Table of contents

About DMXzone Server Connect

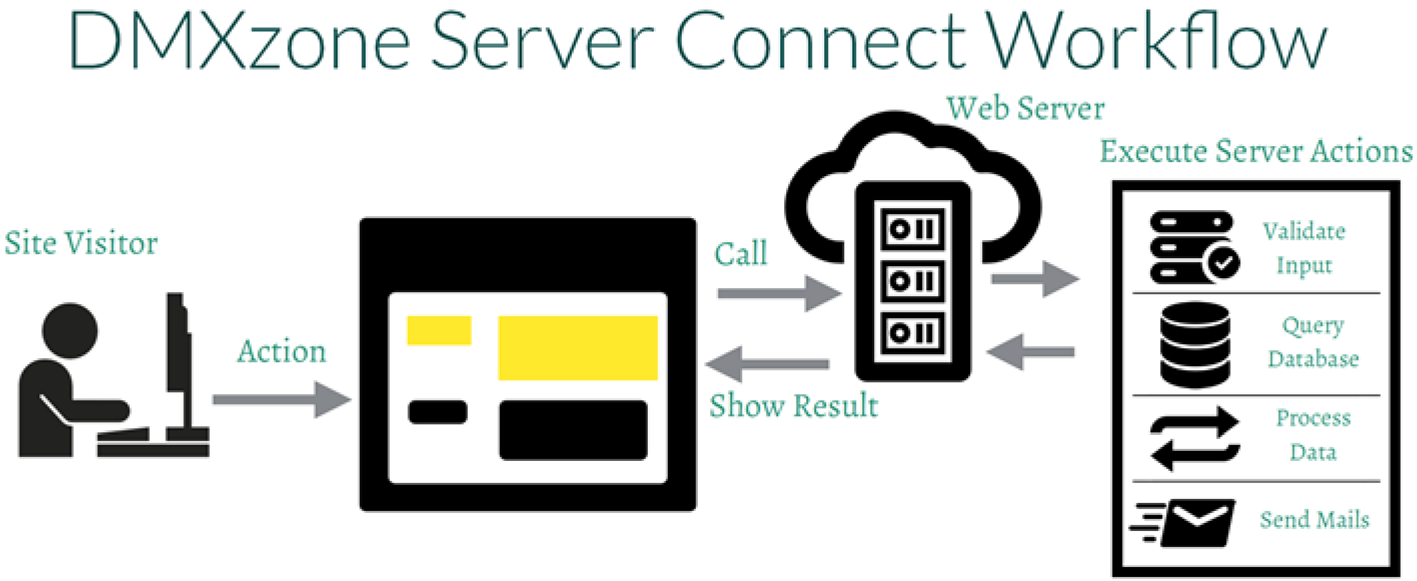
Imagine full visual programming without any coding. It is now possible with the next generation server side framework for Adobe Dreamweaver: the DMXzone Server Connect. It will provide you with a full visual workflow builder for server side workflows and many ready to go server actions at your disposal. The extension is also a replacement of the old deprecated Dreamweaver Server Behaviors and it will give you the power of new modern server connectivity.

The desired workflow, containing your server actions, is executed via Ajax on any page you choose to activate it on. You can display its output or choose to bind it with one of our other extensions. For rich client side data rendering you could bind it with HTML5 Data Bindings, the DMXzone Database Connector PHP/ASP and DMXzone Database Updater PHP/ASP and achieve extensive database connectivity and mutations.

Features in Detail

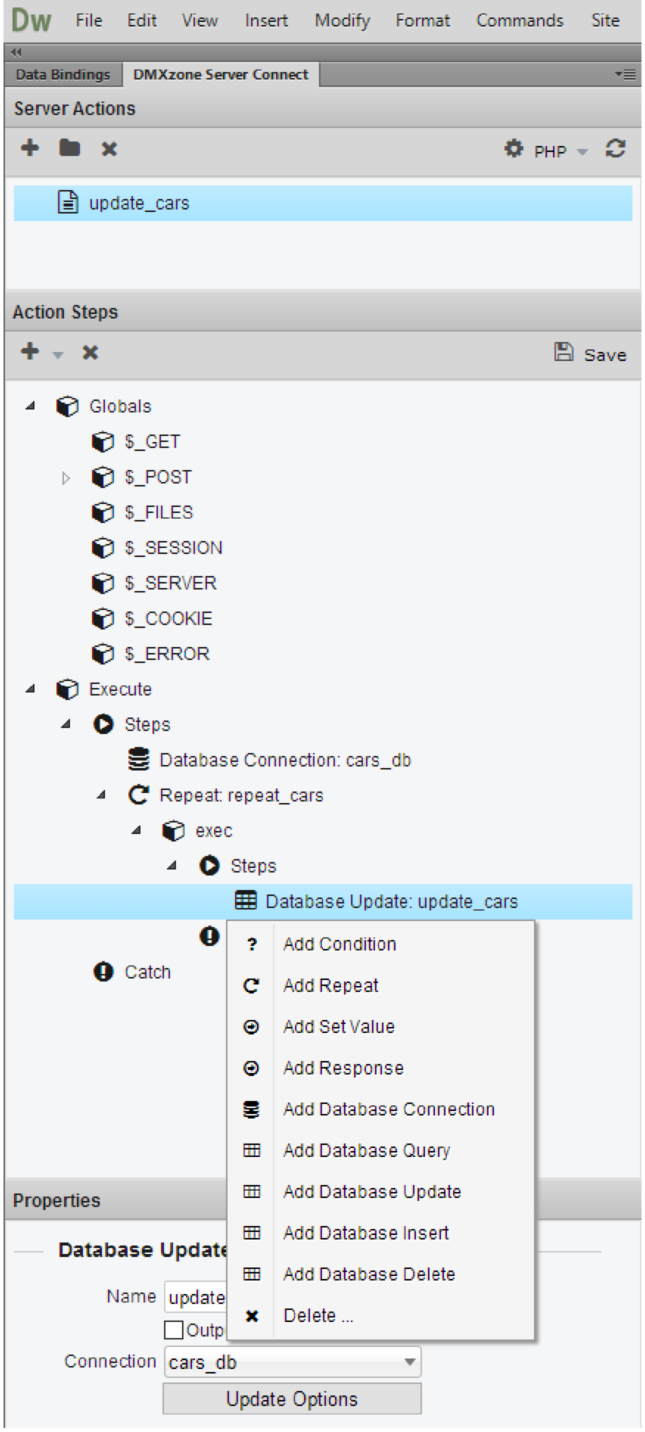
DMXzone Server Connect Workflow

Thanks to the Dreamweaver configuration and our HTML5 Data Bindings you are able to build dynamic client side pages and single page apps. These could also connect to various data sources, such as public feeds or private databases.
However there is still one issue, those used data sources were limited to single server actions and basic data feeds. Complicated Server Actions workflow was not possible. This is where DMXzone Server Connect comes in.

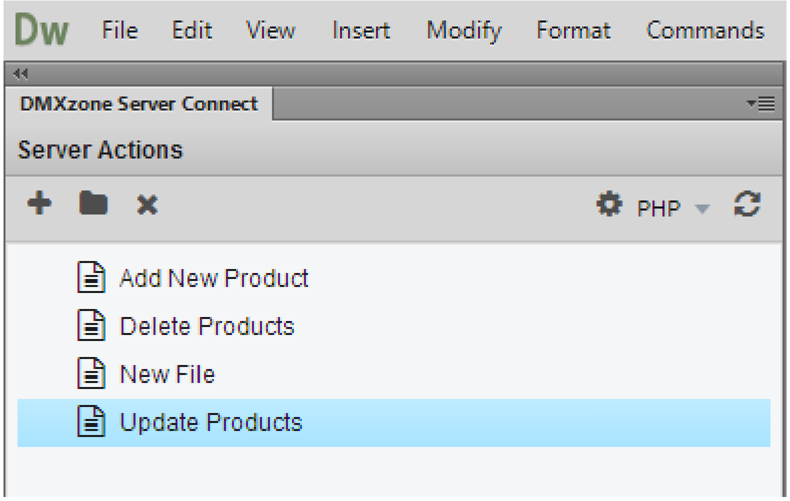


DMXzone Server Connect Action Builder

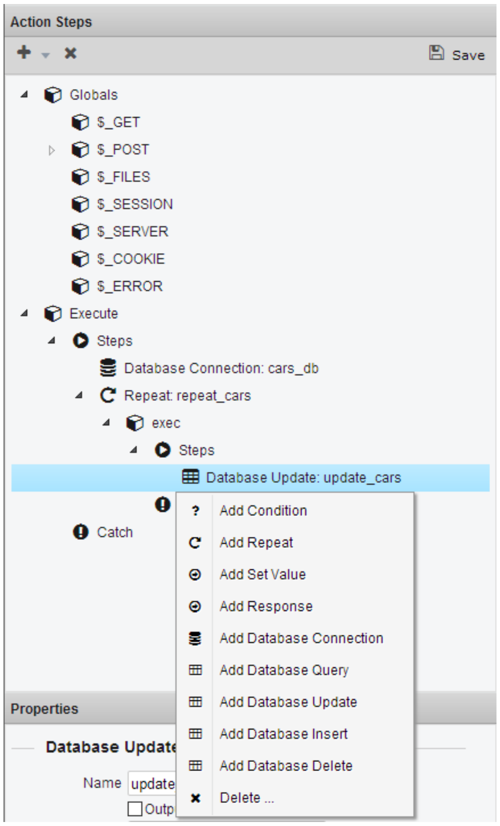
A visual way of building server actions workflow. Add any available action in a specific order, loop through them, add conditional execution and produce output.



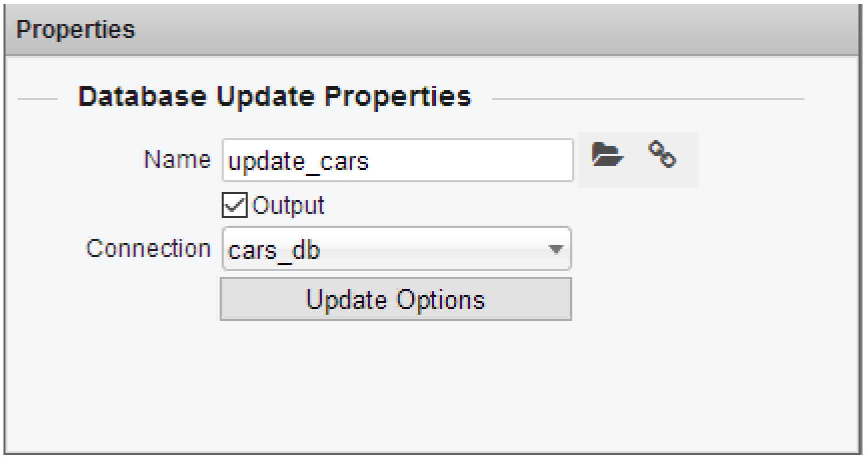
- **Server Connect Action Files** - List of all the available Server Connect Action Files for the current website in Dreamweaver. Action Files can be organized in folders for logical hierarchy.



- **Server Connect Action Steps** - A list of Action Steps that are executed when the Action is ran. Some other things that are available here are global input variables for Server Side Data Bindings and a full drag & drop so users can easily reorganize the order of the Action Steps.



- **Property inspector** - A Property Inspector for each Action Step that provides access to the specific action step properties. Fields often contain value pickers for easy file access and reference to other steps or data.


Properties

Database Update Properties

Name: update_cars

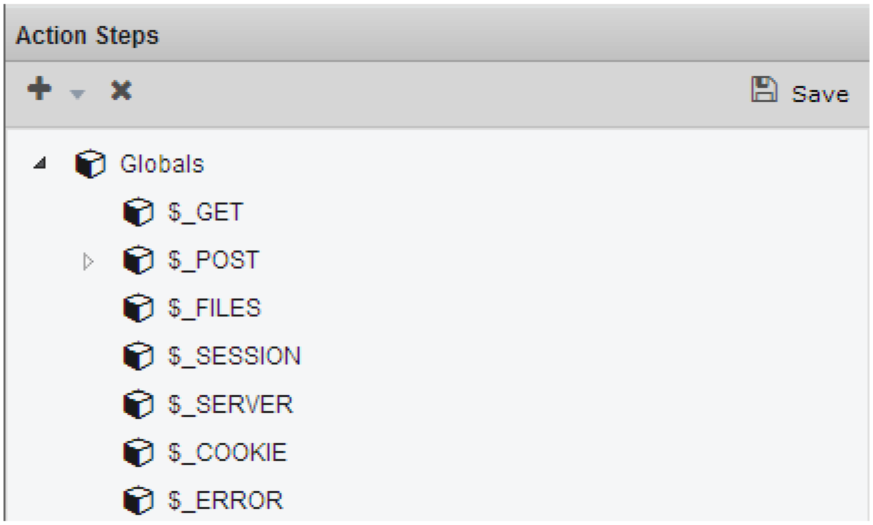
☑ Output

Connection: cars_db

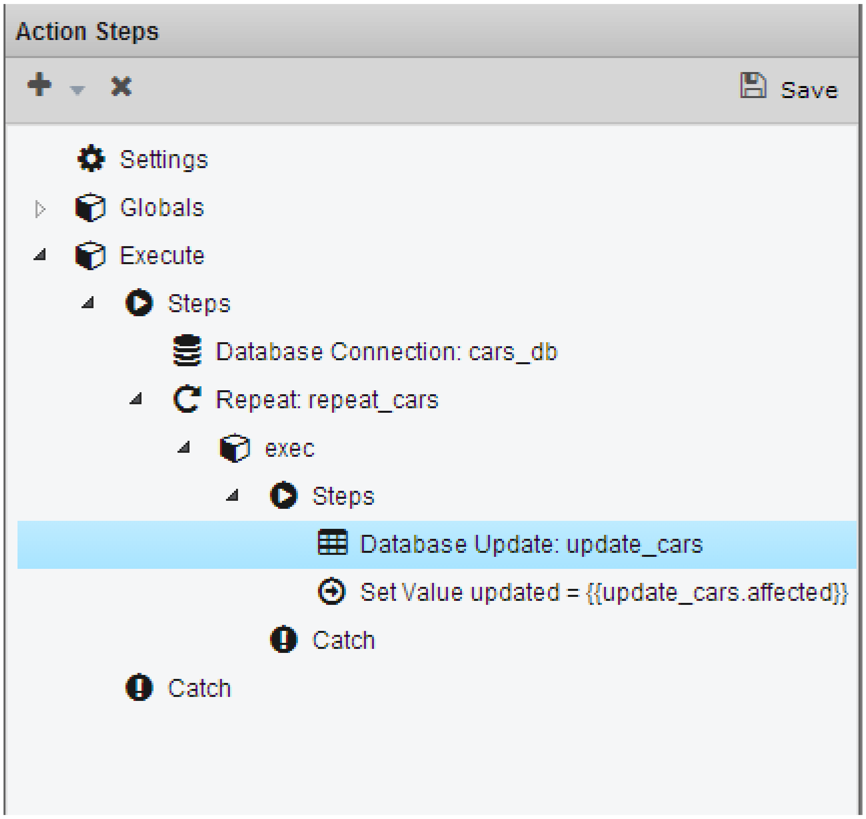
Available actions for DMXzone Server Connect

Action steps List defines the workflow of execution. It contains two main parts: Global input variables and Execution steps list.

