Maqetta: Visual Authoring of HTML5 User Interfaces

Maqetta GridX Configuration Wizard

tonyerwin | *June 28 2012* | *Visits (3851)*

Updated Sept 10, 2012 for Release 7.

Data-bound grids for the display of tabular data are a key building block of many applications. One option developers have to enable grids in their Dojo applications is the powerful GridX library. GridX features a compact core built for scalability and offers a plug-in architecture so modules for advanced features can be loaded on demand. It is hosted in its own GitHub repository and available under the same BSD/AFLv2 licensing as the Dojo Toolkit. Maqetta supports GridX 1.0.0 and includes an advanced GridX Configuration Wizard as a companion. Using the wizard, designers can easily bind a GridX widget to a data source, configure which columns to display, and set properties of the selected columns (e.g., order, width, label, etc.). This article will take a high-level walk through the process of adding a GridX widget to a page in Maqetta.

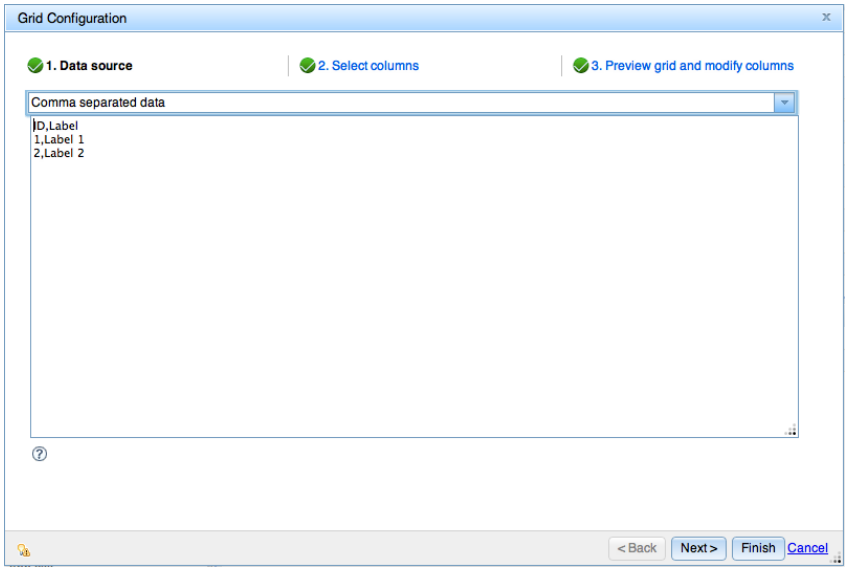
Adding GridX from Widgets Palette

GridX can be found in the Maqetta Widgets Palette in the *Dojo Controls* folder. When you drag GridX from the Widgets Palette onto your canvas, the GridX Configuration Wizard will be launched in a separate dialog.

Configuring the Data Source

The first panel of the wizard is devoted to configuring the data source from which your GridX will retrieve its data. There are three types of data sources that can be used. (For the technically minded, in the end no matter which option you choose, either `adojo.data.ItemFileReadStore` or `dojox.data.CsvStore` will be created on your behalf and associated with the GridX widget.)

- *Comma-separated data* -- With this option, the data to show in the GridX is entered directly into the text field using comma-separated values. The first line of data is used to define identifiers for your columns. The remaining lines should then contain the data for each row. Each data line should have a number of comma-separated values that match the number of columns on the first line.



- *Data file from workspace* -- Use this option if your data is in a file (in either JSON or CSV format) that has been uploaded to your Maqetta workspace. Click on the folder icon next to the text box to see a separate file navigator dialog that allows you to find the files of interest in your project. There are several sample files provided by default in the *samples/sample_data* directory.

About this blog

In this blog, Tony Erwin covers a range of topics related to Maqetta. Maqetta is an open source project that provides WYSIWYG visual authoring of HTML5 user interfaces (desktop and mobile).

