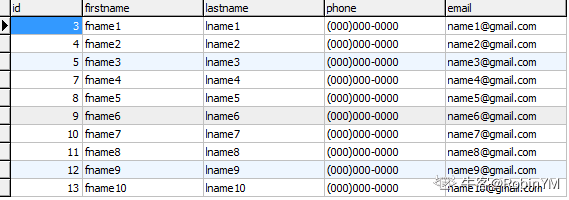
**前端连载学习笔记概要四（jQuery EasyUI）**+关注

jQuery EasyUI 是一个基于 jQuery 的框架，集成了各种用户界面插件。提供建立现代化的具有交互性的 javascript 应用的必要的功能。不需要写太多 javascript 代码，一般情况下您只需要使用一些 html 标记来定义用户界面，网页的完整框架。

jEasyUI应用

**创建CRUD应用**

**1. 准备数据库**



**2.创建DataGrid来显示用户信息（datagrid：向用户展示列表数据）**

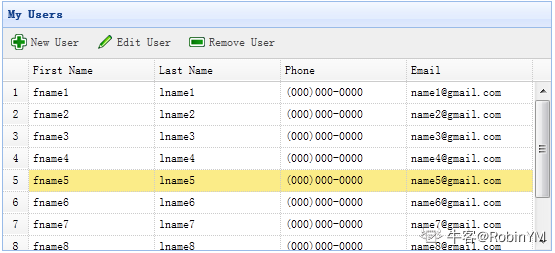
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | <table id="dg" title="My Users" class="easyui-datagrid" style="width:550px;height:250px"          url="get\_users.php"          toolbar="#toolbar"          rownumbers="true" fitColumns="true" singleSelect="true">      <thead>          <tr>              <th field="firstname" width="50">First Name</th>              <th field="lastname" width="50">Last Name</th>              <th field="phone" width="50">Phone</th>              <th field="email" width="50">Email</th>          </tr>      </thead>  </table>  <div id="toolbar">      <a href="#" class="easyui-linkbutton" iconCls="icon-add" plain="true" onclick="newUser()">New User</a>      <a href="#" class="easyui-linkbutton" iconCls="icon-edit" plain="true" onclick="editUser()">Edit User</a>      <a href="#" class="easyui-linkbutton" iconCls="icon-remove" plain="true" onclick="destroyUser()">Remove User</a>  </div> |

DataGrid 使用 'url' 属性，并赋值为 'get\_users.php'，用来从服务器检索数据。

**get\_users.php** 文件代码如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | $rs = mysql\_query('select \* from users');  $result = array();  while($row = mysql\_fetch\_object($rs)){      array\_push($result, $row);  }    echo json\_encode($result); |

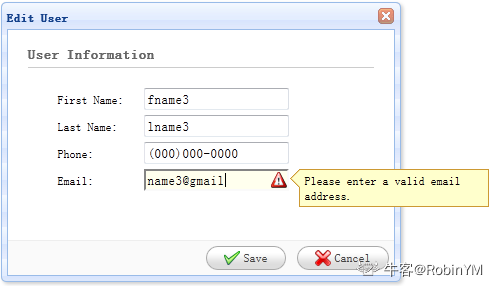
显示效果如下：



**3.创建表单对话框**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | <div id="dlg" class="easyui-dialog" style="width:400px;height:280px;padding:10px 20px"          closed="true" buttons="#dlg-buttons">      <div class="ftitle">User Information</div>      <form id="fm" method="post">          <div class="fitem">              <label>First Name:</label>              <input name="firstname" class="easyui-validatebox" required="true">          </div>          <div class="fitem">              <label>Last Name:</label>              <input name="lastname" class="easyui-validatebox" required="true">          </div>          <div class="fitem">              <label>Phone:</label>              <input name="phone">          </div>          <div class="fitem">              <label>Email:</label>              <input name="email" class="easyui-validatebox" validType="email">          </div>      </form>  </div>  <div id="dlg-buttons">      <a href="#" class="easyui-linkbutton" iconCls="icon-ok" onclick="saveUser()">Save</a>      <a href="#" class="easyui-linkbutton" iconCls="icon-cancel" onclick="javascript:$('#dlg').dialog('close')">Cancel</a>  </div> |

效果如下：



**4.实现创建和编辑用户**

当创建用户时，打开一个对话框并清空表单数据

|  |  |
| --- | --- |
| 1  2  3  4  5 | function newUser(){      $('#dlg').dialog('open').dialog('setTitle','New User');      $('#fm').form('clear');      url = 'save\_user.php';  } |

