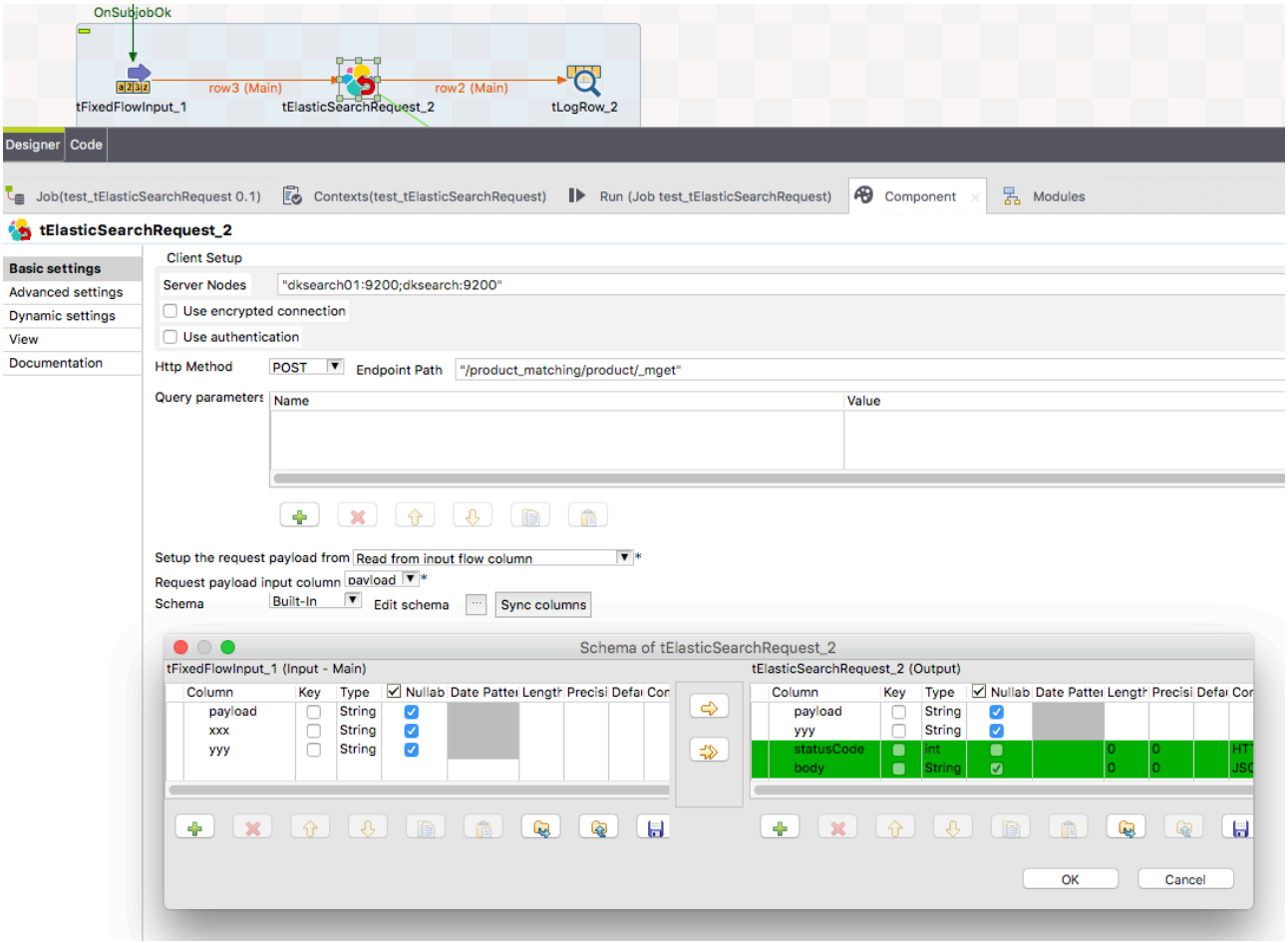
Get the request payload from an input field.

The component will be triggered with iteration or via OnSubjobOk or onComponentOk. The component send the request and the response appears in the output flow (see the schema)



Part of the job for send a request which payload comes from an incoming flow.

The flow goes through the component. Incoming columns will be directed to output columns if they exist in the outgoing flow.



This way you can use information from the input (request) also for processing the response of the request.