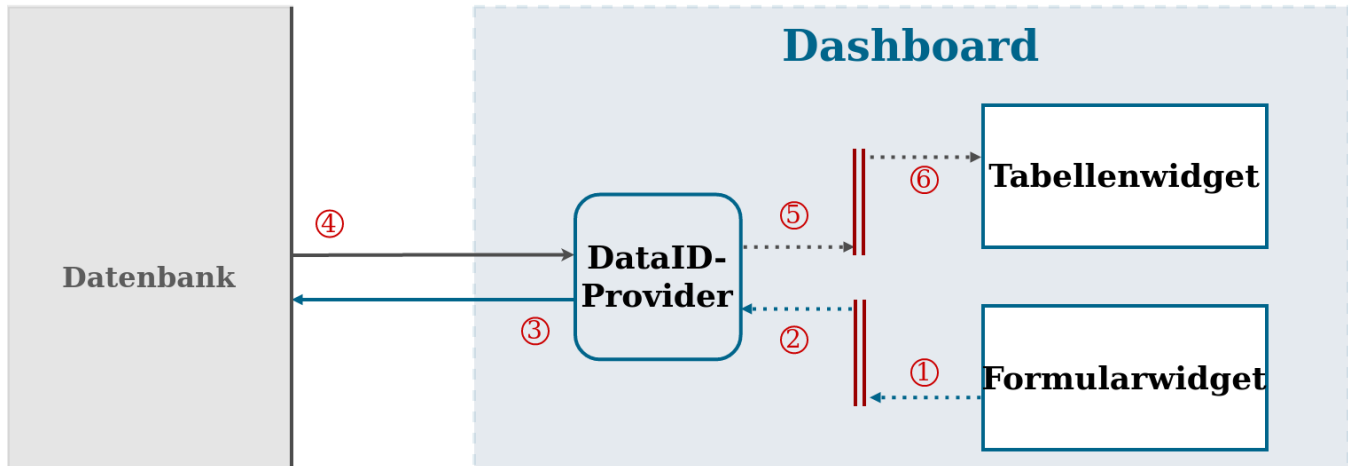


# DataID-Provider

Der DataID-Provider ermöglicht es eine DataID auszuführen und das Ergebnis in YUNA-Dashboards zu integrieren.

**Beispielgrafik:** Ausführung einer DataID gesteuert über ein Formularwidget und anschließende Darstellung in einem Tabellenwidget:



1. Bei Betätigung des Absenden-Buttons im Formularwidget wird die DataID an den Output-Channel des Formularwidgets übergeben.
2. Die durch den IO-Channel bereitgestellten Daten werden durch den DataID-Provider konsumiert, da dieser *IO-Channel als Input konfiguriert ist*.
3. Der DataID-Provider führt die DataID auf der Datenbank aus.
4. Die Datenbank liefert das Ergebnis der DataID an den DataID-Provider.
5. Das Ergebnis wird in den *Output-Channel* des DataID-Providers übergeben.
6. Das Tabellenwidget erhält die neuen Daten aus dem IO-Channel und kann diese darstellen.

## YUNAML Beispielkonfigurationen für DataID-Provider:

- i** Durch Klick auf die verschiedenen YUNAML-Elemente gelangen Sie zu den jeweiligen detaillierten Informationen in der folgenden Tabelle.  
*Kursive Elemente* beziehen sich auf andere Inhalte Ihrer YUNAML-Dashboards, wie z.B. Namen von DataIDs oder IO-Channels.

Beispiel für einen DataID-Provider

```
<io-provider>

  <type>DataId</type>

  <config>

    <dataId>qy_nameOfADataId</dataId>

    <triggerParams>

      <optional>OptionalInputChannel</optional>


    </triggerParams>

    <output>OutputChannel</output>

  </config>

</io-provider>
```

