Database Wizard List Data Store

New Feature
This is a new feature in **Joget DX 8**

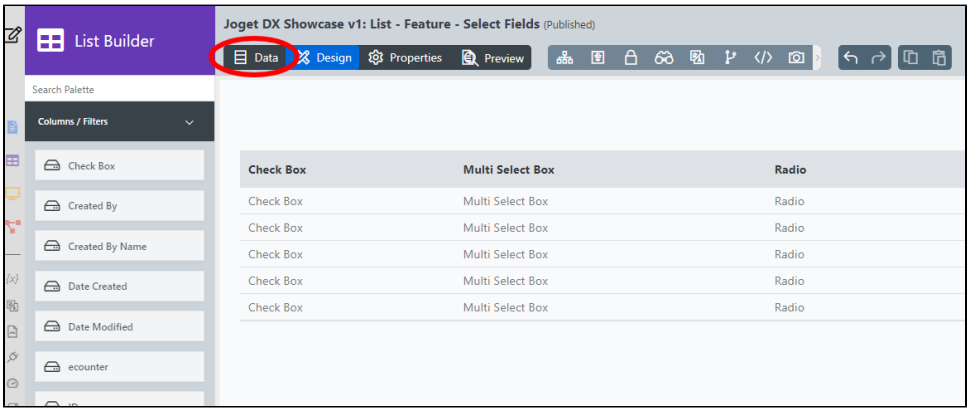
Introduction

Database Wizard for Loading to the List is a new feature introduced in Joget DX 8. **Database Wizard** allows the user to integrate external databases as a source in a Datalist Builder. Instead of dealing with SQL code, users can now visually configure database connections, queries and updates.

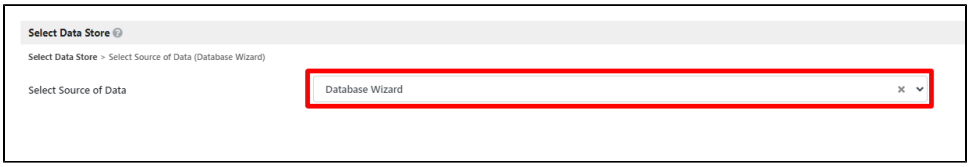
Get Started

To use the **Database Wizard** inside a Datalist, here are the steps:

