

Build CRUD DataGrid with jQuery EasyUI using PHP and MySQL

By: CodexWorld | In: [jQuery](#) | Last Updated: Sep 17, 2019

DataGrid with CRUD operations is the most used functionality for the data management section. Generally, the web page is reloaded when the CRUD operation occurs. To make the web application user-friendly, you can integrate the CRUD functionality without page refresh. The EasyUI framework provides an easy way to integrate DataGrid with CRUD feature in the web application.

EasyUI is a jQuery framework that helps to build modern and interactive **DataGrid CRUD** application quickly. The DataGrid functionality can be integrated into the web page in less time by writing less code. The jQuery EasyUI allows interacting with the server-side script to make the DataGrid more powerful. In this tutorial, we will show you how to build **CRUD with search and pagination** using EasyUI, PHP, and MySQL.

In the EasyUI integration example code, the following functionality will be implemented.

- Fetch data from the database and list them in a tabular format.
- Add/Edit data in dialog window without page refresh.
- Delete data from the database without page refresh.
- Add the search and pagination feature to the list.

jQuery EasyUI Integration

The following code shows how to integrate jQuery EasyUI plugin in the web page to **build a CRUD application** and create or edit user information using dialog component.

1. Include the CSS and JavaScript files of the **EasyUI plugin** on the web page.

```
<link rel="stylesheet" type="text/css" href="easyui/themes/default/easyui.css">
<link rel="stylesheet" type="text/css" href="easyui/themes/icon.css">
```

```
<script type="text/javascript" src="easyui/jquery.min.js"></script>
<script type="text/javascript" src="easyui/jquery.easyui.min.js"></script>
```

2. Add HTML code for data list table and toolbar.

- Specify the URL of the server-side script (`getData.php`) in `url` attribute of the `<table>` tag.
- Use pagination attribute and set it TRUE (`pagination="true"`) to add pagination links to the data list.

```
<table id="dg" title="Users Management" class="easyui-datagrid" url="getData.php" toolbar="#toolbar"
pagination="true" rownumbers="true" fitColumns="true" singleSelect="true" style="width:100%;
height:350px;">
  <thead>
    <tr>
      <th field="first_name" width="50">First Name</th>
      <th field="last_name" width="50">Last Name</th>
      <th field="email" width="50">Email</th>
      <th field="phone" width="50">Phone</th>
    </tr>
  </thead>
</table>
```

```
<div id="toolbar">
  <div id="tb">
    <input id="term" placeholder="Type keywords...">
    <a href="javascript:void(0);" class="easyui-linkbutton" plain="true" onclick="doSearch()">Search</a>
  </div>
  <div id="tb2" style="">
    <a href="javascript:void(0)" class="easyui-linkbutton" iconCls="icon-add" plain="true" onclick="newUser()">New User</a>
    <a href="javascript:void(0)" class="easyui-linkbutton" iconCls="icon-edit" plain="true" onclick="editUser()">Edit User</a>
    <a href="javascript:void(0)" class="easyui-linkbutton" iconCls="icon-remove" plain="true" onclick="destroyUser()">Remove User</a>
  </div>
</div>
```

3. Add HTML code for add/edit form dialog.

```
<div id="dlg" class="easyui-dialog" style="width:450px" data-options="closed:true,modal:true,border:'thin',buttons:'#dlg-buttons'">
  <form id="fm" method="post" novalidate style="margin:0;padding:20px 50px">
    <h3>User Information</h3>
    <div style="margin-bottom:10px">
      <input name="first_name" class="easyui-textbox" required="true" label="First Name:" style="width:100%">
    </div>
    <div style="margin-bottom:10px">
      <input name="last_name" class="easyui-textbox" required="true" label="Last Name:" style="width:100%">
    </div>
    <div style="margin-bottom:10px">
      <input name="email" class="easyui-textbox" required="true" validType="email" label="Email:" style="width:100%">
    </div>
    <div style="margin-bottom:10px">
      <input name="phone" class="easyui-textbox" required="true" label="Phone:" style="width:100%">
    </div>
  </form>
</div>
```

```

00%>
    </div>
</form>
</div>
<div id="dlg-buttons">
    <a href="javascript:void(0);" class="easyui-linkbutton c6" iconCls="icon-ok" onclick="saveUser()"
style="width:90px;">Save</a>
    <a href="javascript:void(0);" class="easyui-linkbutton" iconCls="icon-cancel" onclick="javascrip
t:$('#dlg').dialog('close');" style="width:90px;">Cancel</a>
</div>