Related posts

Maqetta Means Mock-u...
UpdatedJune 12 1 0

New course - Develop...
UpdatedApr 29 1 0

Resolving the virtua...
UpdatedApr 18 0 0

Creating your own UI...
UpdatedApr 18 3 7

Tips and tricks for ...
UpdatedMar 20 1 0

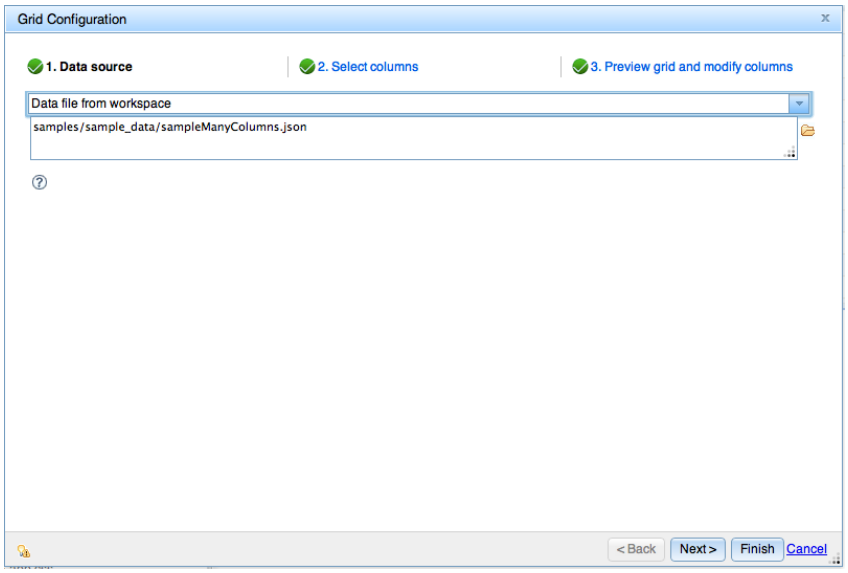
Tags

Find a Tag

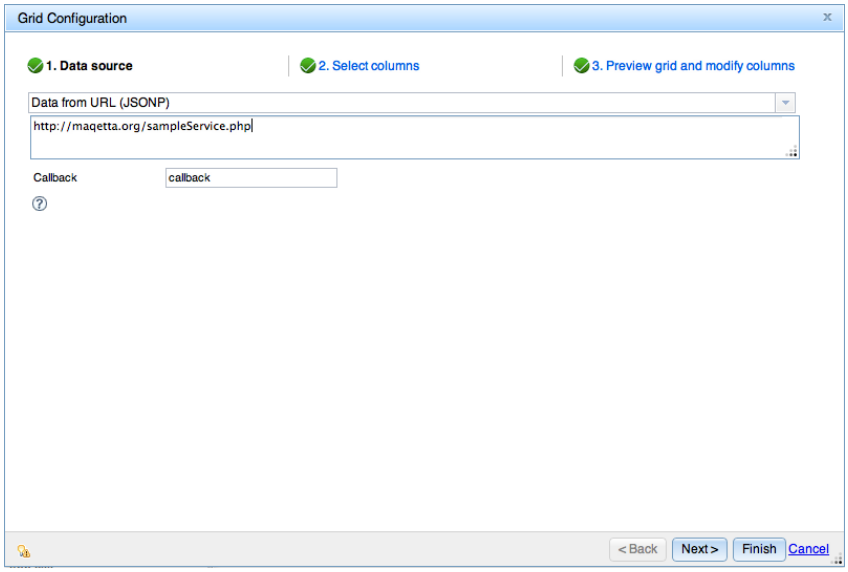
css design dijit dojo gridx html5 javascript maqetta mobile phonegap prototype rational_application_developer rational-application-developer welcome workflow