- **Global Input Variables** – Here you can define any input data that is required for this Action Steps workflow. You can either manually define the variables or bind to a separate page containing regular HTML form.



- **Execution Steps List** – It defines the execution flow of the whole Server Connect Action. You can use many different types of actions. From Repeaters that execute multiple actions and repeat execution, conditional actions, to specific database actions, such as a query or update of databases.



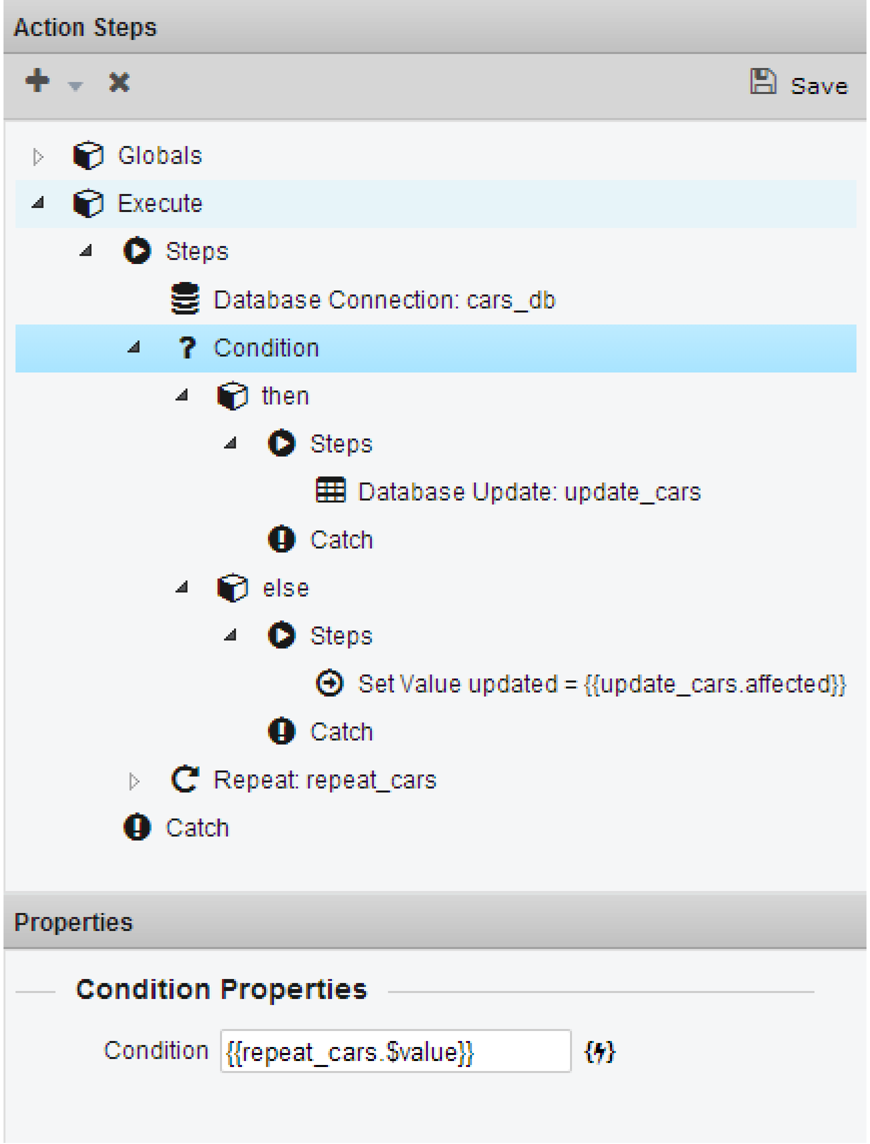
```
Settings
Globals
Execute
  Steps
    Database Connection: cars_db
    Repeat: repeat_cars
      exec
        Steps
          Database Update: update_cars
          Set Value updated = {{update_cars.affected}}
        Catch
  Catch
```

Conditional Action

With the Conditional Action step you can choose a specific expression that will be evaluated. Depending on the result, the flow of actions will go to the then or else branch.

Note that all expressions follow the JavaScript dot syntax and should also be placed within the {{...}} tags to be marked as expressions.

Also next to the global input variables placed under $_GET or $_POST each step can produce its own data that can be referenced in later steps.



Action Steps

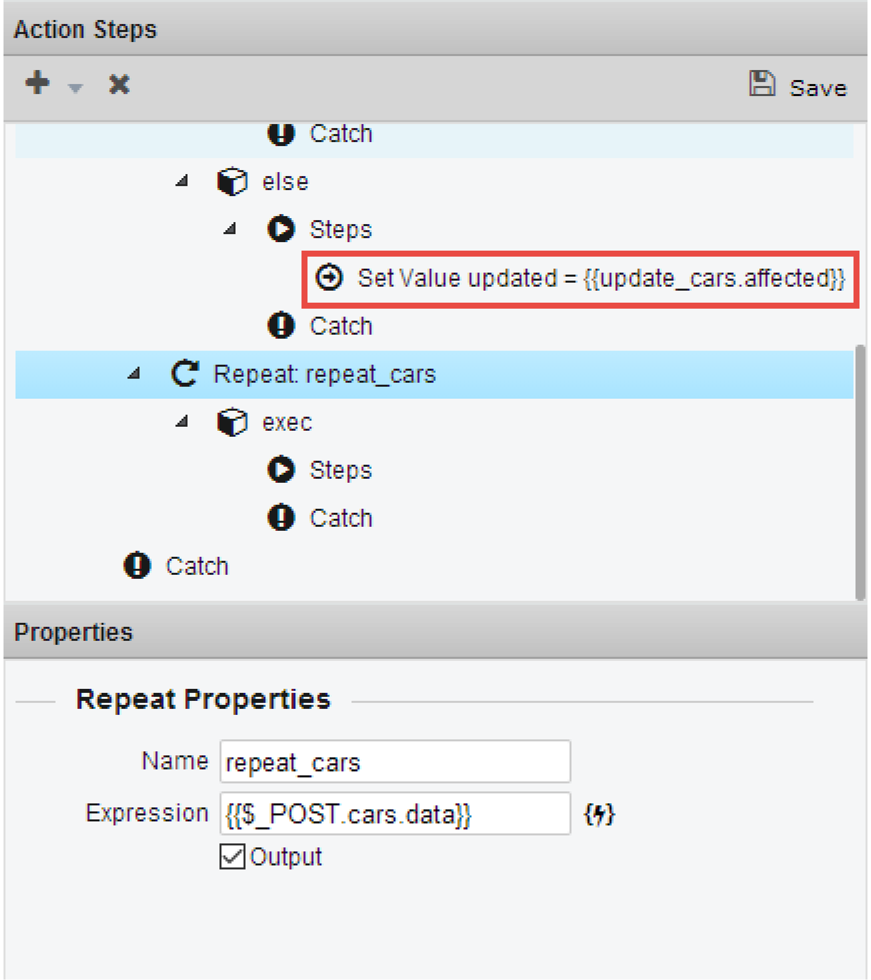
- Globals
- Execute
  - Steps
    - Database Connection: cars_db
    - Condition
      - then
        - Steps
          - Database Update: update_cars
        - Catch
      - else
        - Steps
          - Set Value updated = {{update_cars.affected}}
        - Catch
    - Repeat: repeat_cars
  - Catch

Properties

Condition Properties

Condition: {{repeat_cars.$value}}

- **Set Value Action** - You can manually create a variable by assigning it a value with the Set Value step.

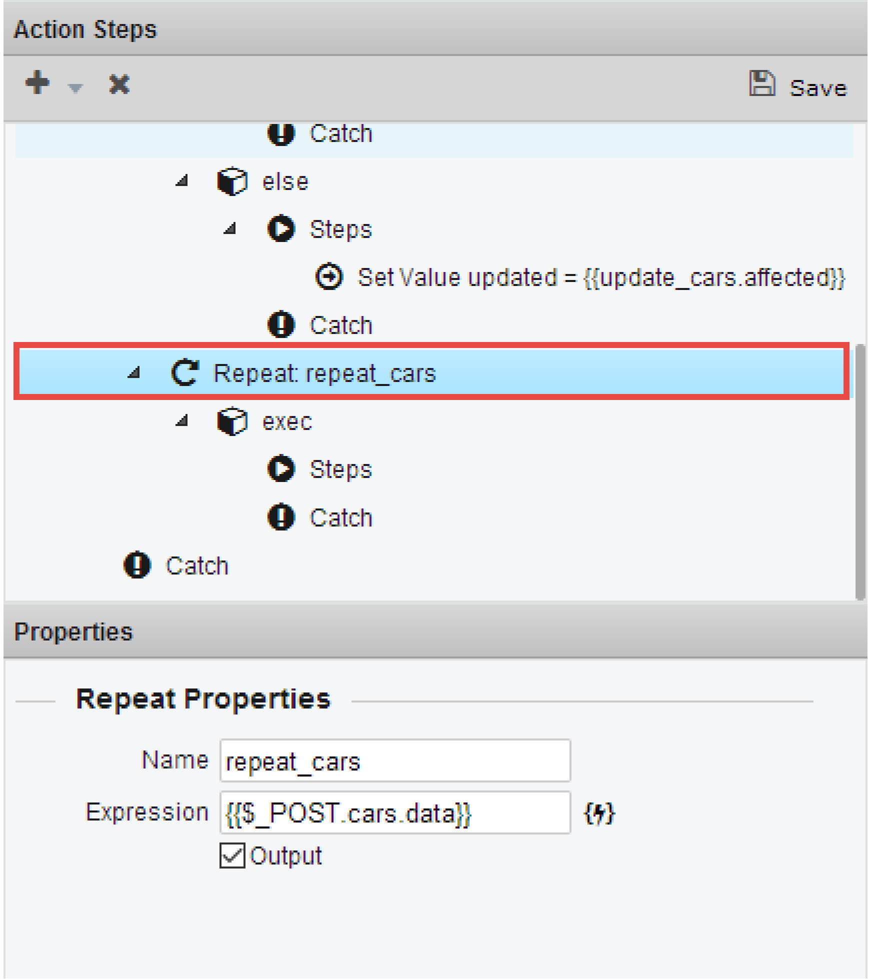


Name: repeat_cars

Expression: {{$_POST.cars.data}}

☑ Output

- **Repeat Action** - The Repeat step allows you to repeat a set of actions a number of times. How often is defined by a specified expression. You can also choose to repeat a collection or an array expression. This way you can iterate through some array input or through the records of a database query for example.



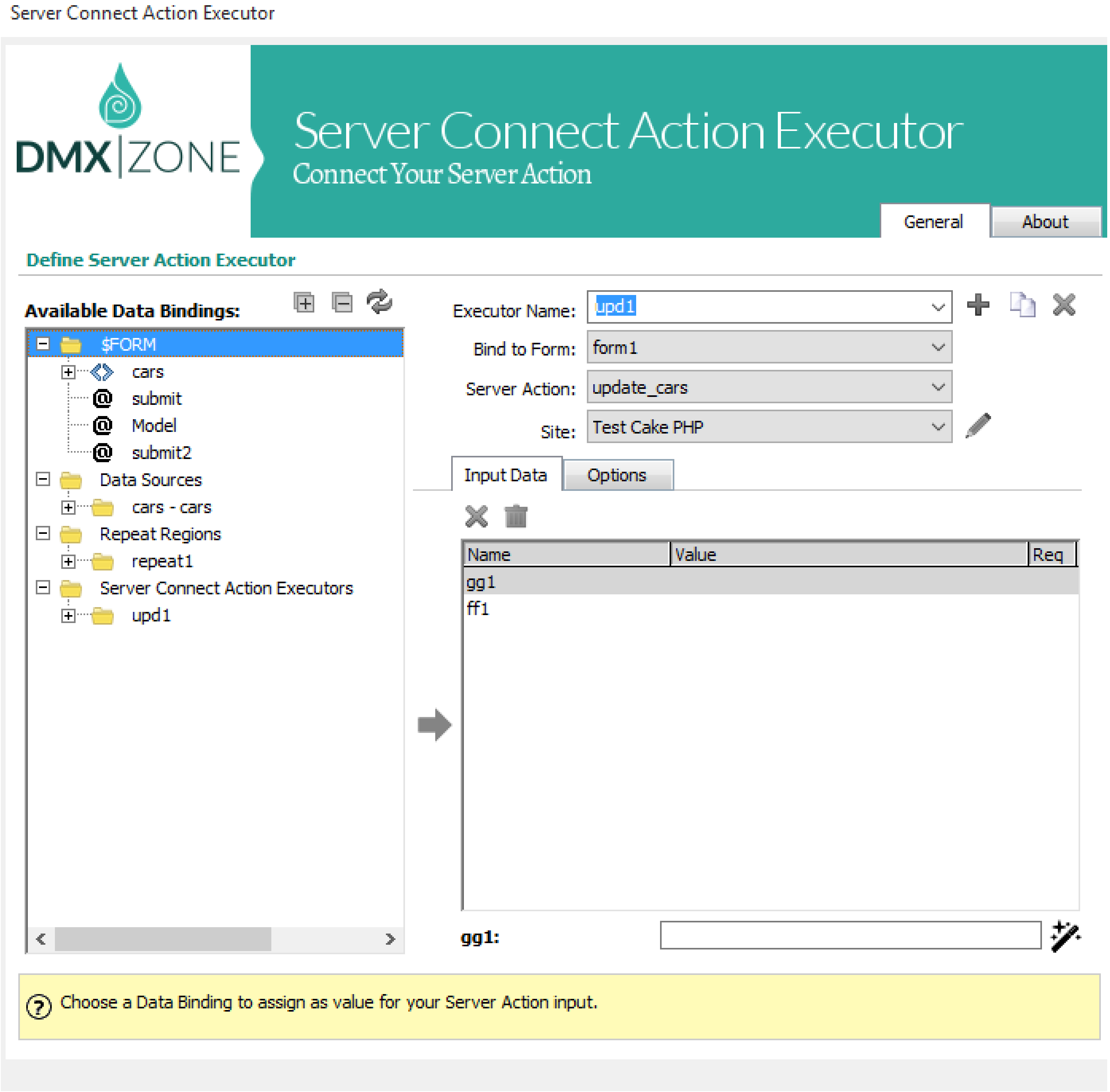
Name: repeat_cars

Expression: {{$_POST.cars.data}}

☑ Output

Executing the DMXzone Server Connect Action workflow

You can easily execute the defined Server Connect Action through Ajax by using the DMXzone Server Action Executor command in Dreamweaver. You can also define when the Server Connect Action should be executed and what data it should receive. Usually, the executor is bound to a form and its data is then submitted. Additional data can also be specified.