```

4. Add the JavaScript code for server-side interaction.

- **doSearch()** – Send terms to the server-side script (`getData.php`) and load the filtered data.
- **newUser()** – Open popup dialog with the HTML form to add data.
- **editUser()** – Open popup dialog with the HTML form to edit data.
- **saveUser()** – Send data to the server-side script (`addData.php` or `editData.php`) for saving in the database.
- **destroyUser()** – Send request to the server-side script (`deleteData.php`) for remove data.

```

<script type="text/javascript">
function doSearch(){
    $('#dg').datagrid('load', {
        term: $('#term').val()
    });
}

var url;
function newUser(){
    $('#dlg').dialog('open').dialog('center').dialog('setTitle','New User');
    $('#fm').form('clear');
    url = 'addData.php';
}
function editUser(){
    var row = $('#dg').datagrid('getSelected');
    if (row){
        $('#dlg').dialog('open').dialog('center').dialog('setTitle','Edit User');
        $('#fm').form('load',row);
        url = 'editData.php?id='+row.id;
    }
}
function saveUser(){
    $('#fm').form('submit',{
        url: url,
        onSubmit: function(){
            return $(this).form('validate');
        },
        success: function(response){
            var respData = $.parseJSON(response);
            if(respData.status == 0){
                $.messager.show({
                    title: 'Error',
                    msg: respData.msg
                });
            }else{
                $('#dlg').dialog('close');
                $('#dg').datagrid('reload');
            }
        }
    });
}

```

```

    }
  });
}
function destroyUser(){
  var row = $('#dg').datagrid('getSelected');
  if (row){
    $.messager.confirm('Confirm','Are you sure you want to delete this user?',function(r){
      if (r){
        $.post('deleteData.php', {id:row.id}, function(response){
          if(response.status == 1){
            $('#dg').datagrid('reload');
          }else{
            $.messager.show({
              title: 'Error',
              msg: respData.msg
            });
          }
        },'json');
      }
    });
  }
}
}
</script>

```

Server-side Processing

Create Database Table:

To store the data a table is required in the database. The following SQL creates an `users` table in the MySQL database with some basic fields.

```

CREATE TABLE `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `first_name` varchar(25) COLLATE utf8_unicode_ci NOT NULL,
  `last_name` varchar(25) COLLATE utf8_unicode_ci NOT NULL,
  `email` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  `phone` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

```

Database Configuration (dbConnect.php):

The `dbConnect.php` file is used to connect with the database. Specify the database host (`$dbHost`), username (`$dbUsername`), password (`$dbPassword`), and name (`$dbName`) as per your MySQL database credentials.

```

<?php
// Database configuration
$dbHost = "localhost";
$dbUsername = "root";
$dbPassword = "root";
$dbName = "codexworld";

// Create database connection
$db = new mysqli($dbHost, $dbUsername, $dbPassword, $dbName);

// Check connection
if ($db->connect_error) {

```

```
die("Connection failed: " . $db->connect_error);
}
```

Fetch Data (getData.php):

This `getData.php` file is used to fetch data from the database using PHP and MySQL.