## Konfiguration des DataID-Providers

YUNAML-Tag	Beschreibung	Konfiguration erforderlich?	Beispiel
<b>type</b>	Der Typ des IO-Providers. Für den DataID-Provider muss dieser Parameter auf "DataID" gesetzt werden.  Weitere mögliche Werte sind "UrlParams" und "HTTP".	✓	<code>&lt;type&gt;DataId&lt;http&gt;</code>
<b>config</b>	Die verschiedenen Konfigurationsparameter des DataID-Providers.	✓	
<b>config &gt; dataid</b>	Die auszuführende DataID des DataID-Providers.	✓	<code>&lt;dataid&gt;dataidName&lt;/dataid&gt;</code>
<b>config &gt; triggerParams</b>	Um den Zeitpunkt für die Ausführung der DataID zu konfigurieren, werden die Trigger-Parameter in drei YUNAML-Tags definiert: <b>&lt;trigger&gt;</b> , <b>&lt;mandatory&gt;</b> und <b>&lt;optional&gt;</b> .	✗	<pre> &lt;triggerParams&gt;   &lt;trigger&gt;some   Channel&lt;   /trigger&gt;   &lt;mandatory&gt;   &lt;list&gt;mandatoryChannel&lt;   /list&gt; &lt;/mandatory&gt;   &lt;optional&gt;   &lt;list&gt;optionalChannel1&lt;   /list&gt;   &lt;list&gt;optionalChannel2&lt;   /list&gt; &lt;/optional&gt; &lt;/triggerParams&gt; </pre>
<b>config &gt; triggerParams &gt; trigger</b>	Bei jeder Änderung eines Channels, der in mindestens einem dieser Tags definiert ist, wird geprüft, ob eine Anfrage durchgeführt werden soll. Dabei werden folgende Bedingungen geprüft:	✗	
<b>config &gt; triggerParams &gt; mandatory</b>	<ol style="list-style-type: none"> <li>Nur wenn Channel in <b>&lt;trigger&gt;</b> definiert sind: Wurde ein <b>&lt;trigger&gt;</b>-Channel aktualisiert?</li> <li>Nur wenn Channel in <b>&lt;mandatory&gt;</b> definiert sind: Wurden alle <b>&lt;mandatory&gt;</b>-Channel mit Daten befüllt?</li> </ol>	✗	
<b>config &gt; triggerParams &gt; optional</b>	<p>Sind weder <b>&lt;trigger&gt;</b>- noch <b>&lt;mandatory&gt;</b>-Channel definiert, wird bei jeder Änderung eines in <b>&lt;optional&gt;</b> angegebenen Kanals eine Anfrage ausgeführt.</p> <p>Ein Channel kann sowohl <b>&lt;trigger&gt;</b> als auch <b>&lt;mandatory&gt;</b> sein.</p> <p>Sollen mehrere Channel in einem der drei Tags definiert werden, müssen diese in <b>&lt;list&gt;</b>-Tags angegeben werden.</p>	✗	
<b>config &gt; output</b>	Unter <b>&lt;output&gt;</b> wird konfiguriert, in welchen IO-Channel das Ergebnis der DataID veröffentlicht wird.	✗	<code>&lt;output&gt;OutputChannel&lt;/output&gt;</code>
<b>config &gt; converter</b>	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;">  Dieser Konfigurationsparameter sollte nur in wenigen Anwendungsfällen angepasst werden und ist daher für erfahrene Anwender vorgesehen. </div> <p>Definiert, in welchem Format das Ergebnis der DataID-Abfrage in den Output-Channel veröffentlicht wird.</p> <p>Es werden zwei Formate unterstützt: 'row' und 'col'. Für Details siehe das <a href="#">Beispiel zu &lt;converter&gt;-Formaten</a>.</p>	✗ Default: <b>row</b>	<code>&lt;converter&gt;col&lt;/converter&gt;</code>



## Beispiel zu <converter>-Formaten

Selektierte Daten:

ID	Name
1	Heinz
2	Sabine
3	Stefan

Die veröffentlichten Daten im 'row'-Format:

```
{
  "dataQueryResult": {
    "rows": [
      {"ID": 1, "Name": "Heinz"},
      {"ID": 2, "Name": "Sabine"},
      {"ID": 3, "Name": "Stefan"}
    ],
    "header": ["ID", "Name"]
  }
}
```

Die veröffentlichten Daten im 'col'-Format:

```
{
  "dataQueryResult": {
    "columns": {
      "ID": [1, 2, 3],
      "Name": ["Heinz", "Sabine", "Stefan"]
    },
    "header": ["ID", "Name"]
  }
}
```



## Unknown Attachment

### Beispiel

In diesem Beispiel wird über den DataID-Provider eine Data-ID ausgeführt. Das Ergebnis wird anschließend in einem Tabellen-Widget dargestellt.

```
<xml>
<view name="DataID-Provider" roles="System_Admin, AdHoc_Full_Issue">
  <caption>DataID-Provider</caption>
  <description>IO-Provider example</description>
  <userdocu>https://confluence.eoda.de/display/DD1/.IO-Provider</userdocu>

  <!-- DATA-ID IO-Provider -->
  <io-provider>
    <type>DataId</type>
    <config>
      <!-- define the DATA-ID -->
      <dataId>qy_dataid_provider_example</dataId>
      <triggerParams>
        <!-- define the parameters, which will trigger the DATA-ID -->
        <trigger>buttonPressed</trigger>
      </triggerParams>
      <!-- define the outputchannel, where the result from the DATA-ID will go -->
      <output>dataFromQuery</output>
    </config>
  </io-provider>

  <!-- Form-Widget for DATA-ID IO-Provider-->
  <widget>
```

```

<widgettype>formwidget</widgettype>
<caption>
  <show>true</show>
  <label>DATA-ID Form</label>
</caption>
<position>
  <x>0</x>
  <y>0</y>
</position>
<size>
  <x>2</x>
  <y>2</y>
</size>
<outputs>
  <!-- on submit, buttonPressed will trigger the DATA-ID -->
  <submit>buttonPressed</submit>
</outputs>
<submitButton>
  <label>Run Query!</label>
</submitButton>
<formTemplate>
  <![CDATA[
<div>
<p>Run query and show data</p>
</div>
]]>
  </formTemplate>
</widget>

<!-- Table-Widget for DATA-ID IO-Provider-->
<widget name="tbl_dataid_device_basic_info">
<widgettype>tabledirective</widgettype>
<caption>
  <show>true</show>
  <label>Data from DATA-ID IO-Provider</label>
</caption>
<position>
  <x>2</x>
  <y>0</y>
</position>
<size>
  <x>16</x>
  <y>4</y>
</size>
<sorting>ProductGroup</sorting>
<sortingorder>asc</sortingorder>
<inputs>
  <!-- get the data from the DATA-ID IO-Provider -->
  <!-- takes the data json in row format -->
  <data>dataFromQuery</data>
</inputs>
<generalOptions>
  <addColumnns>true</addColumnns>
  <selectable>>false</selectable>
</generalOptions>
</widget>
</view>
</xml>

```