Server Connect Action Executor

Server Connect Action Executor
Connect Your Server Action

Executor Name: upd1

Bind to Form: form1

Server Action: update_cars

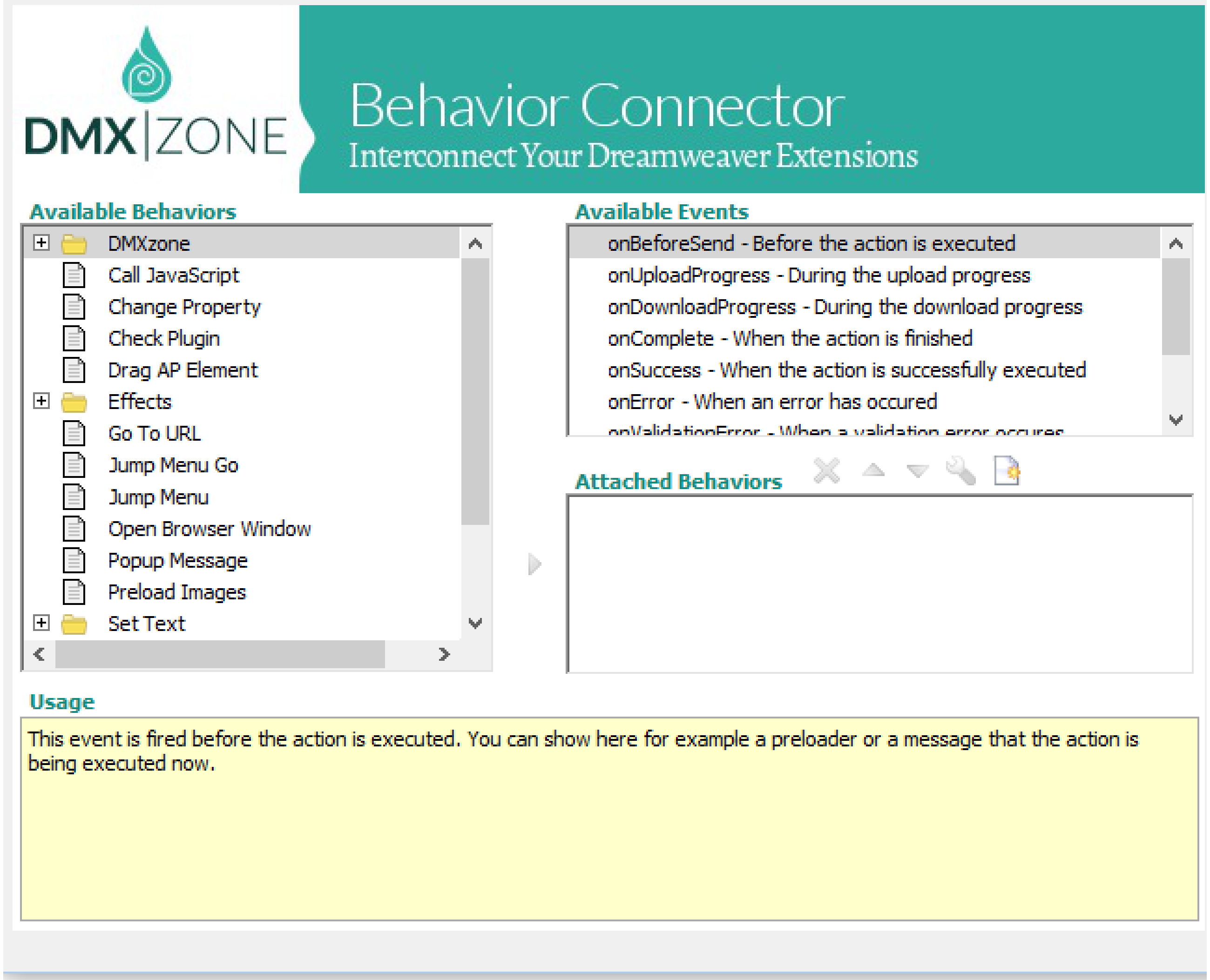
Site: Test Cake PHP

Name	Value	Req
gg1		
ff1		

gg1:

Choose a Data Binding to assign as value for your Server Action input.

- **DMXzone Behavior Connector** - Fire an event before an action is executed.

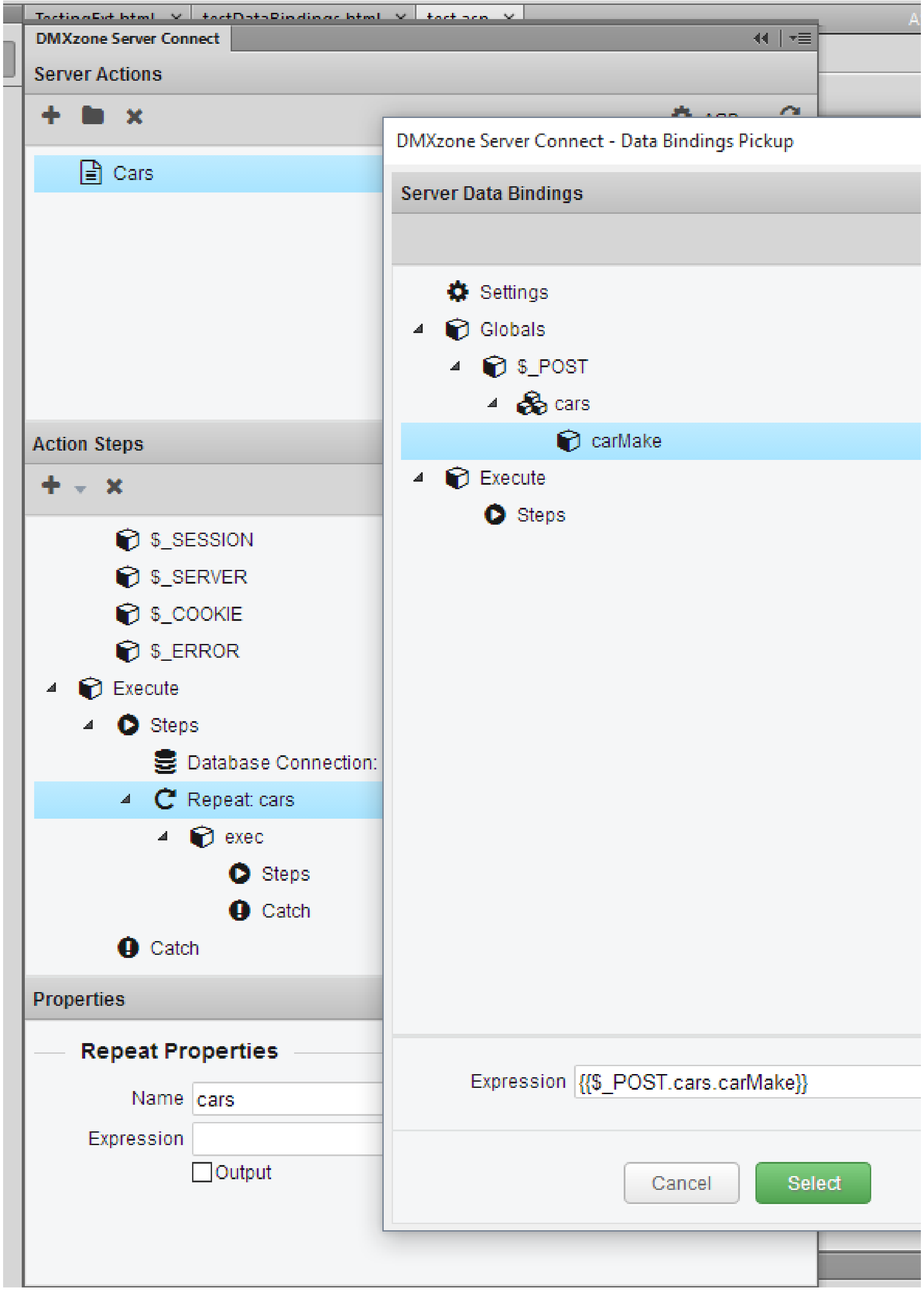


onBeforeSend - Before the action is executed
onUploadProgress - During the upload progress
onDownloadProgress - During the download progress
onComplete - When the action is finished
onSuccess - When the action is successfully executed
onError - When an error has occured
onValidationError - When a validation error occures

This event is fired before the action is executed. You can show here for example a preloader or a message that the action is being executed now.

Server Data Bindings Picker

The Server Data Bindings Picker allows the user to easily pick data variables for the different action steps. Only Data Variables available from the previous steps are shown.



Name: cars

Expression:

☐ Output

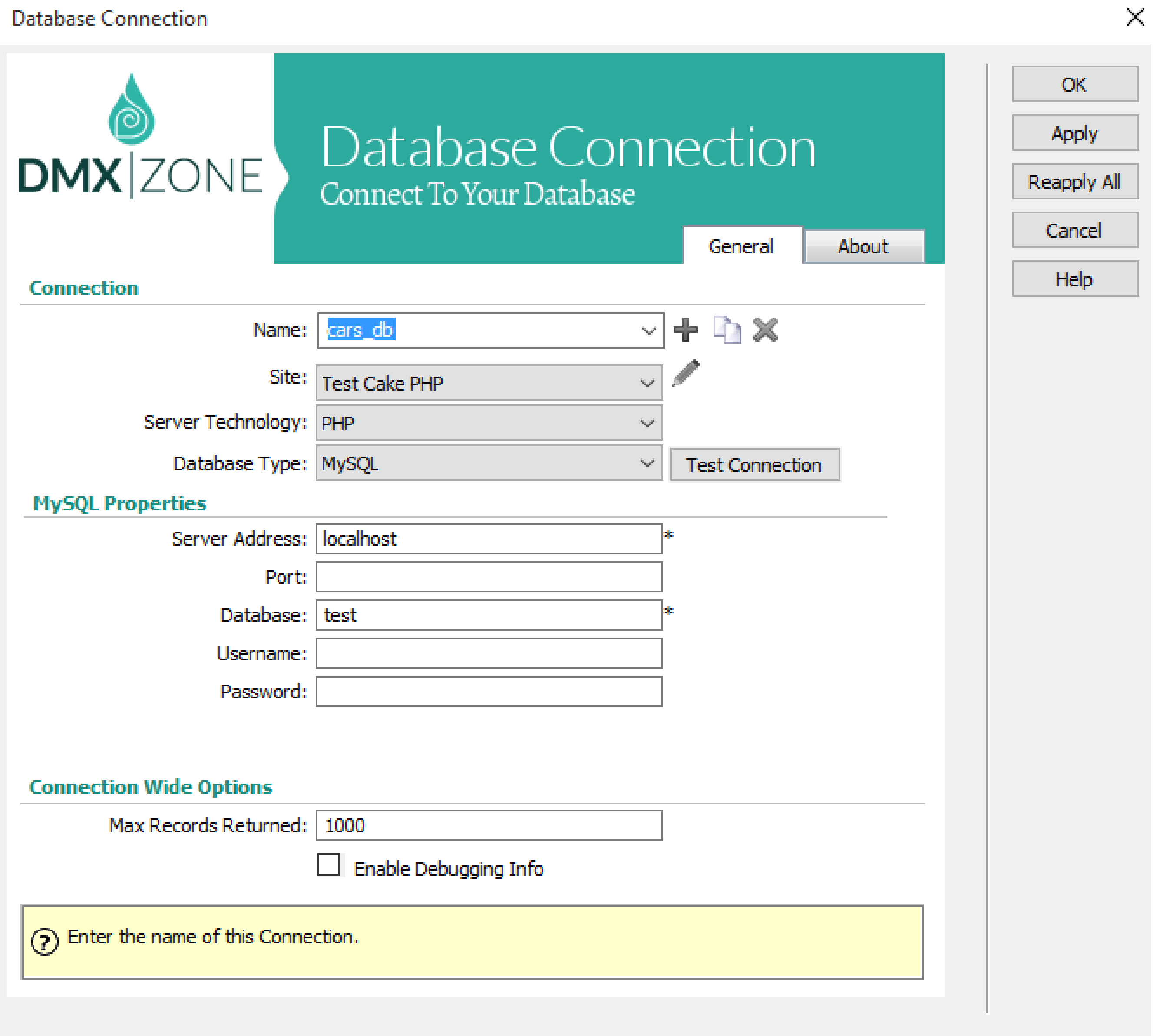
Expression: {{$_POST.cars.carMake}}

Database Connectivity

The database connectivity is done by the DMXzone Database Connector and the DMXzone Database Updater for cross database compatibility. Those two extensions integrate nicely in the DMXzone Server Connect.

Defining a database connection is easy with the DMXzone Database Connector Connection Builder. It supports a wide range of well-known databases on both ASP and PHP server platforms.

- **A built-in connection wizard** - It lets you define and manage your connections to all well-known databases. It is possible to choose from ready to go connections for MySQL, Microsoft SQL Server, MS Access and many others.