- Based on the search terms and page number, the records are returned in JSON format.

```
<?php
// Include the database config file
require_once 'dbConnect.php';

$page = isset($_POST['page']) ? intval($_POST['page']) : 1;
$rows = isset($_POST['rows']) ? intval($_POST['rows']) : 10;

$searchTerm = isset($_POST['term']) ? $db->real_escape_string($_POST['term']) : "";

$offset = ($page-1)*$rows;

$result = array();

$whereSQL = "first_name LIKE '$searchTerm%' OR last_name LIKE '$searchTerm%' OR email LIK
E '$searchTerm%' OR phone LIKE '$searchTerm%'";
$result = $db->query("SELECT COUNT(*) FROM users WHERE $whereSQL");
$row = $result->fetch_row();
$response["total"] = $row[0];

$result = $db->query( "SELECT * FROM users WHERE $whereSQL ORDER BY id DESC LIM
IT $offset,$rows");

$users = array();
while($row = $result->fetch_assoc()){
    array_push($users, $row);
}
$response["rows"] = $users;

echo json_encode($response);
```

Add Data (addData.php):

The `addData.php` file is used to insert data in the database using PHP and MySQL.

```
<?php
$response = array(
    'status' => 0,
    'msg' => 'Some problems occurred, please try again.'
);
if(!empty($_REQUEST['first_name']) && !empty($_REQUEST['last_name']) && !empty($_REQUE
ST['email']) && !empty($_REQUEST['phone'])){
    $first_name = $_REQUEST['first_name'];
    $last_name = $_REQUEST['last_name'];
    $email = $_REQUEST['email'];
    $phone = $_REQUEST['phone'];

    // Include the database config file
    require_once 'dbConnect.php';

    $sql = "INSERT INTO users(first_name,last_name,email,phone) VALUES ('$first_name','$last_n
ame','$email','$phone)";
```

```

$insert = $db->query($sql);

if($insert){
    $response['status'] = 1;
    $response['msg'] = 'User data has been added successfully!';
}
}else{
    $response['msg'] = 'Please fill all the mandatory fields.';
}

echo json_encode($response);

```

Update Data (editData.php):

The `editData.php` file is used to update data based on the row ID using PHP and MySQL.

```

<?php
$response = array(
    'status' => 0,
    'msg' => 'Some problems occurred, please try again.'
);
if(!empty($_REQUEST['first_name']) && !empty($_REQUEST['last_name']) && !empty($_REQUEST['email']) && !empty($_REQUEST['phone'])){
    $first_name = $_REQUEST['first_name'];
    $last_name = $_REQUEST['last_name'];
    $email = $_REQUEST['email'];
    $phone = $_REQUEST['phone'];

    if(!empty($_REQUEST['id'])){
        $id = intval($_REQUEST['id']);

        // Include the database config file
        require_once 'dbConnect.php';

        $sql = "UPDATE users SET first_name='$first_name', last_name='$last_name', email='$email', phone='$phone' WHERE id = $id";
        $update = $db->query($sql);

        if($update){
            $response['status'] = 1;
            $response['msg'] = 'User data has been updated successfully!';
        }
    }
}else{
    $response['msg'] = 'Please fill all the mandatory fields.';
}

echo json_encode($response);

```

Delete Data (deleteData.php):

The `deleteData.php` file is used to delete data from the database based on the row ID.

```

<?php
$response = array(
    'status' => 0,
    'msg' => 'Some problems occurred, please try again.'
);
if(!empty($_REQUEST['id'])){
    $id = intval($_REQUEST['id']);

```

```
// Include the database config file
require_once 'dbConnect.php';

$sql = "DELETE FROM users WHERE id = $id";
$delete = $db->query($sql);

if($delete){
    $response['status'] = 1;
    $response['msg'] = 'User data has been deleted successfully!';
}

echo json_encode($response);
```

PHP CRUD Operations without Page Refresh using jQuery, Ajax, and MySQL

Conclusion

EasyUI is very useful when you want to integrate CRUD functionality instantly without writing much code. It helps to build a CRUD application with server-side processing using PHP and MySQL. There are various plugins available in EasyUI which allow you to enhance the DataGrid functionality.