当编辑用户时，打开一个对话框并从datagrid选择的行中加载表单数据

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | var row = $('#dg').datagrid('getSelected');  if (row){      $('#dlg').dialog('open').dialog('setTitle','Edit User');      $('#fm').form('load',row);      url = 'update\_user.php?id='+row.id;  } |

url' 存储着当保存用户数据时表单回传的 URL 地址。

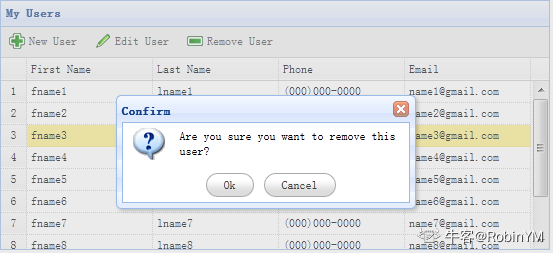
**5.保存用户数据**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | function saveUser(){      $('#fm').form('submit',{          url: url,          onSubmit: function(){              return $(this).form('validate');          },          success: function(result){              var result = eval('('+result+')');              if (result.errorMsg){                  $.messager.show({                      title: 'Error',                      msg: result.errorMsg                  });              } else {                  $('#dlg').dialog('close');        // close the dialog                  $('#dg').datagrid('reload');    // reload the user data              }          }      });  } |

提交表单之前，'onSubmit' 函数将被调用，该函数用来验证表单字段值。当表单字段值提交成功，关闭对话框并重新加载 datagrid 数据。

**6.删除一个用户**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | function destroyUser(){      var row = $('#dg').datagrid('getSelected');      if (row){          $.messager.confirm('Confirm','Are you sure you want to destroy this user?',function(r){              if (r){                  $.post('destroy\_user.php',{id:row.id},function(result){                      if (result.success){                          $('#dg').datagrid('reload');    // reload the user data                      } else {                          $.messager.show({    // show error message                              title: 'Error',                              msg: result.errorMsg                          });                      }                  },'json');              }          });      }  } |

  
移除一行之前,我们将显示一个确认对话框让用户决定是否真的移除该行数据。当移除数据成功之后，调用 'reload' 方法来刷新 datagrid 数据。

7.开启MySQL，在浏览器运行代码。

**创建CRUD数据网络**

**1.在HTML标签中定义数据网格DataGrid**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | <table id="dg" title="My Users" style="width:550px;height:250px"          toolbar="#toolbar" idField="id"          rownumbers="true" fitColumns="true" singleSelect="true">      <thead>          <tr>              <th field="firstname" width="50" editor="{type:'validatebox',options:{required:true}}">First Name</th>              <th field="lastname" width="50" editor="{type:'validatebox',options:{required:true}}">Last Name</th>              <th field="phone" width="50" editor="text">Phone</th>              <th field="email" width="50" editor="{type:'validatebox',options:{validType:'email'}}">Email</th>          </tr>      </thead>  </table>  <div id="toolbar">      <a href="#" class="easyui-linkbutton" iconCls="icon-add" plain="true" onclick="javascript:$('#dg').edatagrid('addRow')">New</a>      <a href="#" class="easyui-linkbutton" iconCls="icon-remove" plain="true" onclick="javascript:$('#dg').edatagrid('destroyRow')">Destroy</a>      <a href="#" class="easyui-linkbutton" iconCls="icon-save" plain="true" onclick="javascript:$('#dg').edatagrid('saveRow')">Save</a>      <a href="#" class="easyui-linkbutton" iconCls="icon-undo" plain="true" onclick="javascript:$('#dg').edatagrid('cancelRow')">Cancel</a>  </div> |

**2.使用可编辑的数据网格DataGrid**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | $('#dg').edatagrid({      url: 'get\_users.php',      saveUrl: 'save\_user.php',      updateUrl: 'update\_user.php',      destroyUrl: 'destroy\_user.php'  }); |

 'url'、'saveUrl'、'updateUrl' 和 'destroyUrl' 属性来编辑数据网格

* url：从服务器端检索用户数据
* saveUrl：保存一个新的用户数据
* updateUrl：更新一个已存在的用户数据
* destroyUrl：删除一个已存在的用户数据

**3.写服务器处理代码**

保存一个新的用户（save\_user.php）：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | $firstname = $\_REQUEST['firstname'];  $lastname = $\_REQUEST['lastname'];  $phone = $\_REQUEST['phone'];  $email = $\_REQUEST['email'];    include 'conn.php';    $sql = "insert into users(firstname,lastname,phone,email) values('$firstname','$lastname','$phone','$email')";  @mysql\_query($sql);  echo json\_encode(array(      'id' => mysql\_insert\_id(),      'firstname' => $firstname,      'lastname' => $lastname,      'phone' => $phone,      'email' => $email  )); |