Connection

Name: cars_db

Site: Test Cake PHP

Server Technology: PHP

Database Type: MySQL

MySQL Properties

Server Address: localhost *

Port:

Database: test *

Username:

Password:

Connection Wide Options

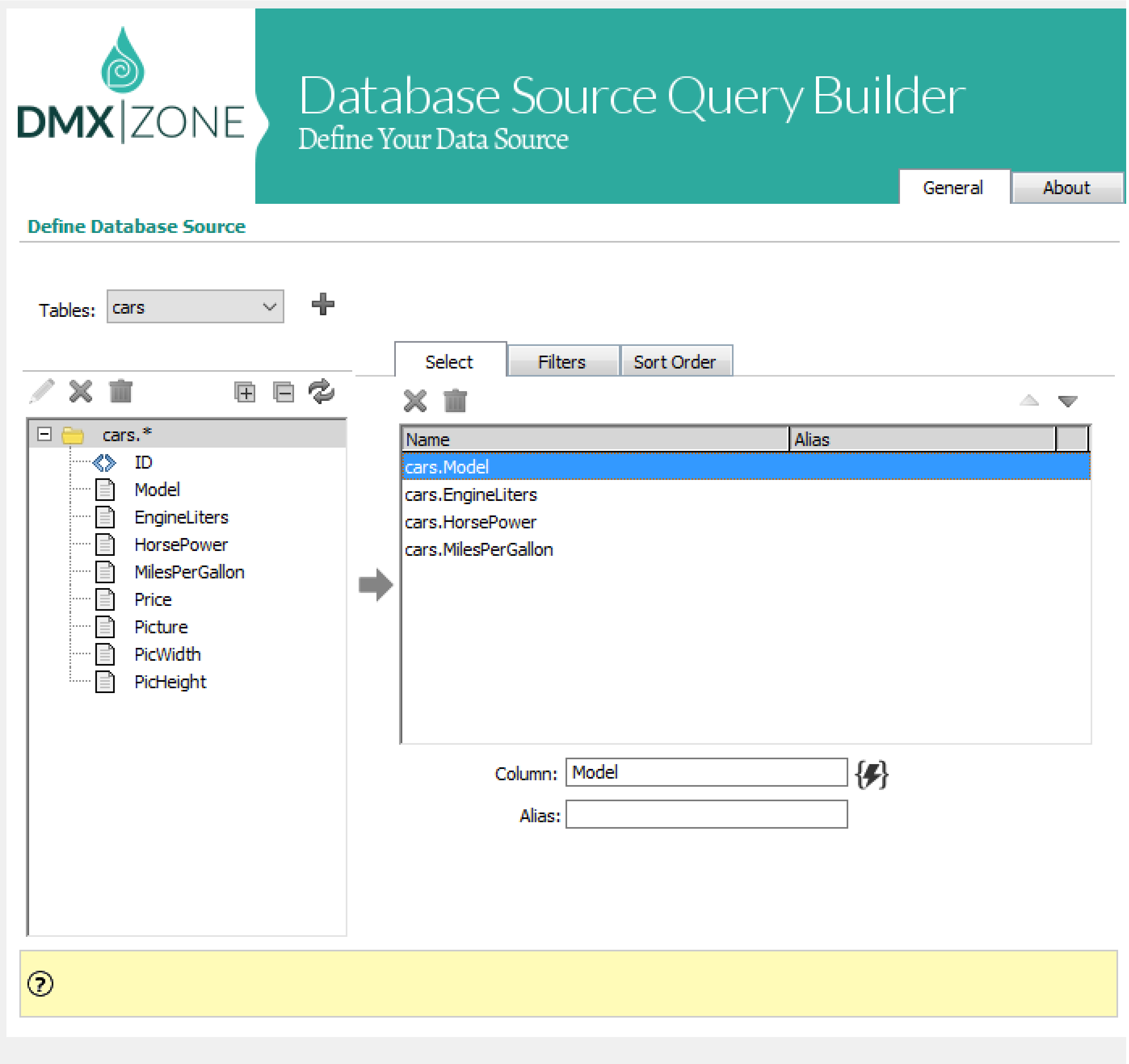
Max Records Returned: 1000

☐ Enable Debugging Info

Enter the name of this Connection.

- **Database source query builder** - The DMXzone Database Connector also includes an amazing Database Source Query Builder that will allow you to build fully visual an advanced query on your database source without any SQL knowledge. It even supports advanced filtering and SQL JOINs options and lets you directly view and test your database.

Database Connector



Tables: cars

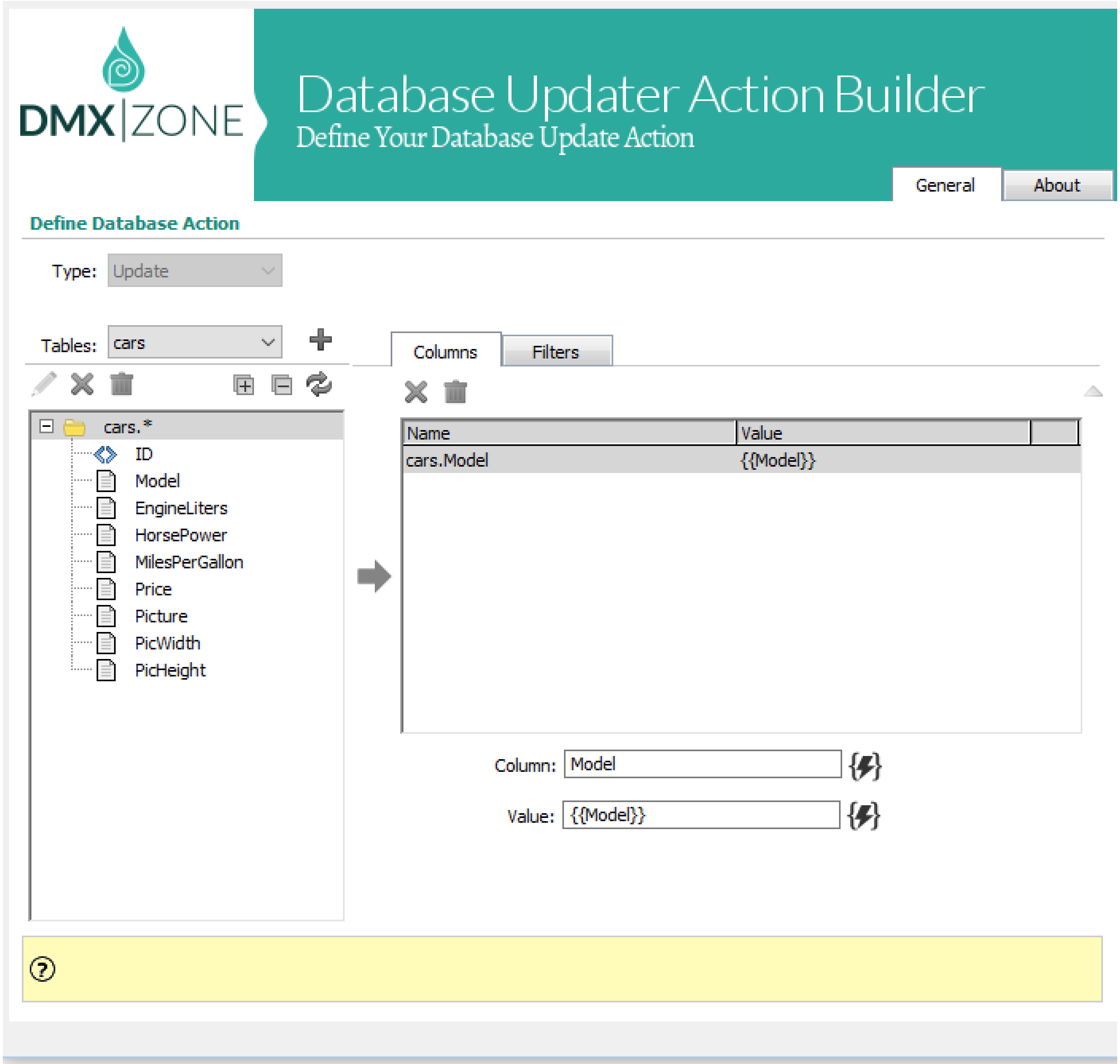
Name	Alias
cars.Model	
cars.EngineLiters	
cars.HorsePower	
cars.MilesPerGallon	

Column: Model

Alias:

- **Database updater action builder** - The Database Updater Action Builder lets you visually define any database update action. Database actions are executed by the so called Database Action Executors fully thought AJAX, without any page refresh!

Database Updater



Database Updater Action Builder

Define Your Database Update Action

Define Database Action

Type: Update

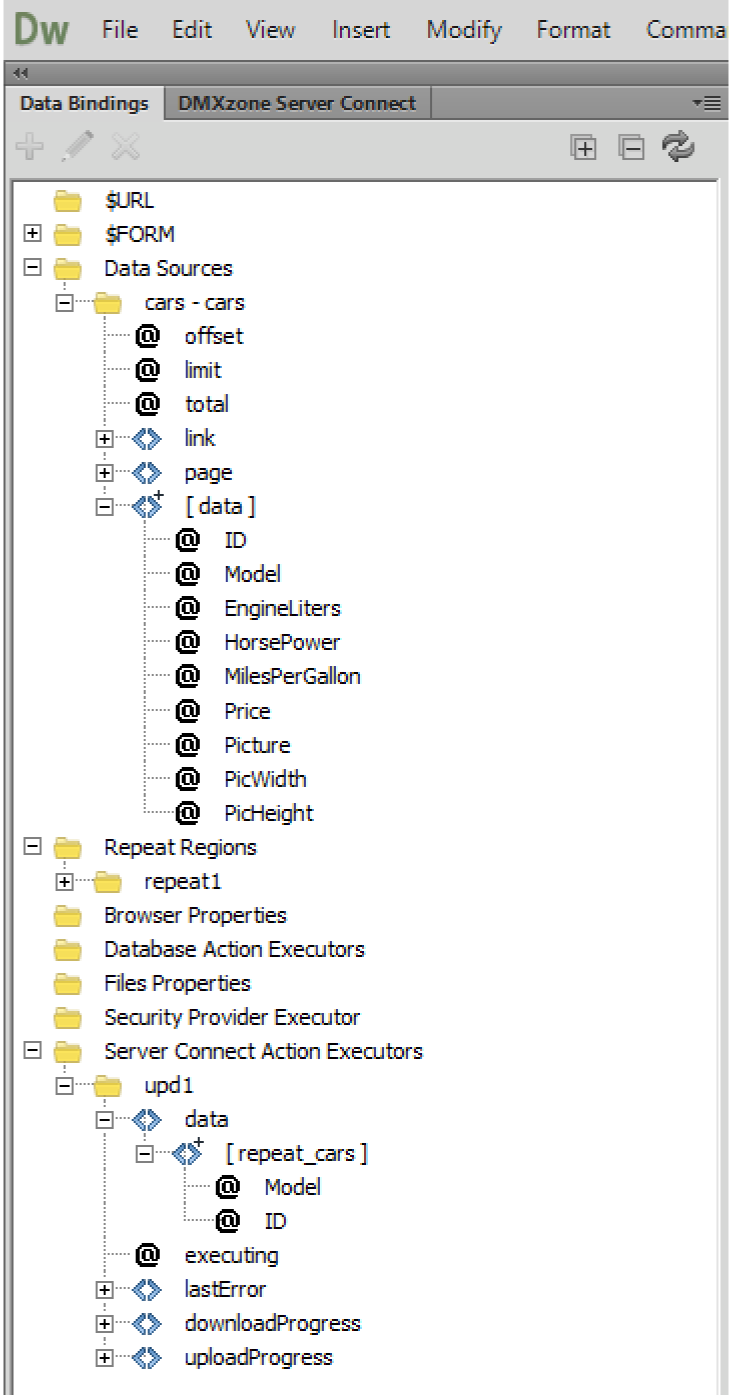
Tables: cars

Name	Value
cars.Model	{{Model}}

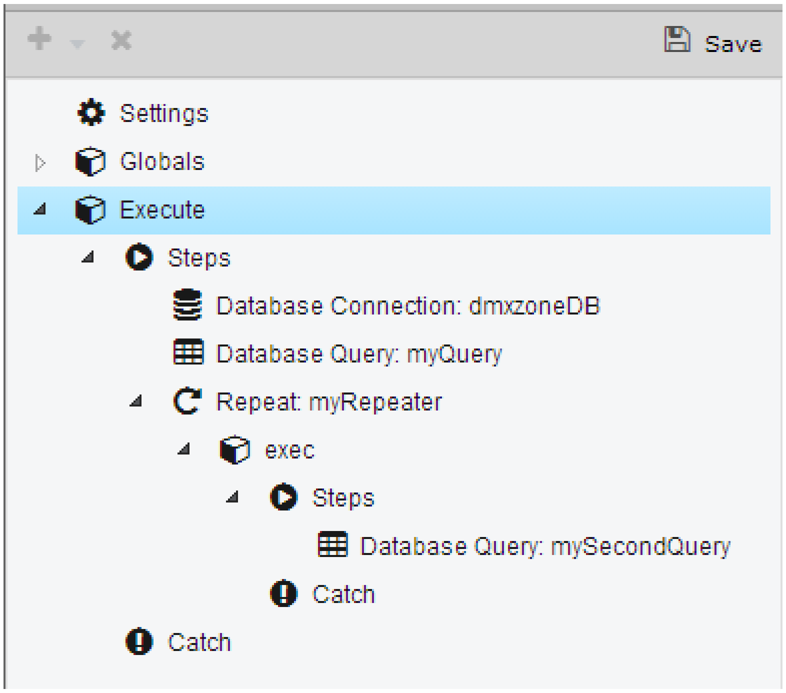
Column: Model

Value: {{Model}}

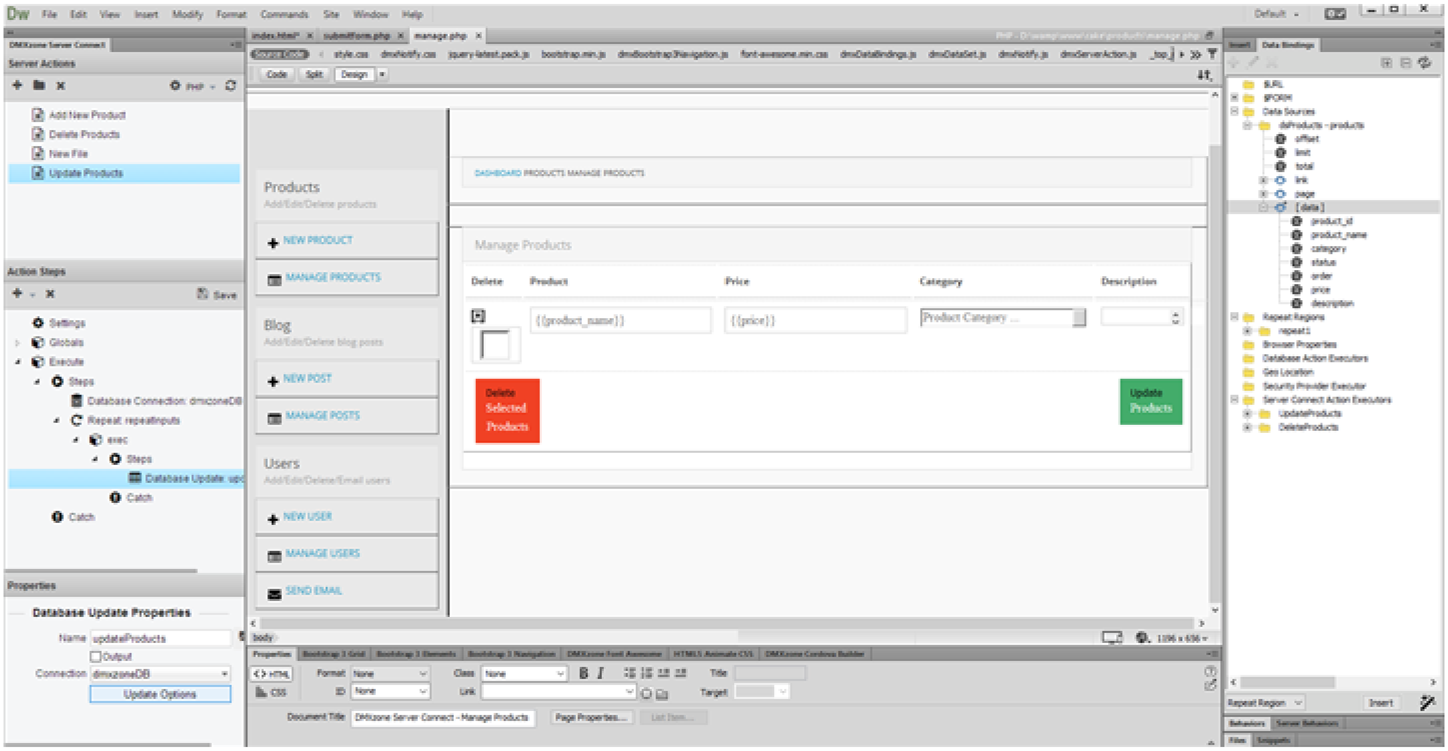
- **HTML5 Data Bindings support** - All the client side data can be easily browsed by the dedicated HTML5 Data Bindings floating panel. All used Server Connect Action Executors are also listed there.



