

Evidence based research of JavaScript Grids according to a set of expected attributes

Darío Macchi

Abstract. Tabular presentation of data is used in almost every website to place results in an organized way, making easy the dialogue between text and exact numbers. There are many JavaScript grids implementations, some of them dedicated to accomplish specific goals and others for general use. This study identifies which are the grids that have some attributes defined apriori by the team and do not pretend to be an exhaustive study about JavaScript grids. Although there are others grids that partially fits, DataTable, dhtmlxGrid, jqxGrid and KendoUI were selected as the ones that fill our requirements.

Keywords: JavaScript, grids, attributes, research, technology, internet

1 Introduction

The representation of data in a table is formally referred to as “tabular presentation.” Tabular presentation of data allows data to be organized for further analysis, allows large amounts of raw data to be sorted and reorganized in a neat format, and allows the inclusion of only the most important or relevant data. It also facilitates a dialogue between the text and the exact numbers in your results, so that you don't have to describe all the specific numerical values in your report. On the other hand, you should never put data in a table if you can describe it efficiently in one or two sentences. In summary, tabular presentation lets you place your results in an organized display of rows and columns that enable you to group your data by different classifications so that you can make comparisons and better understand your data [1].

Specifically talking about javascript grid implementations we had the suspicion that exists many of them, each one with some features that make them unique. Roberto Bicchiearai make a good classification of the grids that he have founded [2]. These are the classification and the criteria for each one.

Data driven: these are generally components with all the features needed for listing a large number of rows, with server-side pagination, scrolling, search and sort functionality; generally are lacking editing functionality, or when the feature is present is single row based, reflecting the fact that there is an underlying database. Data binding is made through XML or JSON, depending on the language supported server-side

Light edit: here the focus is given to the “editing agility”, every row is always in editing status, search and sort are rarely available and in any case working on a limited set of rows isn't a real limitation. Data is filled starting from a JS object, calling methods, or from an html table. Some of these implementation supply callback methods that can be used to interact server side.

Spreadsheets: oh yes! There are also some really sophisticated JS clones of the glorious one! These are generally complex solution (in terms of dependencies), supporting formulas, graphs and lot of functionalities. Here the focus is the “simulation effect”, data management and server side integration are really in background.

The goal of this work is to find the best (or most suitable) grid according to some attributes list defined by the team and not pretend to be an exhaustive research about javascript grid implementations or anything outside the defined scope.

The next section will introduce the research method followed to reach our results; section 3 will show the results; section 4 are the conclusions and final section are the references used to build the study.

2 Research methodology

The team wrote a set of attributes that describe what a grid needs to have in order to be a “good choice” in several different scenarios. The definition of attributes was made from some StackOverflow discussion about jQuery grid recommendations [3]. Although this questions and discussion were over jQuery, many other implementations were named and many of attributes discussed were took in consideration to make the final list. Also the question had 188771 visits in the last 3 years, 353 positive votes, 292 people marked it as favorite question and the preferred answer itself had 267 positive votes making it a very active question and a reference in this topic.

These are the attributes that we will use in this research:

- Filtering
- Grouping
- Sorting
- Data: column types & data binding
- Editable cells
- Big data support (loading while scrolling)
- Good documentation and strong community
- Free (at least in some cases or with some license limits)

2.1 Research questions

The high-level objective is divided into 4 concrete research questions (RQ) in order to obtain a more detailed and comprehensive view on this topic.

- RQ.1. What are the grids founded according to the attributes defined?
- RQ.2. What have the most active community?
- RQ.3. What is the update frequency?

2.2 Source selection and definition of search strategy

Google was used as search engine to simulate the activity of any practitioner looking for javascript grids. Any date restrictions was applied and the Google verbatim tool [4] was used to search without any “synonymization”, spell-correction, personalization or other interpretation. It's just a basic search without any alteration of string typed into the query box. Throughout this section, we'll use square brackets [] to signal a search query [5].

The search strategy was structured and built using the PICOC framework [6] as is showed in Table I.

