

Pagination über mehrere 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 (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']});
12. </script>
13. <script type="text/javascript">
14. /**
15.  * draws the visualization, called via google.setOnLoadCallback
16.  */
17. function drawVisualization() {
18. // how many items should be visible per page
19. var limit = 4;
20. // preload one more page
21. var firstload_entries = (limit*2);
22. // set the first two pages as preloaded (index is important)
23. var cachedpages = [1,1];
24. // Create and populate the data table.
25. var data = new google.visualization.DataTable();
26. data.addColumn('string', 'Item');
27. data.addColumn('number', 'Rating');
28. data.addColumn('boolean', 'Check');
29. // fill the first entries with sample data
30. data.addRows(firstload_entries);
31. for(var i=0; i<firstload_entries; i++) {
32. data.setCell(i, 0, 'Item '+i);
33. data.setCell(i, 1, i*i);
34. data.setCell(i, 2, true);
35. }
36. // Create and draw the visualization.
37. var visualization = new google.visualization.Table(document.getElementById('table'));
38. google.visualization.events.addListener(visualization, 'page', function(e) {
39. var next = e['page']+1;
40. // preload next page if now existing yet
41. if(typeof(cachedpages[next]) != 'undefined') return;
42. // load data via json
43. $.getJSON("status.php", {page:next, limit:limit}, function(jsondata){
44. // specify the index where rows should be inserted
45. data.insertRows(next*limit, limit);
46. // insert data
47. $.each(jsondata, function(i,item) {
48. for(var j=0; j<item.length; j++) {
49. data.setCell(parseInt(i), j, item[j]);
50. }
51. });
52. });
```

```

62. // redraw the current page (activates the next button)
63. visualization.draw(data, {page:'enable', pageSize: limit, startPage: e['page']}));
64. // save as cached
65. cachedpages.push(1);
66. });
67. });
68. });
69. visualization.draw(data, {page:'enable', pageSize: limit});
70. }
71. }
72. google.setOnLoadCallback(drawVisualization);
73. </script>
74. </head>
75. <body style="font-family: Arial;border: 0 none;">
76. <h1>Paging using AJAX</h1>
77. <h2>Google Visualization API</h2>
78. <p>Reloads additional data and appends it to the table</p>
79. <a href="http://www.easy-coding.de">http://www.easy-coding.de</a>
80. <div id="table"></div>
81. </body>
82. </html>

```

Alles anzeigen

== Demo ==

demo.easy-coding.de/google-vis...on/paging-using-ajax.html