- **Support for nested database queries** - Now thanks to DMXzone Server Connect you can execute multiple nested queries and return their result at once.



- **Support for multi records forms** - now with the new HTML5 Data Bindings you can have dynamic forms with multiple repeating elements that will be submitted as multi record form to the DMXzone Server Connect Action handler. **Click the image for larger preview.**



Database Update Properties

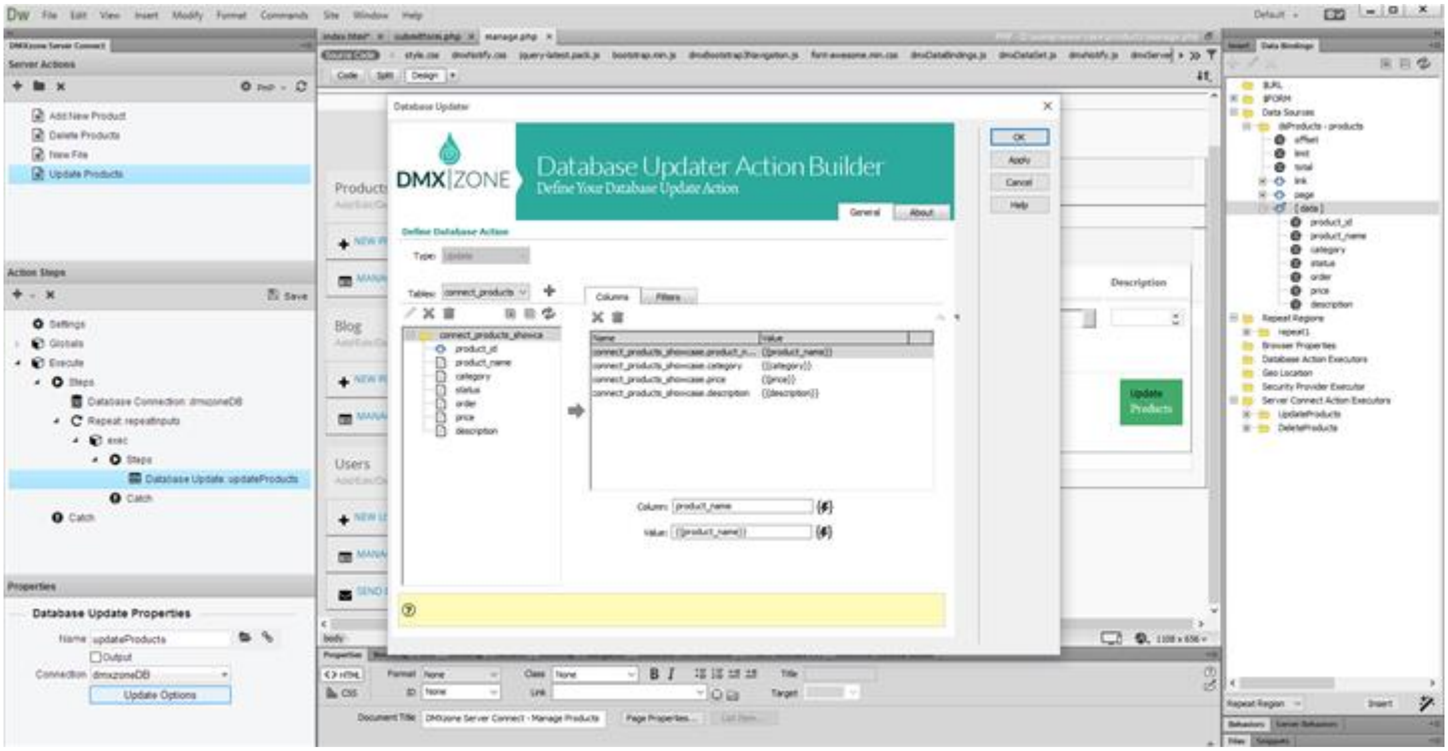
Name updateProducts

☐ Output

Connection dmxzoneDB

Document Title DMXzone Server Connect - Manage Products

- **Multiple Records Update** - with DMXzone Server Connect support for multi records forms, now you can process those forms so you can have your own mass update of more than one record. **Click the image for larger preview.**



Type: Update

Table: connect_products

Name	Value
connect_products_showcase.product_n...	{{product_name}}
connect_products_showcase.category	{{category}}
connect_products_showcase.price	{{price}}
connect_products_showcase.description	{{description}}

Column: product_name

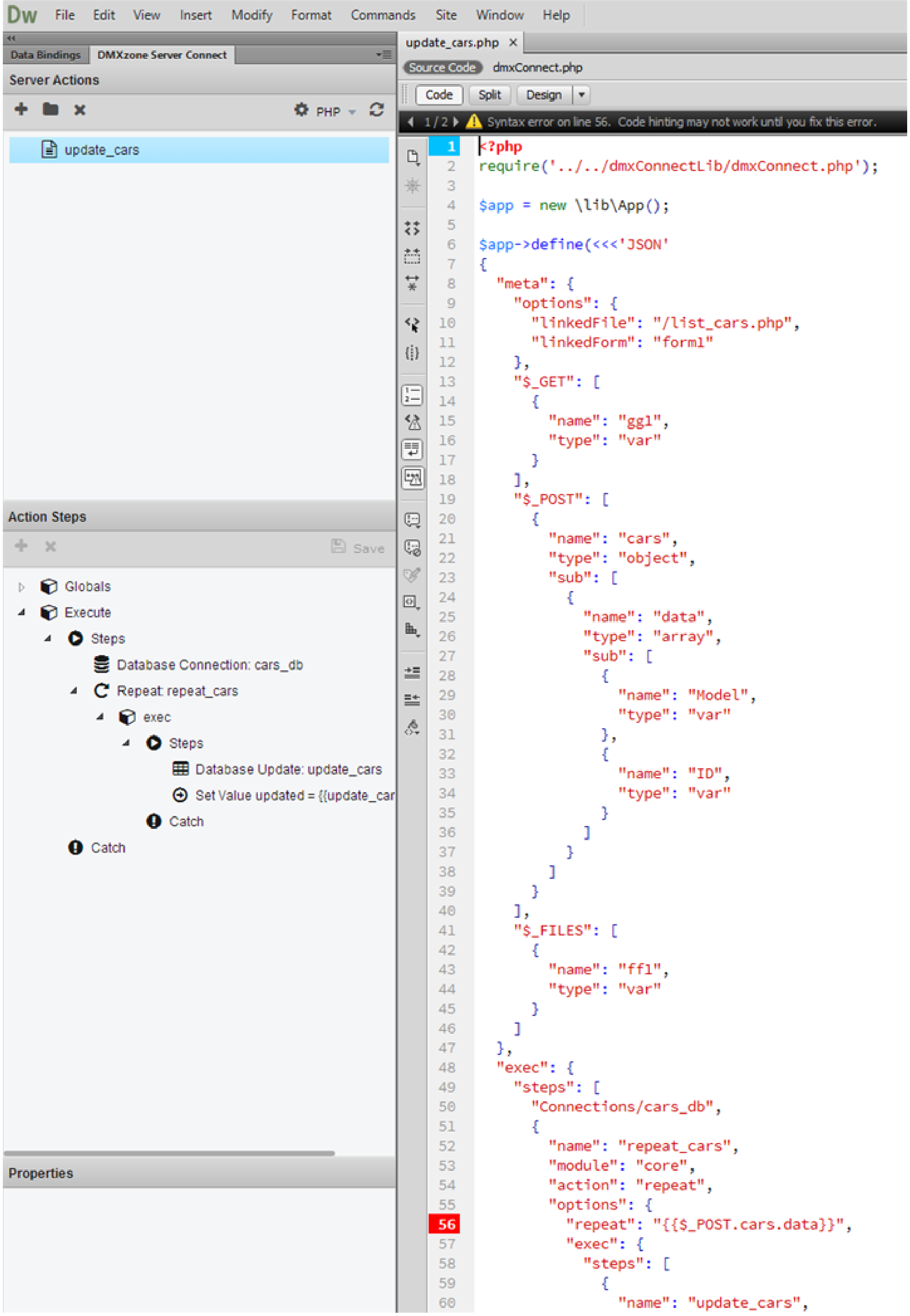
Value: {{product_name}}

Name: updateProducts

Connection: dmxzoneDB

Document Title: DMXzone Server Connect - Manage Products

Server Actions workflow and its JSON representation



Syntax error on line 56. Code hinting may not work until you fix this error.

```
<?php
require('../../dmxConnectLib/dmxConnect.php');

$app = new \lib\App();

$app->define(<<<'JSON'
{
  "meta": {
    "options": {
      "linkedFile": "/list_cars.php",
      "linkedForm": "form1"
    },
    "$_GET": [
      {
        "name": "gg1",
        "type": "var"
      }
    ],
    "$_POST": [
      {
        "name": "cars",
        "type": "object",
        "sub": [
          {
            "name": "data",
            "type": "array",
            "sub": [
              {
                "name": "Model",
                "type": "var"
              },
              {
                "name": "ID",
                "type": "var"
              }
            ]
          }
        ]
      }
    ],
    "$_FILES": [
      {
        "name": "ff1",
        "type": "var"
      }
    ]
  },
  "exec": {
    "steps": [
      "Connections/cars_db",
      {
        "name": "repeat_cars",
        "module": "core",
        "action": "repeat",
        "options": {
          "repeat": "{{$_POST.cars.data}}",
          "exec": {
            "steps": [
              {
                "name": "update_cars",
```

Before you begin

Make sure to setup your site in Dreamweaver first before creating a page. If you don't know how to do that please read this.

Installing the extension

Read the tutorial at http://www.dmxzone.com/go/4671/how-to-install-extensions-from-dmxzone/

Login, download the product, and look up your serial at http://www.dmxzone.com/user/products

Get the latest DMXzone Extension Manager at
http://www.dmxzone.com/go/22670/dmxzone-extension-manager-for-dreamweaver

Open DMXzone Extension Manager and go to your Purchased products tab and click install. If you have a subscription, the extension is to be found and installed in the Explore tab.

Get the latest extension manager from Adobe at
http://www.adobe.com/exchange/em_download/

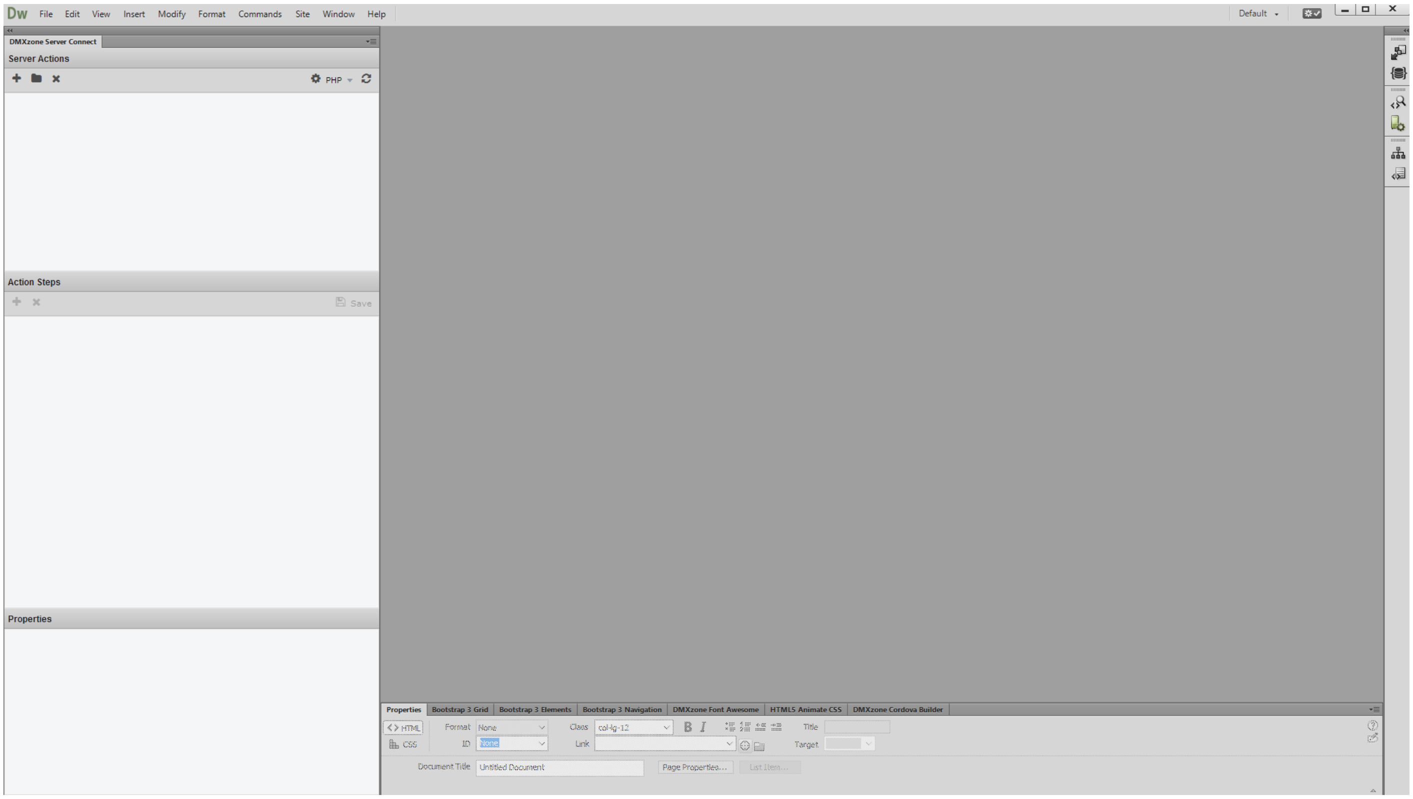
Open the .zxp file if you're using Dreamweaver CC or the .mxp file for Dreamweaver CS+ with Adobe Extension manager and install the extension.

