

Pagination mit AJAX und mehreren Visualisierungen

== Code ==

Quellcode

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4. <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
5. <title>
6. Paging using AJAX with refresh of Chart (Google Visualization API)
7. </title>
8. <script type="text/javascript" src="http://www.google.com/jsapi"></script>
9. <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.3/jquery.min.js"></script>
10. <script type="text/javascript">
11. google.load('visualization', '1', {packages: ['table','barchart']});
12. </script>
13. <script type="text/javascript">
14. /**
15.  * draw partly
16.  *
17.  * @param visualisation ob can be type of google.visualization.BarChart
18.  * @param DataTable data the original full data
19.  * @param DataTable datacopy a reference with existing columns and a fixed row length
20.  * @param integer offset data offset
21.  * @param integer limit data limit
22.  */
23. function drawPartly(ob, data, datacopy, offset, limit) {
24.     var columns = data.getNumberOfColumns();
25.     // reset the datacopy (important, if switching from empty to full page)
26.     datacopy.removeRows(0,datacopy.getNumberOfRows());
27.     for(var i=offset; i<offset+limit && i<data.getNumberOfRows(); i++) {
28.         // insert new row
29.         var k = datacopy.addRow();
30.         // fill columns
31.         for(var j=0; j<columns; j++) {
32.             datacopy.setCell(k, j, data.getValue(i,j));
33.         }
34.     }
35.     ob.draw(datacopy);
36. }
37. /**
38.  * draws the visualization, called via google.setOnLoadCallback
39.  */
40. function drawVisualization() {
41.     // how many items should be visible per page
42.     var limit = 4;
43.     // preload one more page
44.     var firstload_entries = (limit*2);
45.     // set the first two pages as preloaded (index is important)
46.     var cachedpages = [1,1];
47.     // Create and populate the data table.
48.     var data = new google.visualization.DataTable();
49.     data.addColumn('string', 'Item');
50.     data.addColumn('number', 'Rating');
51.     data.addColumn('boolean', 'Check');
```

```

60. // construct the (global) data copy
62. var datacopy = data.clone();
63. // the copy will never hold more entries than one page can display
64. datacopy.insertRows(0, limit);
66. // fill the first entries with sample data
67. data.addRows(firstload_entries);
68. for(var i=0; i<firstload_entries; i++) {
69. data.setCell(i, 0, 'Item '+i);
70. data.setCell(i, 1, i*i);
71. data.setCell(i, 2, true);
72. }
73. // Create and draw the visualization.
75. var visualization = new google.visualization.Table(document.getElementById('table'));
76. google.visualization.events.addListener(visualization, 'page', function(e) {
77. var next = e['page']+1;
79. // update the chart
80. drawPartly(chart, data, datacopy, e['page']*limit, limit);
82. // preload next page if now existing yet
83. if(typeof(cachedpages[next]) != 'undefined') return;
85. // load data via json
86. $.getJSON("status.php", {page:next, limit:limit}, function(jsondata){
87. // specify the index where rows should be inserted
88. data.insertRows(next*limit, limit);
89. // insert data
91. $.each(jsondata, function(i,item) {
92. for(var j=0; j<item.length; j++) {
93. data.setCell(parseInt(i), j, item[j]);
94. }
95. });
96. // redraw the current page (activates the next button)
98. visualization.draw(data, {page:'enable', pageSize: limit, startPage: e['page']});
100. // save as cached
101. cachedpages.push(1);
102. });
103. });
105. // Create the visualization bar.
106. var chart = new google.visualization.BarChart(document.getElementById('chart'));
107. drawPartly(chart, data, datacopy, 0, limit);
109. visualization.draw(data, {page:'enable', pageSize: limit});
110. }
112. google.setOnLoadCallback(drawVisualization);
113. </script>
114. </head>
115. <body style="font-family: Arial;border: 0 none;">
116. <h1>Paging using AJAX with reloading of Chart</h1>
117. <h2>Google Visualization API</h2>
118. <p>Reloads additional data and appends it to the table</p>
119. <p>Updates two visualizations when using pagination.</p>
120. <a href="http://www.easy-coding.de">http://www.easy-coding.de</a>
122. <div id="table"></div>
123. <div id="chart" style="width:400px;height:200px"></div>
124. </body>
125. </html>

```

Alles anzeigen

== Demo ==

demo.easy-coding.de/google-vis...-ajax-visualizations.html

easy-coding.de/Attachment/565/...a283d0a35647cbab39d413e2c