Table I - PICOC framework used in this study

	Population	Intervention	Comparison	Outcome	Context
Main keyword	Javascript grid	The most number of attributes		Comprehensive list	Web applications
Synonym	Javascript plugin for tables	Most robust			

The main strategy is to perform iterative searches in Google to improve domain knowledge and word variations to build the main search string. Also, some comparison need to be done with some lists of javascript grids maiden before by other authors ([3], [7]) as a quality assurance activity. Based on PICOC, the main search string is build following these steps:

1. The search string must start with [javascript + (grid OR table plugin)] according to the population.
2. Attributes must be the next to add to the string. The problem is that some of these attributes are not well defined to be used as a part of a search string (i.e big data support) so others searches were performed in this way:
 - a. Start with [javascript + (grid OR table plugin)]
 - b. Next add some attribute to the string (i.e big data support)
[javascript + (grid OR table plugin) + (big data support)]
 - c. Next, if the results weren't so good, replace the attribute with some synonym until the results are ok (i.e (big OR large) data).
[javascript + (grid OR table plugin) + ((big OR large) data)]
 - d. Finally add the string that return the best partial result to the main string and continue with the next attribute.

The resulting search string was: [javascript + (grid OR table plugin) + (filtering OR grouping OR sorting OR ("column types" + (data + (binding OR source)))) OR editable OR (big OR large) data]

2.3 Study selection

For this study we worked only with “data driven” and “light edit” grids according to Bicchiearai’s classification. Moreover the attributes defined, the paging attribute was used as a very simple parameter to get rid of some grid or include it in the study (if the grid can't paginate, it's out of study).

One constraint in the search string definition was that the result set had to be of a manageable size, but still had to reach the maximum possible coverage. Because Google’s PageRank algorithm rank relevant results higher [8], the first 20 results of the search were selected.

2.4 Data extraction and synthesis

The extraction method was very simple. The first 20 results were opened and then read, looking for javascript grid references. The reference may be in the page body, in banners related with the content (i.e ads) or in the comments of the page. Every finding was written in a spreadsheet avoiding

duplicates for later process, but counting every time each grid reference appears. Once all grid references are extracted from every page included in the study, every grid website was visited. In this stage it's important to know what of the referenced grids satisfied most of the attribute defined earlier. For these reason every grid feature list was read to see which attribute of the defined list was supported by each one. Finally the results are filtered by those that have all attributes and ordered by frequency of appearance in the result pages.

In order to answer the second question (RQ.2) about the community activity, each Forum/Support Group was visited looking for the number of discussions or topics, discarding the number of posts/messages inside each one. For this study we only take care about the quantity of post/message and not about of the quality of them.

To answer the third question (RQ.3) about the update frequency of each javascript grid solution, the team had to send an email to each company/author/support team asking for this information. The reason for this is because the information about the history of releases wasn't online (or at least, the team didn't find it).

3 Results

The answer to the first questions (RQ.1) can be one of the following three tables. Table II shows the grids that support all the desired attributes by itself (without plugins) and have the **best** community.

Table II - Grids with all desired attributes (without plugins, best community)

Freq (#)	Grid	Forum / Group	StackOverflow (tagged questions)	StackOverflow (search results)
7	jqGrid	5398	3865	3743
4	SlickGrid	1021	267	339
3	Ext Js Grid	12073	293	1533
1	KendoUi	2619	101	72

The next table (Table III) shows the grids that support all the desired attributes by itself (without plugins) as the first one, but this time showing the ones that have **good** community.

Table III - Grids with all desired attributes (without plugins, good community)

Freq (#)	Grid	Forum / Group	StackOverflow (questions tagged)	StackOverflow (search results)
7	jqGrid	5398	3865	3743
6	dhtmlxGrid	4483	0	38
4	SigmaGrid	918	0	5
4	SlickGrid	1021	267	339
3	Ext Js Grid	12073	293	1533
2	jqxGrid	107	0	0
1	KendoUi	2619	101	72

The last result may be the most applicable because it shows the grids that support all the attributes (with or without plugins). Plugins extends the behaviour of some grids but may lead to compatibility

issues (some plugin version may or may not work with newer grid versions). However in this study, the necessary plugins seems to have the support of the community (even they are listed in the official page of the grid) so we have included it as if the grid support it natively. About community, it seems to have good ones (and if you compare it with the Table II, better one).

