

WebPascal script model tutorial - Js parsing dataset

2017-06-07 22:58:07 18

JSON(JavaScript Object Notation) It is a lightweight data exchange format that is widely used.

Through the previous tutorial, we know that there is a DBQueryToJSON function in the script model, which can directly output the data in the database as JSON formatted string, but in order to facilitate the Delphi developers to easily convert the JSON into a Dataset object, it retains The characteristics of rtc json, js can not directly identify this json string, but we can easily identify it with js, the following is the demonstration code:

```
1  <?
2  // MSAccess
3  $s:=DBConn('mydb','DriverID=MSAcc;Database=.\\db.mdb');
4
5  UnicodeToAnsi(DBQueryToJSON('mydb','select * from tuser'));
6  ?>
```

```

1  <!DOCTYPE html>
2  <html lang="zh-CN">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8      <script src="http://cdn.bootcss.com/jquery/3.1.1/jquery.min.js"></script>
9
10     <script type="text/javascript">
11         function test(){
12             $.ajax({
13                 url: "db.api",
14                 headers:{},
15                 type:"GET",
16                 success: function(result){
17                     $("#aaa").html(result);
18                 },
19                 dataType: "text"
20             });
21             $.ajax({
22                 url: "db.api",
23                 headers:{},
24                 type:"GET",
25                 success: function(result){
26                     var parsedJson = jQuery.parseJSON(
27                         result.replace(/\\\/dsfields/, 'dsfields').replace(/\\\/dsrows/, 'dsrows'));
28                     $("#bbb").html("");
29
30                     $.each(parsedJson.dataset.dsfields, function(i, item) {
31                         $("#bbb").append("name:"+item.name);
32                         if (item.type=="W")
33                             {"#bbb").append(" type:string");}
34                         else if (item.type=="I")
35                             {"#bbb").append(" type:integer");}
36                         else if (item.type=="WM")
37                             {"#bbb").append(" type:note");}
38                         if (item.size!=undefined){$("#bbb").append(" size:"+item.size);}
39                         {"#bbb").append("<hr/>");
40                     });
41
42                     $.each(parsedJson.dataset.dsrows, function(j, item) {
43                         var count = 0;
44                         $.each(parsedJson.dataset.dsrows[j], function(i, item) {
45                             {"#bbb").append(item);
46                             if
47                             (++count!=parsedJson.dataset.dsrows[j].length){
48                                 {"#bbb").append(",");
49                             }
50                         });
51                         {"#bbb").append("<hr/>");
52                     });
53                 },
54                 dataType: "text"
55             });
56         }
57     </script>
58 </head>
59 <body>
60     <button type="button" onclick="test();">test</button>
61     <br>
62     <div id="aaa">...</div>
63     =====
64     <div id="bbb">...</div>
65 </body>
66 </html>

```

do you understand? Yes, just remove "\/", is it very simple?

Of course, in the normal development process, it is not recommended to deal with this, because the browser restrictions are more, you go through js to deal with, the efficiency is very low, you should deal with in the script, such as using DBQuery function to obtain data, The content of the loop output stitching is a JSON string, so the efficiency is much higher.

Attached effect chart:

← → ↻ ⓘ localhost:8833/www/js解析dataset数据集/test.api

测试

```
{ "result": "success", "dataset": { "Vdsfields": [ { "name": "FID", "type": "I" }, { "name": "FUser", "type": "W", "size": 50 }, { "name": "FPwd", "type": "W", "size": 50 }, { "name": "FName", "type": "W", "size": 50 }, { "name": "FAge", "type": "I" }, { "name": "FNote", "type": "WM" } ], "Vdsrows": [ [ 1, "admin", "admin", "管理员", 18, "这是管理员的备注" ], [ 2, "test", "test", "测试", 3, "这是测试员的备注" ] ] } }
```

=====

name:FID type:数值

name:FUser type:字符串 size:50

name:FPwd type:字符串 size:50

name:FName type:字符串 size:50

name:FAge type:数值

name:FNote type:备注

1,admin,admin,管理员,18,这是管理员的备注

2,test,test,测试,3,这是测试员的备注