更新一个已存在用户（update\_user.php）：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | $id = intval($\_REQUEST['id']);  $firstname = $\_REQUEST['firstname'];  $lastname = $\_REQUEST['lastname'];  $phone = $\_REQUEST['phone'];  $email = $\_REQUEST['email'];    include 'conn.php';    $sql="update users set firstname='$firstname',lastname='$lastname',phone='$phone',email='$email' where id=$id";  @mysql\_query($sql);  echo json\_encode(array(      'id' => $id,      'firstname' => $firstname,      'lastname' => $lastname,      'phone' => $phone,      'email' => $email  )); |

删除一个已存在用户（destroy\_user.php）：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | $id = intval($\_REQUEST['id']);    include 'conn.php';    $sql = "delete from users where id=$id";  @mysql\_query($sql);  echo json\_encode(array('success'=>true)); |

**表单的CRUD应用**

1.在HTML标签中定义数据网格DataGrid

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | <table id="dg" title="My Users" style="width:550px;height:250px"          url="get\_users.php"          toolbar="#toolbar"          fitColumns="true" singleSelect="true">      <thead>          <tr>              <th field="firstname" width="50">First Name</th>              <th field="lastname" width="50">Last Name</th>              <th field="phone" width="50">Phone</th>              <th field="email" width="50">Email</th>          </tr>      </thead>  </table>  <div id="toolbar">      <a href="#" class="easyui-linkbutton" iconCls="icon-add" plain="true" onclick="newItem()">New</a>      <a href="#" class="easyui-linkbutton" iconCls="icon-remove" plain="true" onclick="destroyItem()">Destroy</a>  </div> |

2.为数据网格应用明细视图

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | $('#dg').datagrid({      view: detailview,      detailFormatter:function(index,row){          return '<div class="ddv"></div>';      },      onExpandRow: function(index,row){          var ddv = $(this).datagrid('getRowDetail',index).find('div.ddv');          ddv.panel({              border:false,              cache:true,              href:'show\_form.php?index='+index,              onLoad:function(){                  $('#dg').datagrid('fixDetailRowHeight',index);                  $('#dg').datagrid('selectRow',index);                  $('#dg').datagrid('getRowDetail',index).find('form').form('load',row);              }          });          $('#dg').datagrid('fixDetailRowHeight',index);      }  }); |

 html 页面头部引入 'datagrid-detailview.js' 文件，使用 'detailFormatter' 函数来生成行明细内容。 在这种情况下，我们返回一个用于放置编辑表单（form）的空的 <div>。 当用户点击行展开按钮（'+'）时，'onExpandRow' 事件将被触发，我们将通过 ajax 加载编辑表单（form）。 调用 'getRowDetail' 方法来得到行明细容器，所以我们能查找到行明细面板（panel）。 在行明细中创建面板（panel），加载从 'show\_form.php' 返回的编辑表单（form）

3.创建编辑表单Form

编辑表单是从服务器加载的show\_form.php

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | <form method="post">      <table class="dv-table" style="width:100%;background:#fafafa;padding:5px;margin-top:5px;">          <tr>              <td>First Name</td>              <td><input name="firstname" class="easyui-validatebox" required="true"></input></td>              <td>Last Name</td>              <td><input name="lastname" class="easyui-validatebox" required="true"></input></td>          </tr>          <tr>              <td>Phone</td>              <td><input name="phone"></input></td>              <td>Email</td>              <td><input name="email" class="easyui-validatebox" validType="email"></input></td>          </tr>      </table>      <div style="padding:5px 0;text-align:right;padding-right:30px">          <a href="#" class="easyui-linkbutton" iconCls="icon-save" plain="true" onclick="saveItem(&lt;?php echo $\_REQUEST['index'];?&gt;)">Save</a>          <a href="#" class="easyui-linkbutton" iconCls="icon-cancel" plain="true" onclick="cancelItem(&lt;?php echo $\_REQUEST['index'];?&gt;)">Cancel</a>      </div>  </form> |

4.保存或取消编辑

调用 'saveItem' 函数来保存一个用户或者调用 'cancelItem' 函数来取消编辑。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | function saveItem(index){      var row = $('#dg').datagrid('getRows')[index];      var url = row.isNewRecord ? 'save\_user.php' : 'update\_user.php?id='+row.id;      $('#dg').datagrid('getRowDetail',index).find('form').form('submit',{          url: url,          onSubmit: function(){              return $(this).form('validate');          },          success: function(data){              data = eval('('+data+')');              data.isNewRecord = false;              $('#dg').datagrid('collapseRow',index);              $('#dg').datagrid('updateRow',{                  index: index,                  row: data              });          }      });  } |

决定要回传哪一个 URL，然后查找表单（form）对象，并调用 'submit' 方法来提交表单（form）数据。当保存数据成功时，折叠并更新行数据。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | function cancelItem(index){      var row = $('#dg').datagrid('getRows')[index];      if (row.isNewRecord){          $('#dg').datagrid('deleteRow',index);      } else {          $('#dg').datagrid('collapseRow',index);      }  } |