Cloud | List

Recent tweets

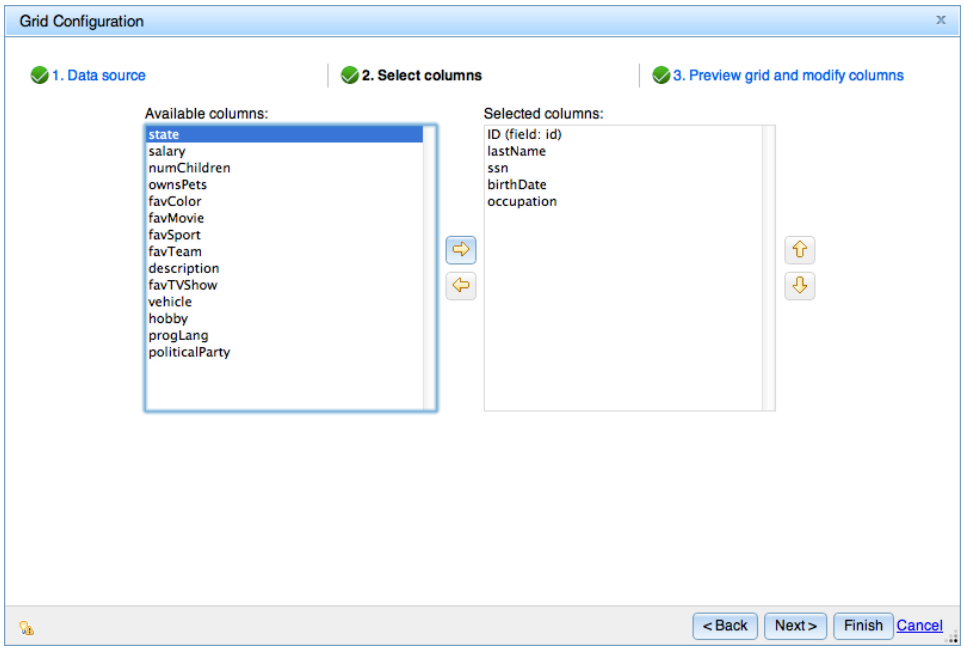


- *Data from URL (JSONP)* -- Use this option if you want the GridX to retrieve JSON data from a remote service. JSON with Padding (JSONP) is used to allow access to cross-domain resources. Two sample services are provided by Maqetta to make it easy to experiment with this feature: *http://maqettaservice.appspot.com/sampleservice* and *http://maqetta.org/sampleService.php* .



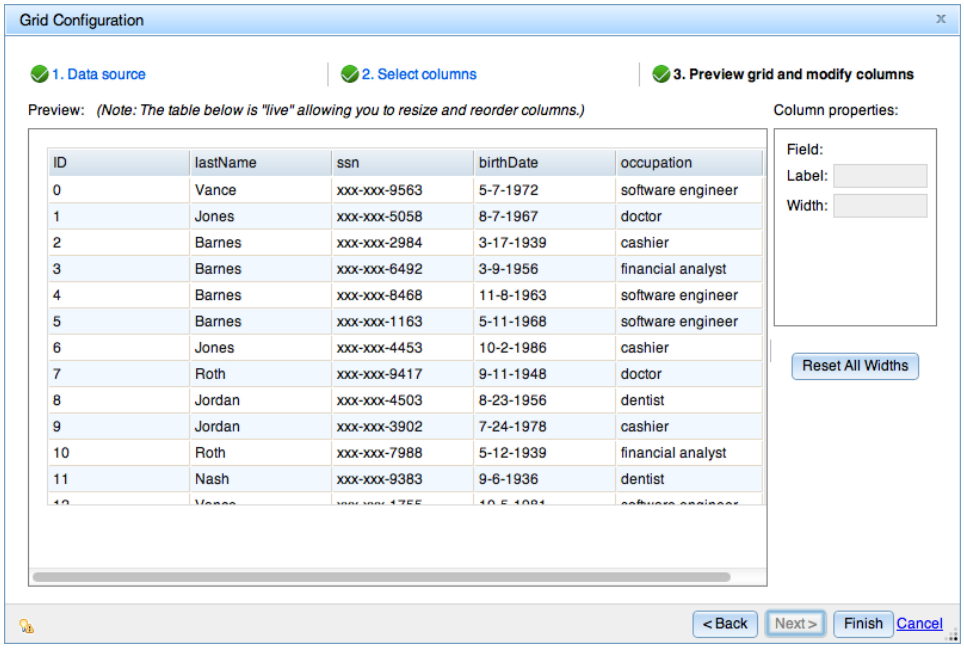
Choosing Which Columns to Display

If you click *Next*, the second panel of the wizard will be displayed. This panel presents the columns that are available from the data source in the *Available columns* list box on the left side, and allows you to choose which ones you want to show the end user by moving them to the *Selected columns* list box box on the right side. This is especially useful because many (if not most) data sources contain far more columns than you'd ever want to expose to an end user on their first viewing of a table. In the screen shot below, I had previously chosen the *samples/sample_data/sampleManyColumns.json* file (a sample available by default in your workspace). Using the right and left pointing arrows in the middle of the panel, I narrowed my list of selected columns from 19 to 5. In addition, I set the desired order of those columns using the up and down pointing arrows on the right side of the *Selected columns* list box.

