

```

1 <?php
2
3 namespace Helper;
4
5
6 class ExportHelper
7 {
8     public static function resultToCSV($results)
9     {
10         $rowCounter = 0;
11
12         header('Content-Type: text/csv');
13
14         $echo = 'Nr,Time,';
15         foreach($results['average'] as $key => $value)
16             $echo .= $key.',';
17         $echo = substr($echo,0,-1);
18         $echo .= "\n";
19
20         $echo .= $rowCounter.',Average,';
21         foreach($results['average'] as $key => $value)
22             $echo .= Self::microtimeToSec($value,false).',';
23         $echo = substr($echo,0,-1);
24         $echo .= "\n";
25         $rowCounter++;
26
27         foreach($results['series'] as $serie)
28         {
29             $echo .= $rowCounter.','.Self::microtimeToTime($serie['startTime']).',';
30             foreach($serie['results'] as $result)
31                 $echo .= Self::microtimeToSec($result['value'],false).',';
32             $echo = substr($echo,0,-1);
33             $echo .= "\n";
34             $rowCounter++;
35         }
36         echo $echo;
37     }
38
39     public static function resultToHTML($results)
40     {
41         $rowCounter = 0;
42
43         echo '<html>';
44         echo '<head>';
45         echo '<title>Benchmarck</title>';
46         echo '<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-MCw98/SFngE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPM0"
crossorigin="anonymous">';
47         echo '</html>';
48         echo '<body>';
49         echo '<div class="container">';
50         echo '<table class="table">';
51
52         echo '<tr>';
53         echo '<th>Nr</th>';
54         echo '<th>Time</th>';
55         foreach($results['average'] as $key => $value)
56             echo '<th>'.$key.'</th>';
57         echo '</tr>';
58
59         echo '<tr>';
60         echo '<td>'.$rowCounter.'</td>';
61         echo '<td>Average</td>';
62         foreach($results['average'] as $key => $value)
63             echo '<td>'.Self::microtimeToSec($value,false). '</td>';
64         echo '</tr>';
65         $rowCounter++;
66
67         foreach($results['series'] as $serie)
68         {
69             echo '<tr>';
70             echo '<td>'.$rowCounter.'</td>';
71             echo '<td>'.Self::microtimeToTime($serie['startTime']). '</td>';
72             foreach($serie['results'] as $result)
73                 echo '<td>'.Self::microtimeToSec($result['value']). '</td>';

```

```
74         echo '</tr>';
75         $rowCounter++;
76     }
77
78     echo '</table>';
79     echo '</div>';
80     echo '</body>';
81 }
82
83 public static function microtimeToTime($microtime)
84 {
85     $microseconds = sprintf("%03d", ($microtime - floor($microtime)) * 1000000);
86     return date('H:i:s.'.$microseconds, $microtime);
87 }
88
89 public static function microtimeToSec($microtime, $asString = true)
90 {
91     return round($microtime * 1000, 3) . ($asString ? " ms" : '');
92 }
93
94 public static function resultToJSON($results)
95 {
96     $print = ['Average' => $results['average']];
97
98     foreach($results['series'] as $serie)
99     {
100         $serieData = [];
101         foreach($serie['results'] as $test)
102             $serieData[$test['name']] = $test['value'];
103
104         $print[$serie['startTime']] = $serieData;
105     }
106
107     echo json_encode($print);
108 }
109 }
```