If you experience any problems contact our Live Support!

The Basics: Accessing Server Actions Builder Panel

In this tutorial we will show you how to access and enable DMXzone Server Connect in Dreamweaver.

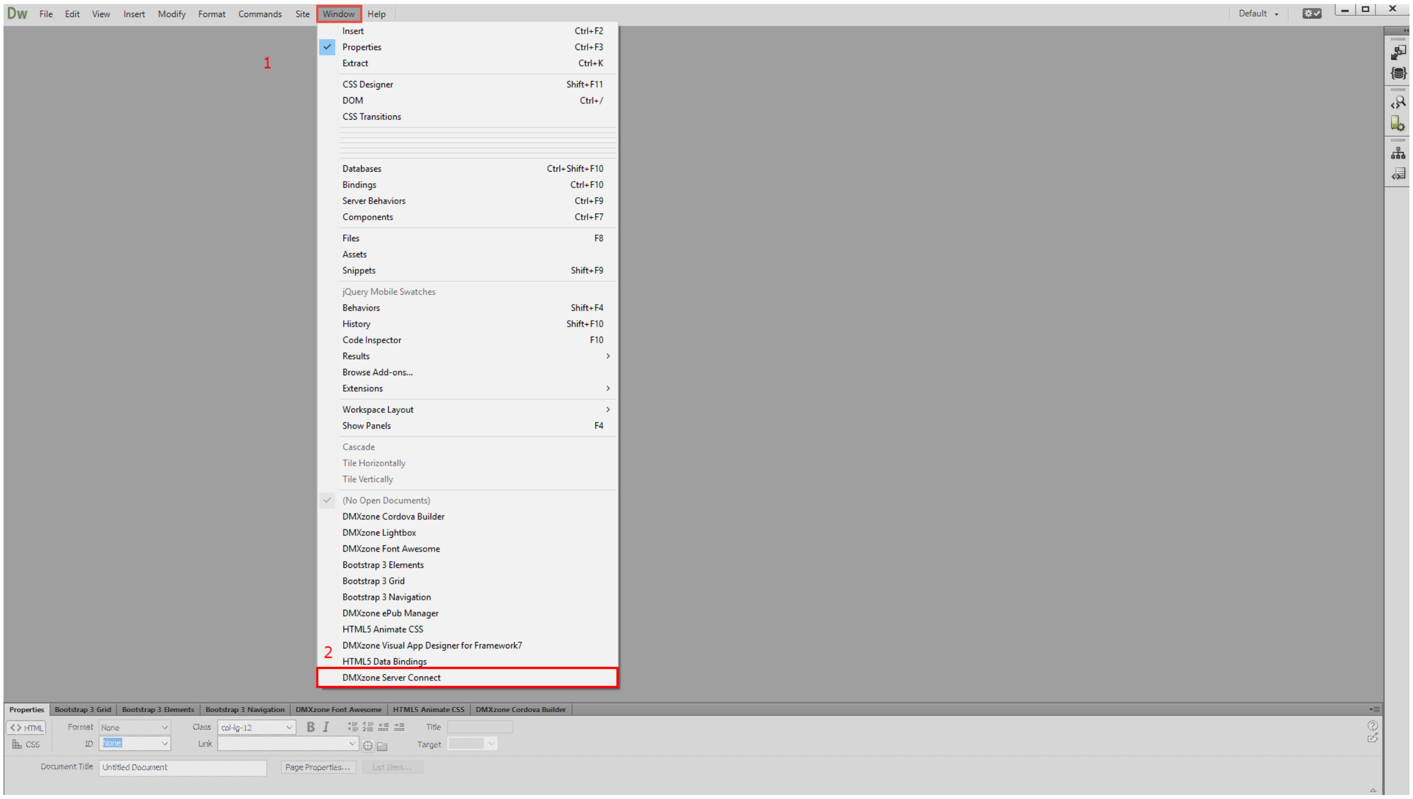
*You can also check out the video.



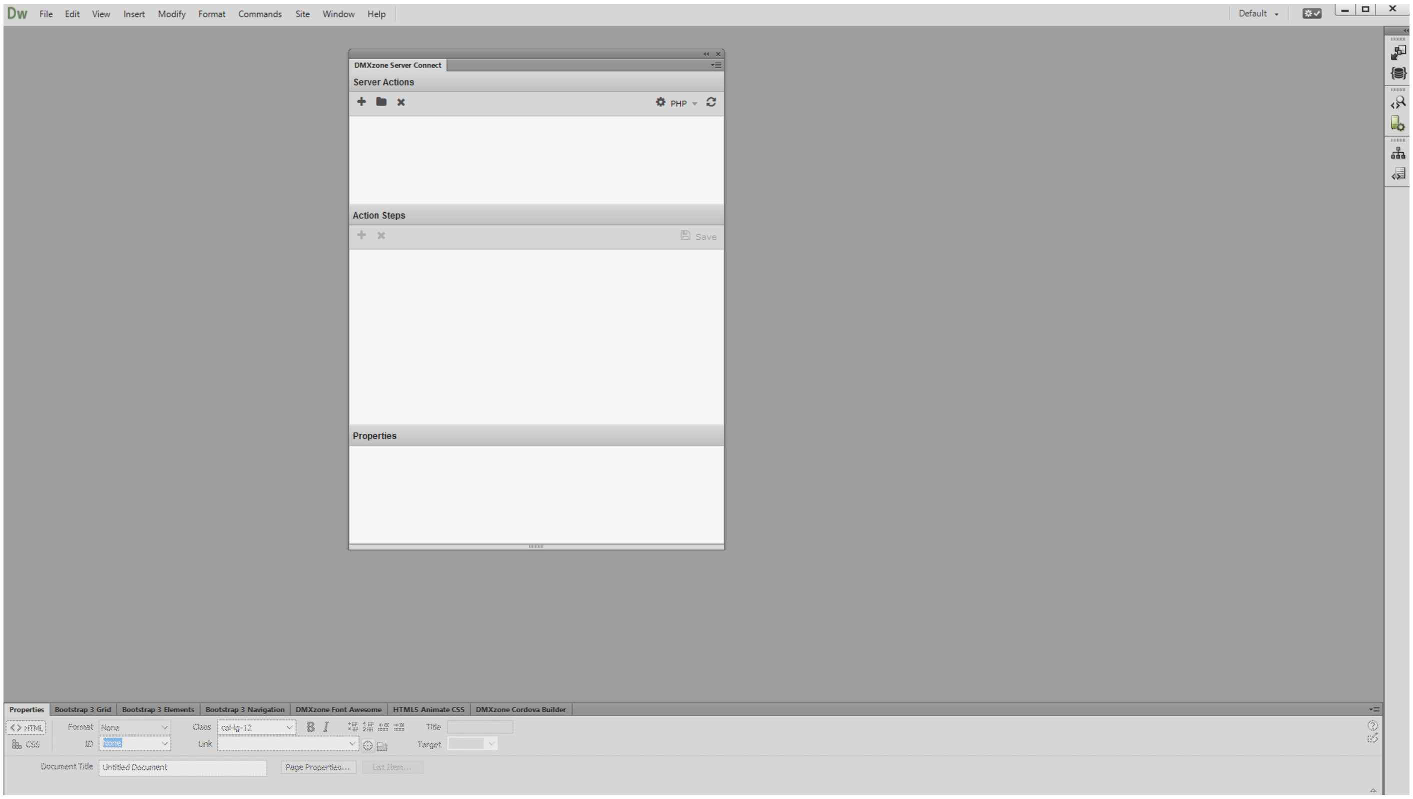
Format None
ID None
Class col-lg-12
Link
Document Title Untitled Document

How to do it

1. In order to access DMXzone Server Connect, make sure that you have it installed in DMXzone Extension Manager first. Then open Dreamweaver and from the **window menu (1)** choose **DMXzone Server Connect (2)**.

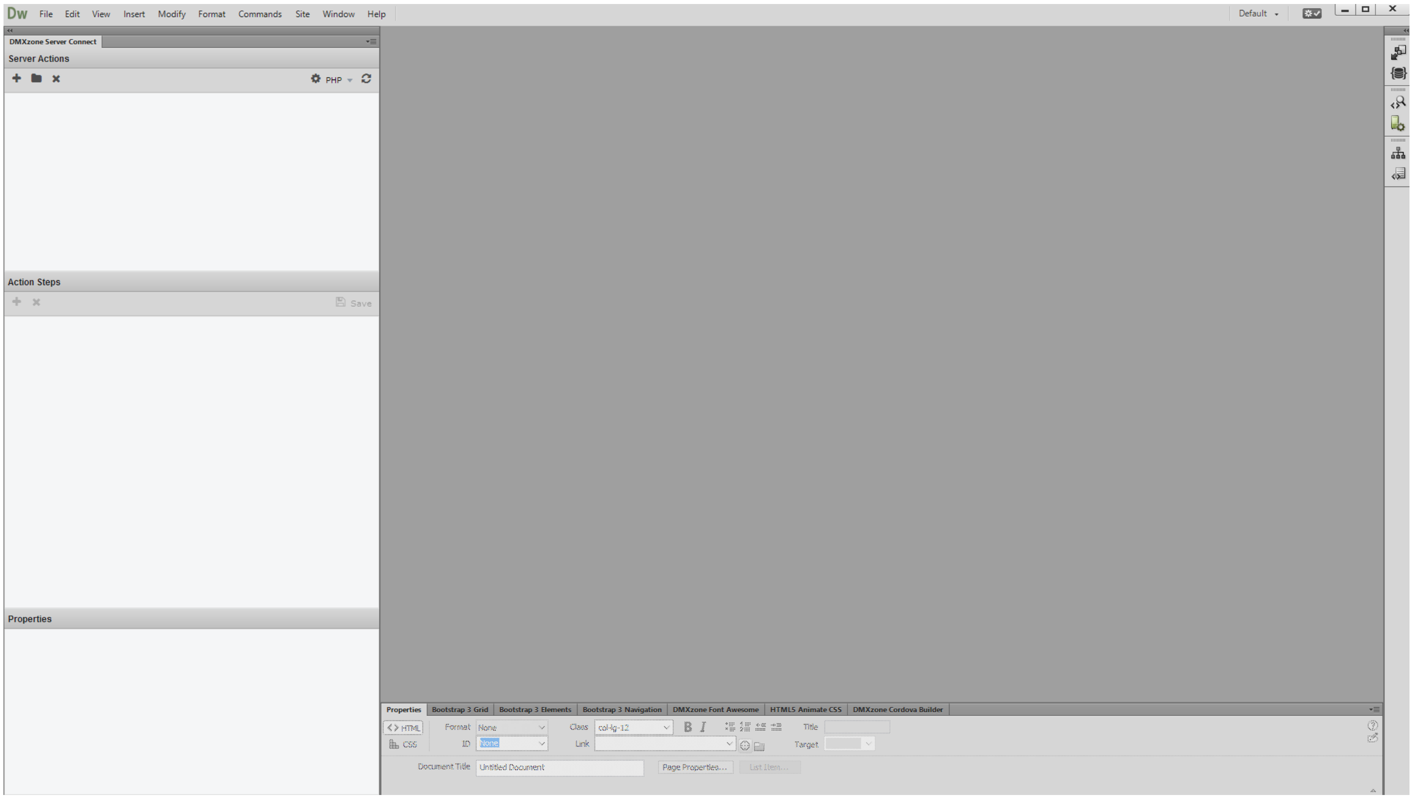


2. The DMXzone Server Connect server action builder panel is now enabled.



Format None
ID None
Class col-lg-12
Link
Title
Target
Document Title Untitled Document

3. For the best results, please drag and dock the panel to the left side of Dreamweaver. Now you're ready to start working with DMXzone Server Connect.

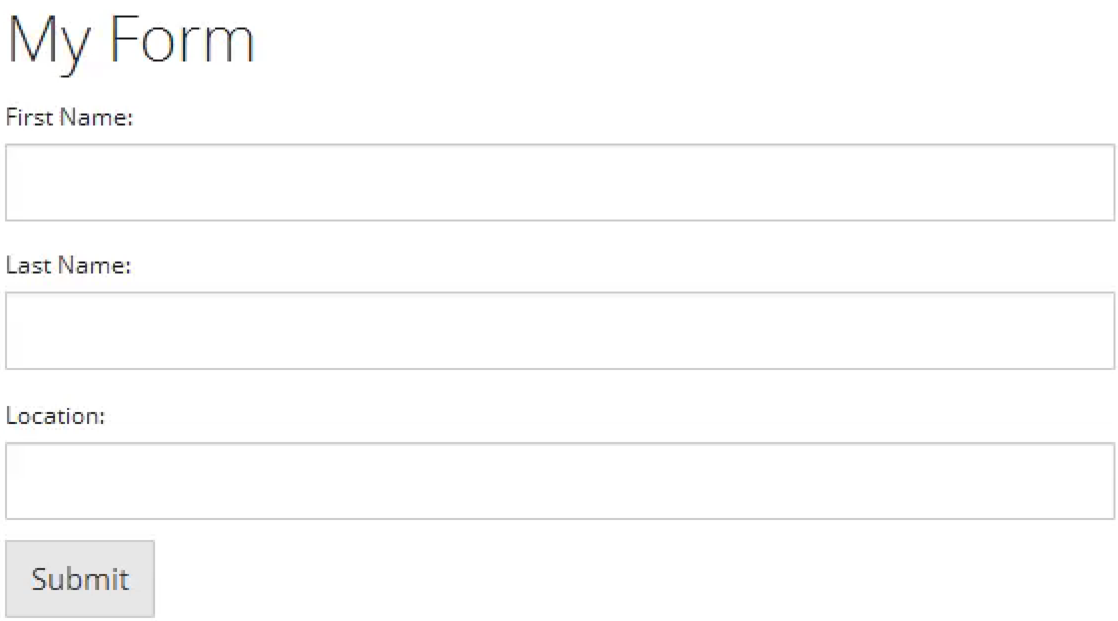


Format: None
Class: col-lg-12
ID: None
Link:
Title:
Target:
Document Title: Untitled Document

Advanced: Adding Form Inputs to DMXzone Server Connect

In this tutorial we will show you how to add your form fields to the global input variables of DMXzone Server Connect.

