# Web Database Programming

Created: 2011-01-21

Last update: 2014-01-14

## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>2</td>
</tr>
<tr>
<td>Use EasyDataSet as Data Source</td>
<td>2</td>
</tr>
<tr>
<td>Add EasyDataSet to web page</td>
<td>3</td>
</tr>
<tr>
<td>Make Database Connection</td>
<td>4</td>
</tr>
<tr>
<td>Use Query Builder to get data</td>
<td>7</td>
</tr>
<tr>
<td>Bind-data to single field</td>
<td>10</td>
</tr>
<tr>
<td>Data Query</td>
<td>13</td>
</tr>
<tr>
<td>Data Search</td>
<td>17</td>
</tr>
<tr>
<td>Data Navigation</td>
<td>18</td>
</tr>
<tr>
<td>Modify and Save Data</td>
<td>20</td>
</tr>
<tr>
<td>MySql Database Credential</td>
<td>24</td>
</tr>
<tr>
<td>Test Data Entry</td>
<td>25</td>
</tr>
<tr>
<td>Bind Data to Html Table</td>
<td>28</td>
</tr>
<tr>
<td>Web Page Security</td>
<td>33</td>
</tr>
<tr>
<td>Field Editors</td>
<td>33</td>
</tr>
<tr>
<td>Data Repeater</td>
<td>33</td>
</tr>
<tr>
<td>Change Query Filters at Runtime</td>
<td>33</td>
</tr>
<tr>
<td>Create New Records</td>
<td>33</td>
</tr>
<tr>
<td>Data Streaming</td>
<td>33</td>
</tr>
<tr>
<td>Fetch Data of One-to-Many Relation</td>
<td>33</td>
</tr>
<tr>
<td>Change Data-binding</td>
<td>33</td>
</tr>
<tr>
<td>Feedbacks</td>
<td>33</td>
</tr>
</tbody>
</table>
Introduction
Web pages can be designed as front tier for showing data from databases and entering data to be stored in databases.

Limnor Studio uses two-way data-binding to simplify web database programming. Ajax is used for retrieving data from databases and sending data to databases.

From visual programming point of view, using databases in a web application is very much the same as using databases in a standalone application.

We create a PHP web application to show the process:

In this document we only use components which can be used in both PHP and .Net web applications. Therefore you may choose to select Web Application (.Net) and every operation described in this document is valid.


The contents of this document were tested on IE, Opera, Chrome, Safari, and Firefox.

Use EasyDataSet as Data Source
We use a Student table with following fields as our sample database.

StudentId, Firstname, Lastname, BirthDate, Grade
We used the following Sql script to create the table in a MySql database.

```sql
CREATE TABLE Student (
  StudentId int not null auto_increment,
  Firstname nvarchar (30),
  Lastname nvarchar (30),
  BirthDate datetime,
  Grade int,
  Primary Key (StudentId)
);
```

Or you may use some GUI tool to create and manage your MySql database. This is not in the scope of this document.

**Add EasyDataSet to web page**

We add an EasyDataSet to the web page to connect the MySql database and get data from the Student table:

Note that the EasyDataSet component can be used in all kinds of applications: standalone application; PHP web applications; .Net web applications. The configuration of this component is also exactly the same for all kinds of applications.

If you already used EasyDataSet in standalone applications then you may skip this chapter.
**Make Database Connection**

Set DatabaseConnection property of the EasyDataSet:

Choose “Other connection type” for the DatabaseType property:

Choose “MySql.MySqlClient.MySqlConnection”. Click OK:
Set the connection string to connect to the MySql database:

Give the connection a name:
Click \( \text{test} \) to test the connection:

Note that we are connecting the database using ADO.NET driver. **This is for design time only.** At runtime for PHP web applications, ADO.NET driver is not used because it may not be available in a Linux machine. PHP library for MySQL database will be used. All this is done behind the scene. You, as a developer,
develop in the same way for web applications and for standalone applications. Limnor Studio compiler will generate PHP code for you.

Note that after setting a database connection, the solution must be re-built before you can click the green arrow button to test the web pages.

Use Query Builder to get data

Set the SQL property of the EasyDataSet to fetch Student data:

Add all the fields we want to display in the web page to the query. Click Filter button to add filters for limiting the range of data:
For this sample, we simply use a range of StudentId as the filter. We will let the web user to enter the range from web browser. So, we will use parameters in the filter.

Select the StudentId, choose >, choose Param, and give a parameter name id1:

Click Add filter. The filter appears:
Select <; give another parameter name id2. Click “Add AND filter”:

For this sample, that is all we want. Click OK buttons.
Bind-data to single field

To allow the user to do data entry we may use a text box. For displaying data only we may use a label.

We use a label to display StudentId:

![Label for StudentId](image)

Rename it to lblStudentId:

![Label properties](image)

Set its Databindings property to bind its Text to StudentId:
Use a text box to show and edit first name:

Rename it to txtFirstname:
Bind its Text to the Firstname field:

In the same way we may create data-bindings for other fields:

To change data-binding at runtime, see chapter “Change Data-binding”.
Data Query
To get data from the database, we need to execute a query action. Since we use parameters in filters, we need to build a UI to let the user enter values for the parameters.

We use two text boxes for entering id1 and id2:

We use a button to trigger a query:

When the user clicks the button we want to execute a query.

Right-click the button; choose “Assign Action”; choose “onclick” event:
Select QueryWithParameterValues of the EasyDataSet; click Next:

Select Property for @id1:
Select the Text property of the text box for starting id:

For @id2, select the Text property of the text box for end id:
Click OK to finish creating the query action and assigning it to the button:
We are done creating an action and assigning it to the button. You can see it in the Event Map and Object Explorer.