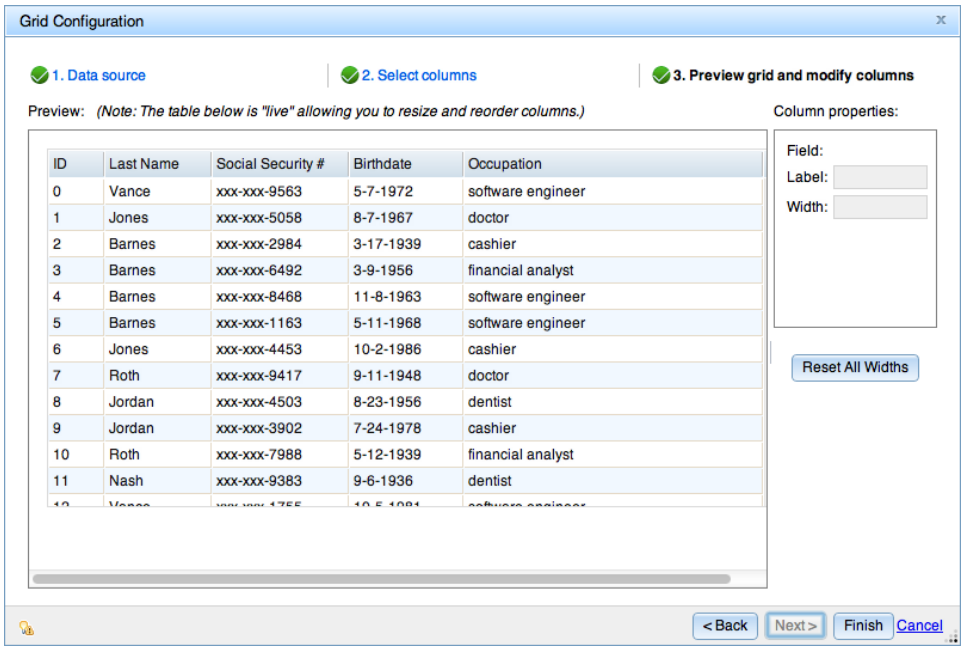


Previewing the Grid and Modifying Column Propertie

If you click *Next* at this point, you will go to the third (and last) page of the wizard. This panel provides a live preview of the GridX widget based on your selections in the first two panels. Any changes made on this panel are reflected immediately within the preview and are persisted when you complete the wizard. The screen shot below shows the initial state of the preview panel using the five columns I selected above. By using the column headers in the preview, you can move and resize the columns directly within the GridX widget. In addition, if you select a column (using the column header), the fields in the *Column properties* section are enabled and you can use them to change column labels and widths.