*You can also check out the video.



My Form

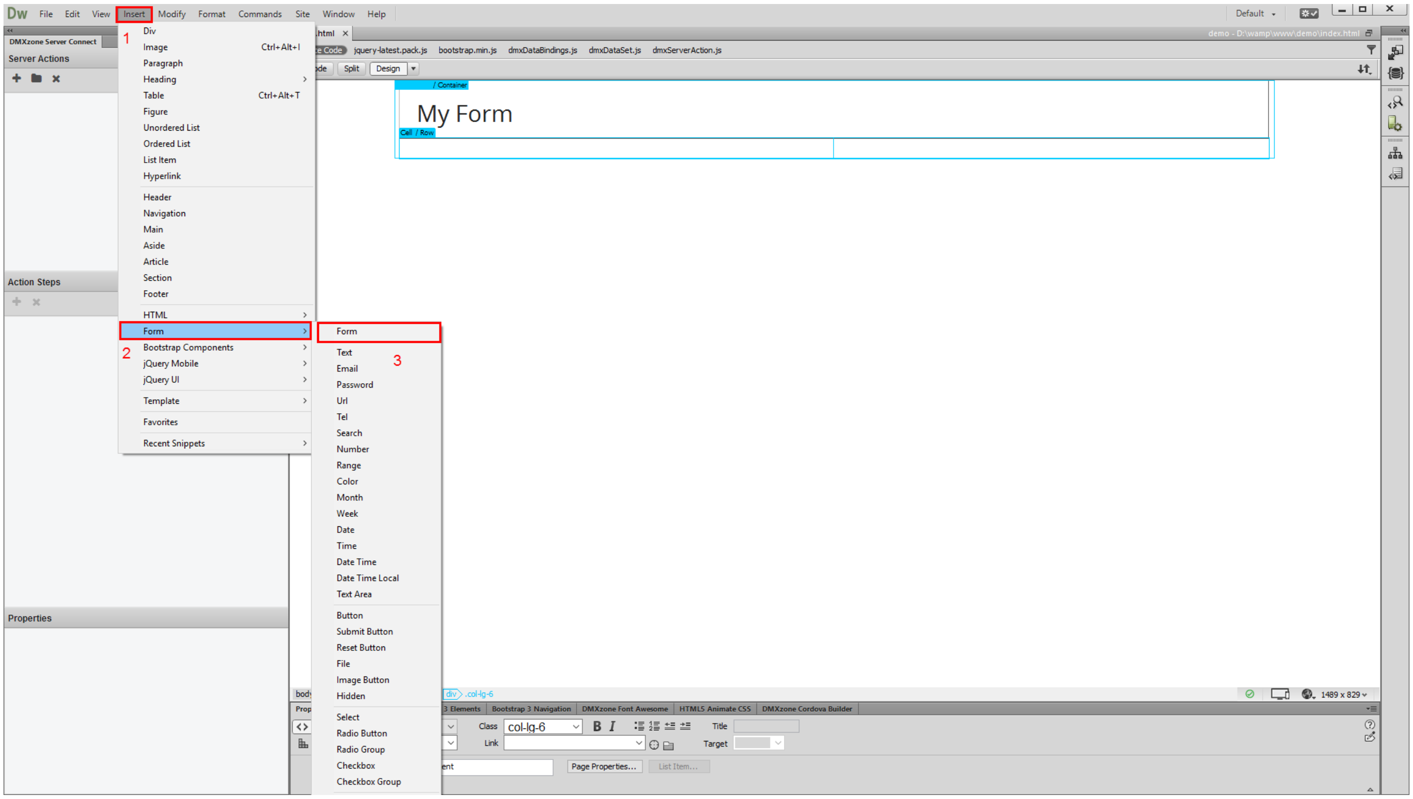
First Name:

Last Name:

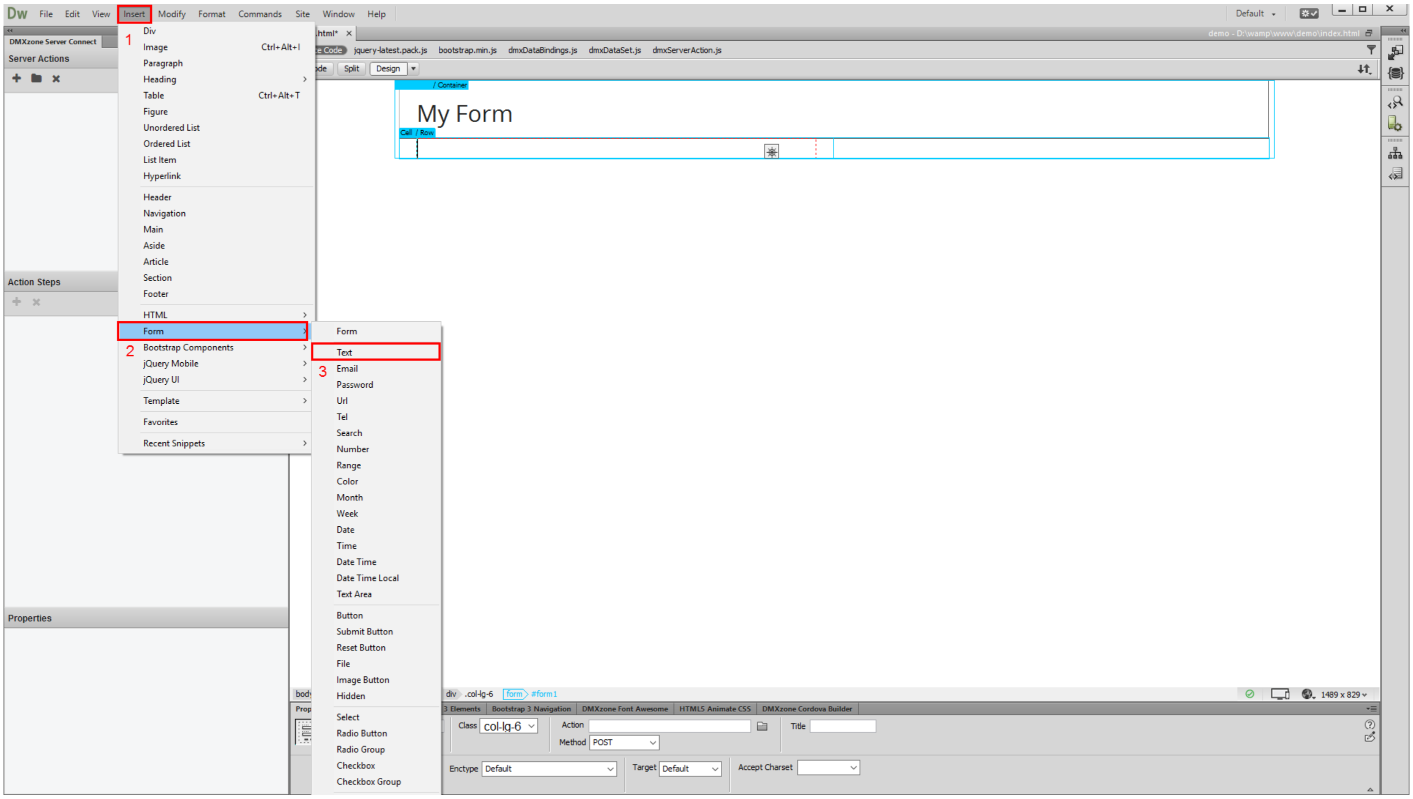
Location:

How to do it

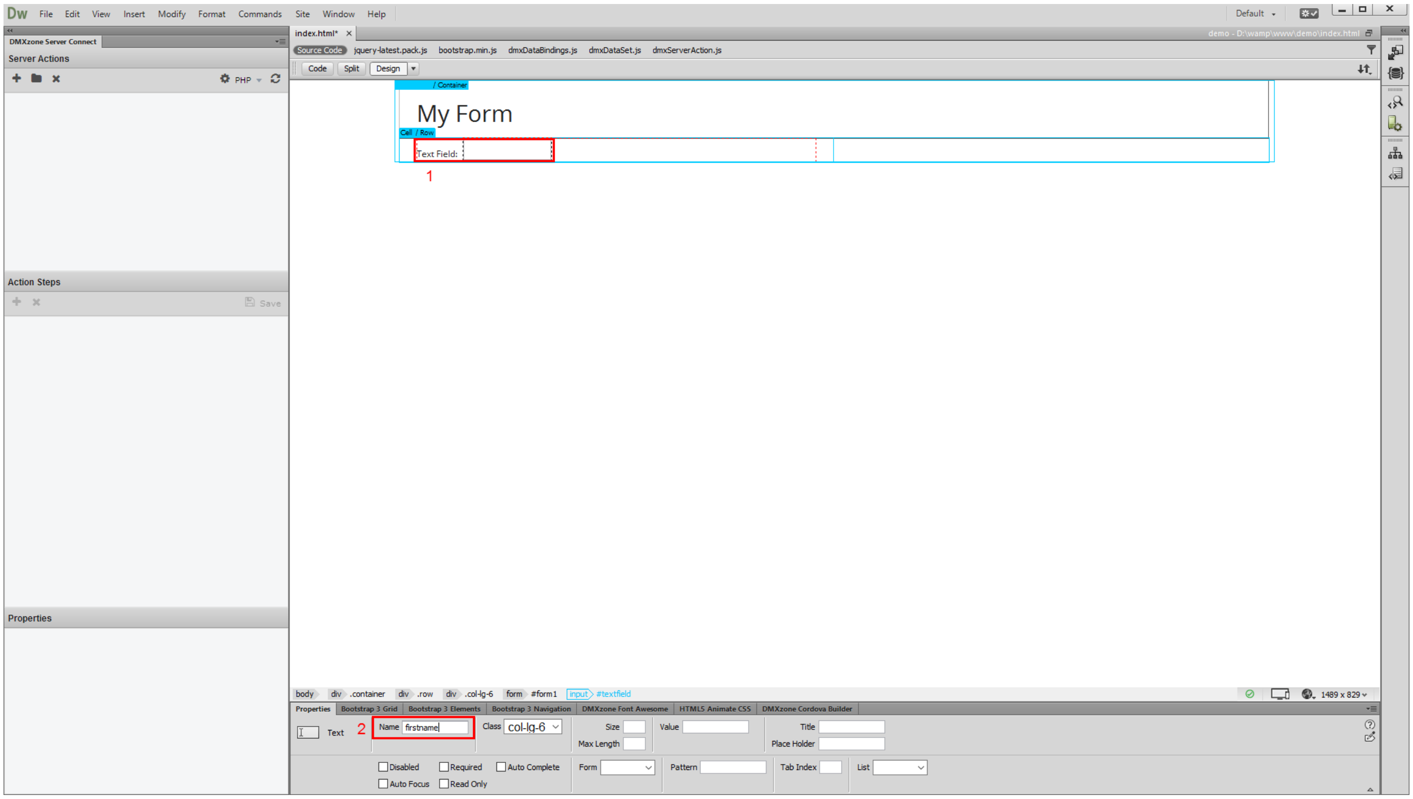
1. First, let's add a form, with several text fields on our page. In order to do that, open **the insert menu (1)**, go to **form submenu (2)** and choose **form (3)**.



2. Now, add the text fields inside it. Open the **insert menu (1)** and from the **form submenu (2)** choose **text (3).**

