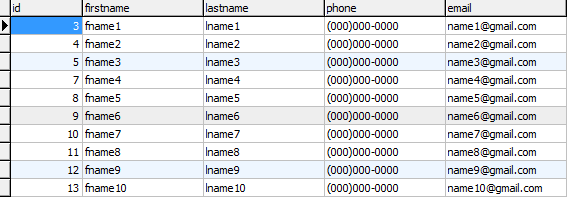
数据收集并妥善管理数据是网络应用共同的必要。CRUD 允许我们生成页面列表，并编辑数据库记录。本教程将向你演示如何使用 jQuery EasyUI 框架实现一个 CRUD DataGrid。

我们将使用下面的插件：

* datagrid：向用户展示列表数据。
* dialog：创建或编辑一条单一的用户信息。
* form：用于提交表单数据。
* messager：显示一些操作信息。

#### 步骤 1：准备数据库

我们将使用 MySql 数据库来存储用户信息。创建数据库和 'users' 表。

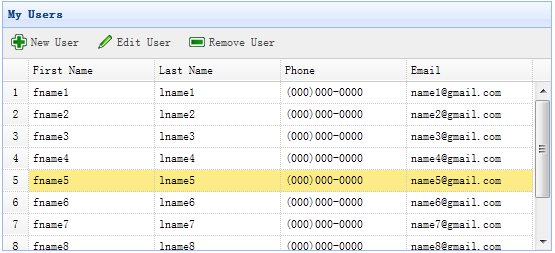


#### 步骤 2：创建 DataGrid 来显示用户信息

创建没有 javascript 代码的 DataGrid。

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

我们不需要写任何的 javascript 代码，就能向用户显示列表，如下图所示：



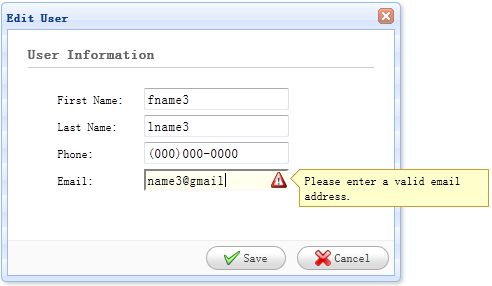
DataGrid 使用 'url' 属性，并赋值为 'get\_users.php'，用来从服务器检索数据。

#### 步骤 3：创建表单对话框

我们使用相同的对话框来创建或编辑用户。

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

这个对话框已经创建，也没有任何的 javascript 代码:。



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

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

'url' 存储着当保存用户数据时表单回传的 URL 地址。

#### 步骤 5：保存用户数据

我们使用下面的代码保存用户数据：

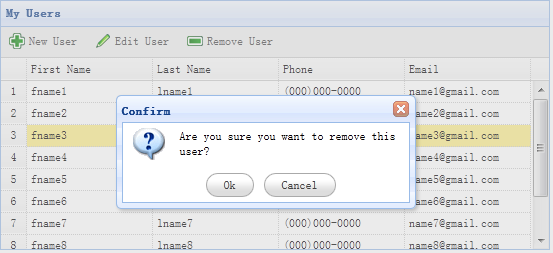
1. function saveUser(){
2. $('#fm').form('submit',{
3. url: url,
4. onSubmit: function(){
5. return $(this).form('validate');
6. },
7. success: function(result){
8. var result = eval('('+result+')');
9. if (result.errorMsg){
10. $.messager.show({
11. title: 'Error',
12. msg: result.errorMsg
13. });
14.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

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。

#### 步骤 6：删除一个用户

我们使用下面的代码来移除一个用户：

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



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。