1. Inside the Datalist Builder, click on the **Data** tab.

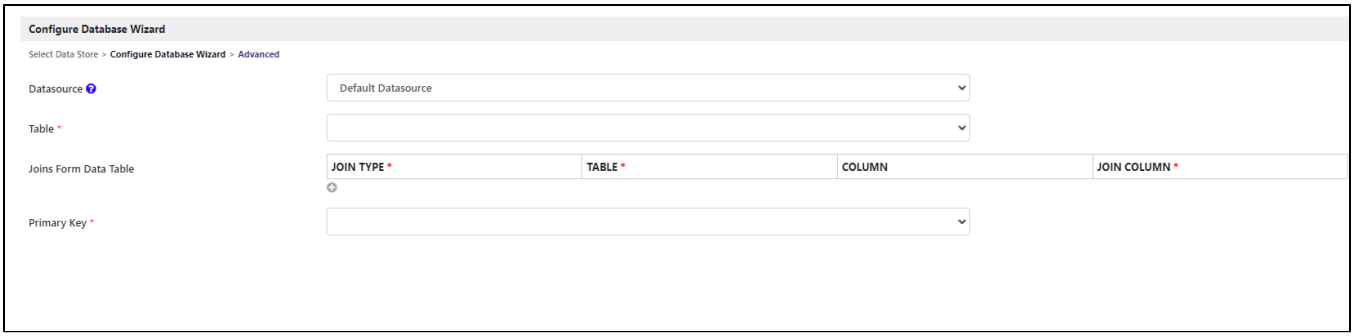


2. Select **Database Wizard** as the Source of Data.



3. Configure the **Database Wizard** accordingly.

Database Wizard Properties

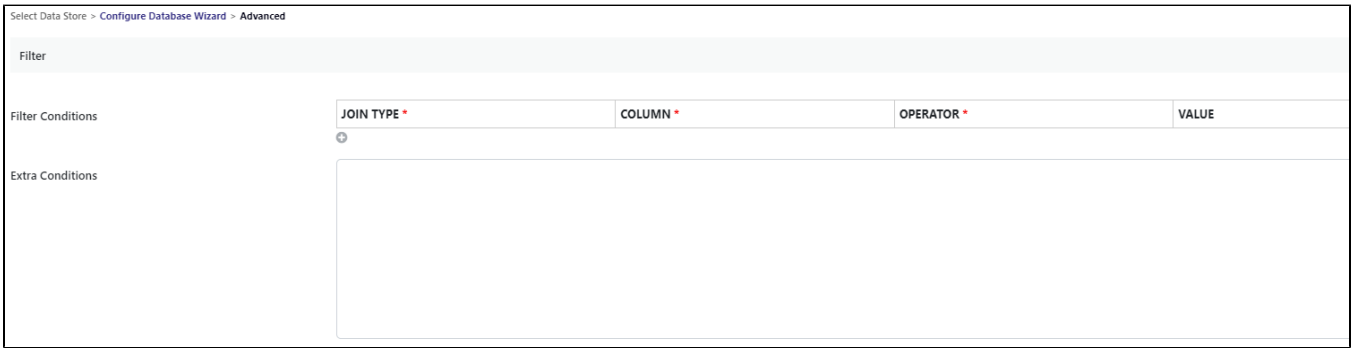


Name	Description

Datasource	• Custom Datasource - setup to connect to an external database, has an additional configuration. • Default Datasource- connect to the Joget database. By selecting **Default Datasource**, the database your Joget is currently using will be selected.
Custom JDBC Driver	Custom JDBC Driver. This field is required when **Custom Datasource** is selected in **Datasource** above. Example: `com.mysql.jdbc.Driver`
Custom JDBC URL	Custom JDBC URL. This field is required when **Custom Datasource** is selected in **Datasource** above. Example: `jdbc:mysql://localhost/jwdb?characterEncoding=UTF8&useSSL=false`
Custom JDBC Username	Custom JDBC Username. This field is required when **Custom Datasource** is selected in **Datasource** above.
Custom JDBC Password	Custom JDBC Password. This field is required when **Custom Datasource** is selected in **Datasource** above. **Test the connection parameters** Click on the "Test Connection" button at the bottom of the page to quickly test out your configurations.
Table	Choose a Table from the database as a source to retrieve data.
Joins Form Data Table	Joins Form Data Table **Field** — **Description** Join Type — Condition type to fulfil. Table — Target table to join with. Column — Field ID from source form to set as option label. Join Column — Field ID from source form to set as option grouping value.
Primary Key	Define the primary key column. By default, it should be "id".

Advance

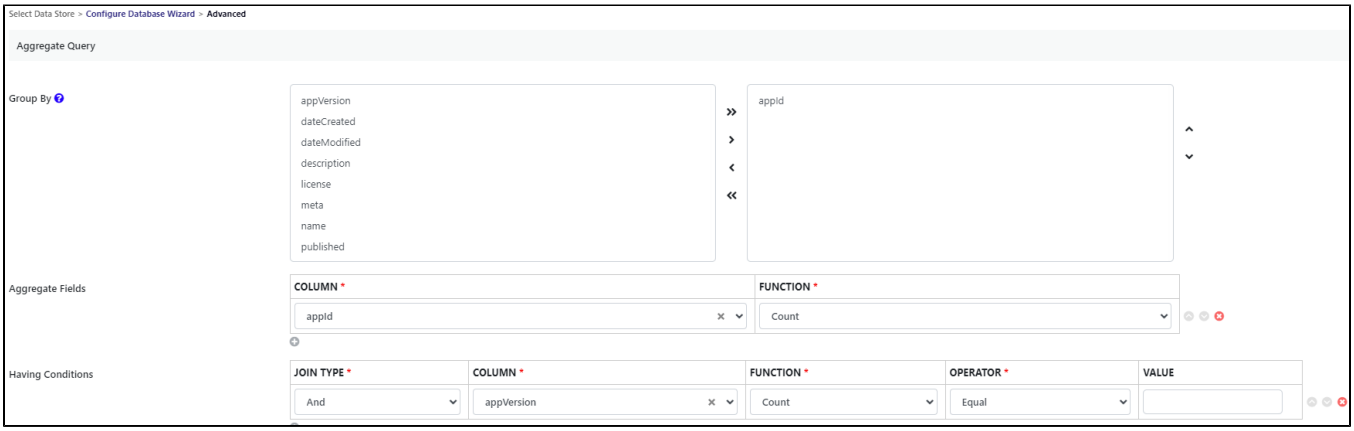
Filter



Name	Description

<table>
<tr><td>Filter Conditions</td><td>Filter Conditions
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Join Type</td><td>• And
• Or</td></tr>
<tr><td>Column</td><td>Table Column ID</td></tr>
<tr><td>Operator</td><td>• Equal
• Not Equal
• Greater Than
• Greater Than Or Equal
• Less Than
• Less Than Or Equal
• Like
• Not Like
• In
• Not In
• Is True
• Is False
• Is Null
• Is Not Null</td></tr>
<tr><td>Value</td><td>Filter value</td></tr>
</table></td></tr>
<tr><td>Extra Conditions</td><td>Additional condition(s) for filtering the data set. HQL is expected here.</td></tr>
</table>

Aggregate Query



Name	Description
Group By	Add **grouping** clause/function to the eventual data set. This can be used together with the **Aggregate Fields** above.
Aggregate Fields	This field will be displayed once any number of columns has been added to the **Group By** field. The select field is to aggregate. • Count • Count Distinct • Sum • Min • Max • Avg

Having Conditions	This field will be displayed once any number of columns has been added to the **Group By** field. The **HAVING** clause enables you to specify conditions that filter which **group** results appear in the final results. The WHERE clause places conditions on the selected columns, whereas the **HAVING** clause places conditions on **groups** created **by** the **GROUP BY** clause. Read more at http://www.dofactory.com/sql/having

Expression Columns

Name	Description
Expression Columns	An additional column can be added to this expression column using Hibernate Query Language (HQL). This is especially useful when you need to perform additional computation on multiple columns.