Table IV - Grids with all desired attributes (with plugins)

Freq (#)	Grid	Grouping	Editable	Forum / Group	StackOverflow (questions tagged)	StackOverflow (search results)
12	DataTables (Data Table jQuery plugin)	X.plugin	X.plugin	9692	837	Inf.
7	jQGrid	X	X	5398	3865	3743
6	dhtmlxGrid	X	X	4483	0	38
4	SigmaGrid	X	X	918	0	5
4	SlickGrid	X	X	1021	267	339
3	Ext Js Grid	X	X	12073	293	1533
2	jqxGrid	X	X	107	0	0
1	KendoUi	X	X	2619	101	72

The answer of the next question, **RQ.2** it's about the communities around each grid solution. We have found that **Ext Js Grid** have the most numerous forum/group with 12073 posts/messages. About StackOverflow tagged questions and search results, **jQGrid** won it with 3865 questions with jQgrid tag and 3743 search results.

Finally, the last question (**RQ.3**) about the average number of updates per year of each solution we received only 4 answers (50% of Table IV results) as we show below in Table V.

Table V - Updates per year

Grid	Updates/year	Computation
DataTable	13	51 non-beta releases / 4 years
dhtmlxGrid	2	None. Explicit in email response.
jqxGrid	12	None. Explicit in email response.
KendoUi	6	None. Explicit in email response.

4 Conclusions

Tabular representation has many advantages over other kind of data representation, like better organization, numerical results and large amount of raw data display. For these reason many javascript implementations are available in the web, with different features, licensing modes and support. In this study a set of attributes were defined in order to find what grid fit into this features description.

According to our research we have found that 8 grids fit our requirements, some of them directly and some with plugins help. Their communities seem to be very active, mainly through their forums or groups. Some of them also are very active in StackOverflow, having a lot of tagged questions and search results.

The number of updates per year wasn't so easy to get as we initially thought. We get this information through emails contact, but we think that this information should be public and available in each grid website. This information let you know the development history of the grid, the versioning schema and more important, how often you need to make updates if you want to stay updated.

As we said, this study doesn't pretend to be an exhaustive research about javascript grid implementations, but we think that this may be the base of new researches about this topic. Also, the methodology used in this study can be use in similar scenarios. Future works may include some empirical activities to test which have better performance, which is easier to implement or which grid is easier to extend through plugins.

5 References

- [1] "Designing Tables," *North Carolina State University - LabWrite resources*. [Online]. Available: <http://www.ncsu.edu/labwrite/res/gh/gh-tables.html>.
- [2] R. Bicchierai, "JavaScript grid editor: I want to be Excel," *Eltit Golb*, 2010. [Online]. Available: <http://roberto.open-lab.com/2010/01/30/javascript-grid-editor-i-want-to-be-excel/>.
- [3] Wilco, "jQuery Grid Recommendations," *StackOverflow*, 2008. [Online]. Available: <http://stackoverflow.com/questions/159025/jquery-grid-recommendations>.
- [4] Google, "Google verbatim tool." [Online]. Available: <http://support.google.com/websearch/bin/answer.py?hl=en&answer=1734130>. [Accessed: 08-May-2012].
- [5] Google, "Google Basic search help." [Online]. Available: <http://support.google.com/websearch/bin/answer.py?hl=en&answer=134479>.
- [6] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," vol. 2, no. EBSE 2007-001, pp. 2007-01, 2007.
- [7] K. Liew, "7 Robust and Feature Packed Javascript Grid Plugins," 2011. .
- [8] Wikipedia-Contributors, "PageRank," *Wikipedia*. [Online]. Available: <http://en.wikipedia.org/wiki/PageRank>.