Data Search
EasyDataSet has a method Search for searching data fetched from database:
There are huge differences between a Search action and a Query action. A Query action searches data in the database and downloads found data to web page. A Search action searches among the data downloaded from the database; the found record becomes the current record.

To do a Search, a Query action must be executed first to fetch data from the database server.

**Data Navigation**

On fetching a range of student records, we may use “go next”, “go previous”, “go first” and “go last” actions to navigate through the records.

We add 4 buttons for this purpose:
Let these buttons trig actions. Right-click button "|<"; choose “Assign Action”; choose “onclick” event:

Choose “MoveFirst” from the EasyDataSet; click Next:

Click OK to finish creating the action and assigning the action to the event:
Do the same for the other 3 buttons and use “MovePrevious”, “MoveNext” and “MoveLast” respectively.

Modify and Save Data
We may use a button for adding new record and a button for removing the current record.

Right-click “Add student” button; choose “Assign Action”; choose “onclick” event:
Choose “AddNewRecord” and click Next:

Click OK to finish creating the action and assigning it to the event:

Right-click button “Delete”; choose “Assign Action”; choose “onclick” event:
Choose “DeleteCurrentRecord”; click Next:

Click OK to finish creating the action and assigning it to the event:

One click of the “Add student” button will add one new student record on the page. Click the “Delete” button will remove the current record from the page. Typing in the text boxes will modify the student data on the page.
Note that all the modifications are kept only in the web page. Nothing is changed in the database.

To save all the modifications kept in the page to the database, we need to execute an Update action. We use a “Save” button to trig such an action.

Right-click the “Save” button; choose “Assign Action”; choose “onclick” event:

Choose “Update”; click Next:

Click OK to finish creating the action and assigning it to the event:
We have all the buttons triggering actions executed by the EasyDataSet:

**MySql Database Credential**

Compile the project. All web files are generated under WebFiles folder.

The credential for MySql database is in a file named mySqlcredential.php in the libPhp folder:

You may edit this page to set the credential for you database.
You edit the PHP code in the above file to provide MySQL credential in a variable named $cr. The file will not be overwritten by the compiler if you modify it.

**Test Data Entry**

Let’s turn on DebugMode so that we may watch client/server communications and watch for any errors:

Start debugging:
Enter id range and click “Query database” to get data:

We’ll see empty data if no records are found in the id range:
Click “Add student” button. Enter some data in the text boxes.
You may click “Add student” many times to add many students. You may click the navigation buttons "|<", "<", ">", and "|" to go through the students. You may also click “Delete” button to remove the student record currently displayed on the page.

After making all the changes, click “Save” button to save the data to the database.

From the debugging information, we can see that how many records are added; how many records are deleted and how many records are modified:

### Debug Information from WebPage1.php

#### Client request

```json
client request:{"Calls":[]}"method":"jdbc.Db.putData", "value": "Table132102189"}, "Data": [{"tableName": "Table132102189"}, "Columns": 
[{"Name": "StudentID", "ReadOnly": 1, "Type": 3}, "Name": "Firstname", "ReadOnly": 0, "Type": 208},
{"Name": "Lastname", "ReadOnly": 0, "Type": 208}, "Name": "BirthDate", "ReadOnly": 0, "Type": 16],
{"Name": "Grade", "ReadOnly": 0, "Type": 3}, "PrimaryKey": 1, "StudentID": 4, "Rows": 3,"added": true,"ItemArray": []}, "KeyValues": [{"changed": true,"added": true,"ItemArray": [{"David"}], "Jackson", "6"}, "KeyValues": [], "changed": true,"ItemArray": [], "columnsIndexes": [{"StudentID": 0, "Firstname": 1, "Lastname": 2, "BirthDate": 3, "Grade": 4}, "rowindex": 5}]}
```

#### Server response

```json
{"Calls":[]}"Data":null,"values":[]
```

We may close the web page and re-open it. Set the id range and click “Query database” button. Now we will see student records we just entered. It proves that the data we entered have been saved in the database.

### Bind Data to Html Table
Records from a database can be bound to a HTML table for displaying and editing.

Drop an HtmlTable to the web page:
Set its DataSource property to EasyDataSet1:

Run the project:
Query the database by clicking the “Query database” button:

The data appears in the html elements:
The highlighted row of the html table indicates the current record of the data. The user may click a row in the html table to make it the current record. Let’s click the second row:
The user may click a cell in the html table to edit the data in the cell. Let's click the Grade cell of the second row:

<table>
<thead>
<tr>
<th>StudentId</th>
<th>Firstname</th>
<th>Lastname</th>
<th>BirthDate</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>David</td>
<td>Jackson</td>
<td>{null}</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>StudentId</th>
<th>Firstname</th>
<th>Lastname</th>
<th>BirthDate</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mike</td>
<td>Jonse</td>
<td>{null}</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>StudentId</th>
<th>Firstname</th>
<th>Lastname</th>
<th>BirthDate</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>David</td>
<td>Jackson</td>
<td>{null}</td>
<td>8</td>
</tr>
</tbody>
</table>
Data Repeater
In chapter “Bind-data to single field”, we saw that a form can be designed with data-bound controls. Data from database are automatically displayed on the controls. The user may modify data on the controls and the data modifications can be saved back to database.

In such arrangement, one record is displayed on one web page.

Data Repeater allows you to design the form with data-bound controls, but the same design can be repeated on one web page. Thus many records can be displayed on one web page.

For details, see [http://www.limnor.com/support/WebDataRepeater.pdf](http://www.limnor.com/support/WebDataRepeater.pdf)

Change Query Filters at Runtime

Create New Records

Data Streaming

Fetch Data of One-to-Many Relation

Change Data-binding

Feedbacks
Please send your feedbacks to support@limnor.com, thanks!