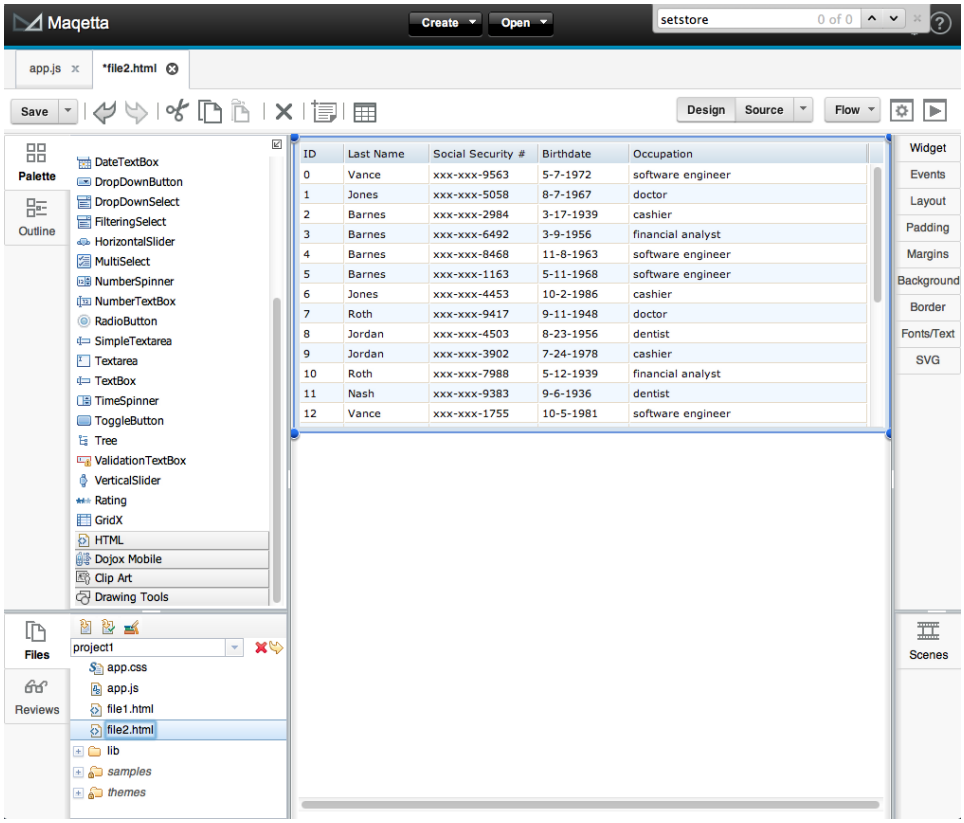
The next screenshot shows the state of things after all of the column labels have been changed to human readable strings (rather than the ids provided in the data source) and all of the columns have been resized to my desired widths.



ID	Last Name	Social Security #	Birthdate	Occupation
0	Vance	xxx-xxx-9563	5-7-1972	software engineer
1	Jones	xxx-xxx-5058	8-7-1967	doctor
2	Barnes	xxx-xxx-2984	3-17-1939	cashier
3	Barnes	xxx-xxx-6492	3-9-1956	financial analyst
4	Barnes	xxx-xxx-8468	11-8-1963	software engineer
5	Barnes	xxx-xxx-1163	5-11-1968	software engineer
6	Jones	xxx-xxx-4453	10-2-1986	cashier
7	Roth	xxx-xxx-9417	9-11-1948	doctor
8	Jordan	xxx-xxx-4503	8-23-1956	dentist
9	Jordan	xxx-xxx-3902	7-24-1978	cashier
10	Roth	xxx-xxx-7988	5-12-1939	financial analyst
11	Nash	xxx-xxx-9383	9-6-1936	dentist

Final Result

Finally, after clicking *Finish*, the dialog containing the wizard will go away and you will be left with a fully configured GridX widget in the Maqetta page editor. From there, you can continue designing the rest of your application. And, if later on you wish to modify the data source and/or columns used by the GridX widget, simply double-click on the GridX widget to invoke the GridX Configuration Wizard again. The wizard will be pre-populated with your current settings making it easy to tweak your design.



ID	Last Name	Social Security #	Birthdate	Occupation
0	Vance	xxx-xxx-9563	5-7-1972	software engineer
1	Jones	xxx-xxx-5058	8-7-1967	doctor
2	Barnes	xxx-xxx-2984	3-17-1939	cashier
3	Barnes	xxx-xxx-6492	3-9-1956	financial analyst
4	Barnes	xxx-xxx-8468	11-8-1963	software engineer
5	Barnes	xxx-xxx-1163	5-11-1968	software engineer
6	Jones	xxx-xxx-4453	10-2-1986	cashier
7	Roth	xxx-xxx-9417	9-11-1948	doctor
8	Jordan	xxx-xxx-4503	8-23-1956	dentist
9	Jordan	xxx-xxx-3902	7-24-1978	cashier
10	Roth	xxx-xxx-7988	5-12-1939	financial analyst
11	Nash	xxx-xxx-9383	9-6-1936	dentist
12	Vance	xxx-xxx-1755	10-5-1981	software engineer

Addendum

For an example of using GridX and custom JavaScript in Maqetta, see my subsequent posting on *Adding Your Own JavaScript to Maqetta Prototypes*.

Tags: *maqetta gridx dojo*

Add a Comment | More Actions

Comments (0) *Add a Comment* | *More Actions*

Previous Entry | Main | Next Entry