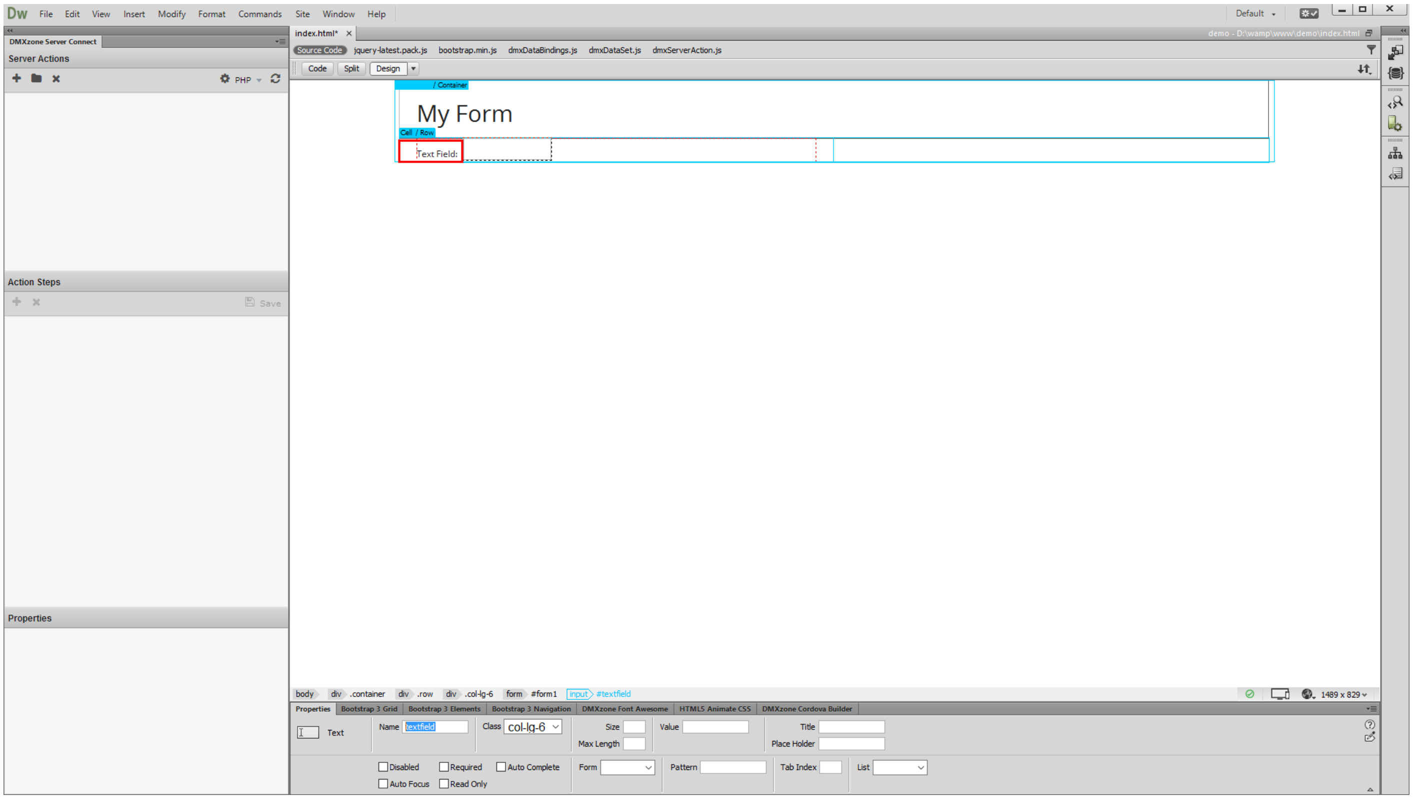


3. This adds a **text field** inside the form **(1)**. We change the default field name to **first name (2)**.

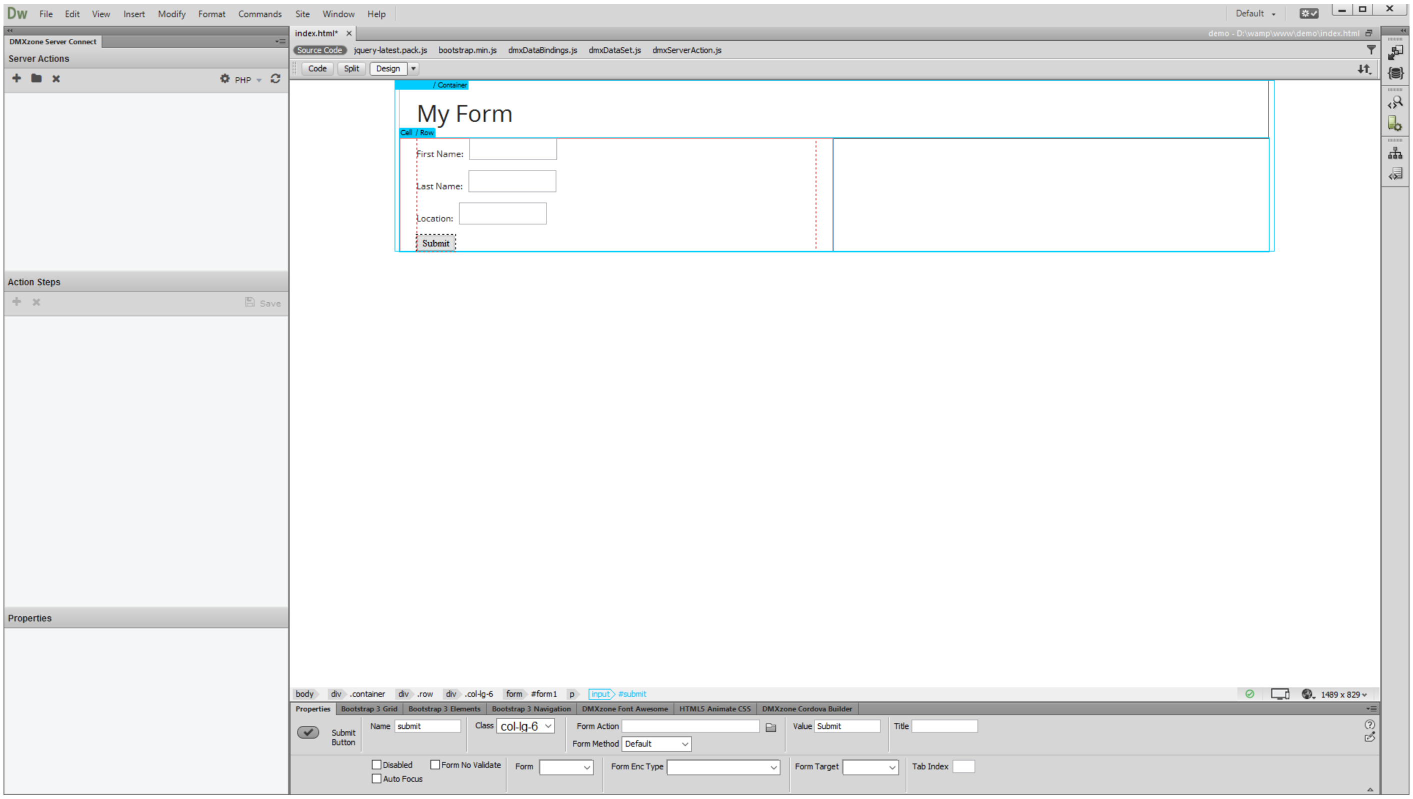

Dw File Edit View Insert Modify Format Commands Site Window Help
DMXzone Server Connect
Server Actions
PHP
Action Steps
Save
Properties
index.html*
Source Code jquery-latest.pack.js bootstrap.min.js dmxDataBindings.js dmxDataSet.js dmxServerAction.js
Code Split Design
Container
My Form
Cell / Row
Text Field:
1
body div .container div .row div .col-lg-6 form #form1 input #textfield
1489 x 829
Properties Bootstrap 3 Grid Bootstrap 3 Elements Bootstrap 3 Navigation DMXzone Font Awesome HTML5 Animate CSS DMXzone Cordova Builder
Text
2

Name firstname
Class col-lg-6
Size
Max Length
Value
Title
Place Holder
□ Disabled □ Required □ Auto Complete
□ Auto Focus □ Read Only
Form
Pattern
Tab Index
List

4. We also change the default label text to **first name (2)**. You can add as many text fields as you need.


Dw File Edit View Insert Modify Format Commands Site Window Help
Default
DMXzone Server Connect
Server Actions
PHP
Action Steps
Save
Properties
index.html*
Source Code jquery-latest.pack.js bootstrap.min.js dmxDataBindings.js dmxDataSet.js dmxServerAction.js
Code Split Design
Container
My Form
Cell / Row
Text Field:
body div .container div .row div .col-lg-6 form #form1 input #textfield
1489 x 829
Properties Bootstrap 3 Grid Bootstrap 3 Elements Bootstrap 3 Navigation DMXzone Font Awesome HTML5 Animate CSS DMXzone Cordova Builder
Text
Name textfield
Class col-lg-6
Size
Max Length
Value
Title
Place Holder
Disabled
Required
Auto Complete
Auto Focus
Read Only
Form
Pattern
Tab Index
List

5. For our tutorial, we added 3 different text fields and a submit button, as we need to submit the form.



My Form

First Name:

Last Name:

Location:

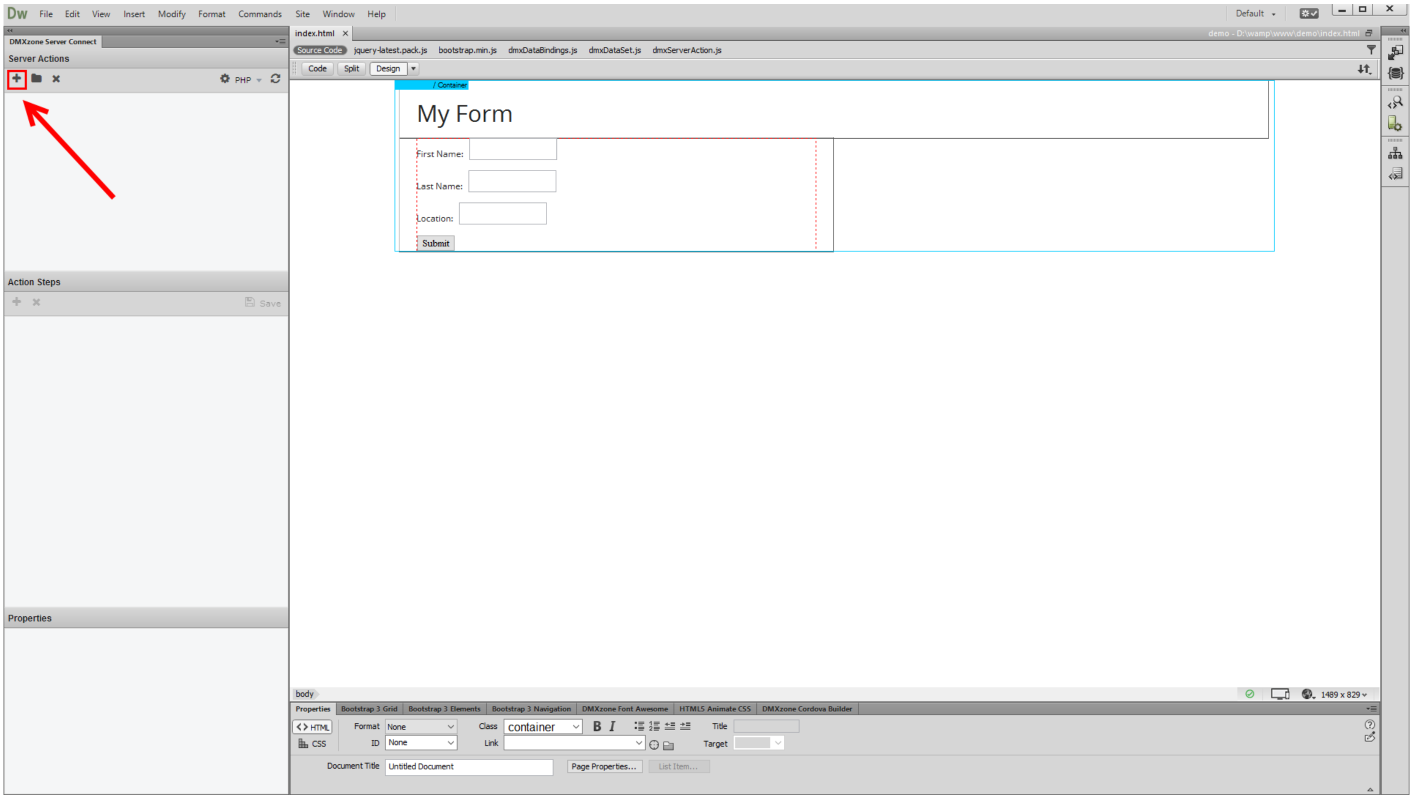
Submit

Name submit · Class col-lg-6 · Form Action · Form Method Default · Value Submit · Title

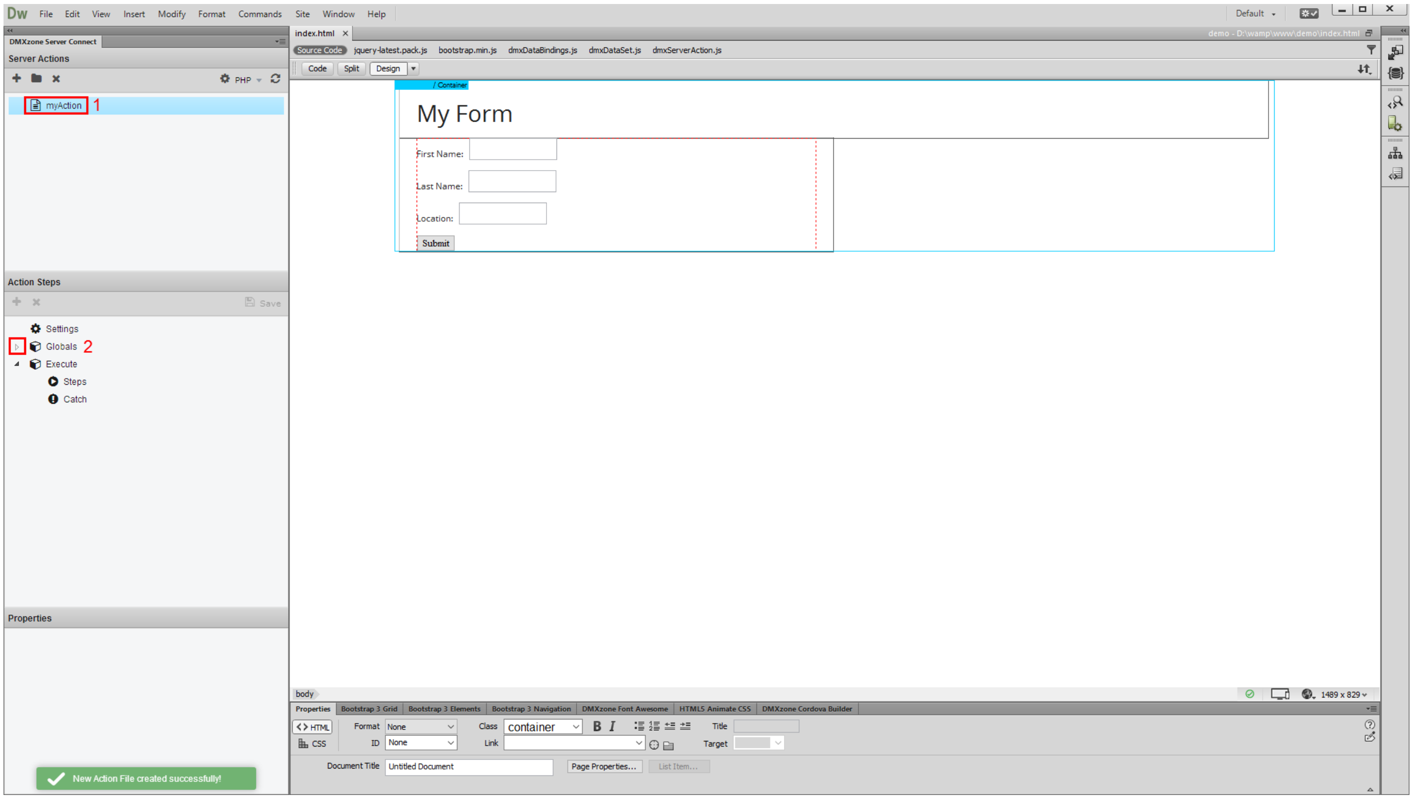
☐ Disabled ☐ Form No Validate ☐ Auto Focus · Form · Form Enc Type · Form Target · Tab Index

Dw File Edit View Insert Modify Format Commands Site Window Help
DMXzone Server Connect
Server Actions
PHP
Action Steps
Save
Properties
index.html*
Source Code jquery-latest.pack.js bootstrap.min.js dmxDataBindings.js dmxDataSet.js dmxServerAction.js
Code Split Design
Container
Cell / Row
body div .container div .row div .col-lg-6 form #form1 p input #submit
1489 x 829
Properties Bootstrap 3 Grid Bootstrap 3 Elements Bootstrap 3 Navigation DMXzone Font Awesome HTML5 Animate CSS DMXzone Cordova Builder
Submit Button

6. Now, let’s create our server action. Click **add new server action** button.

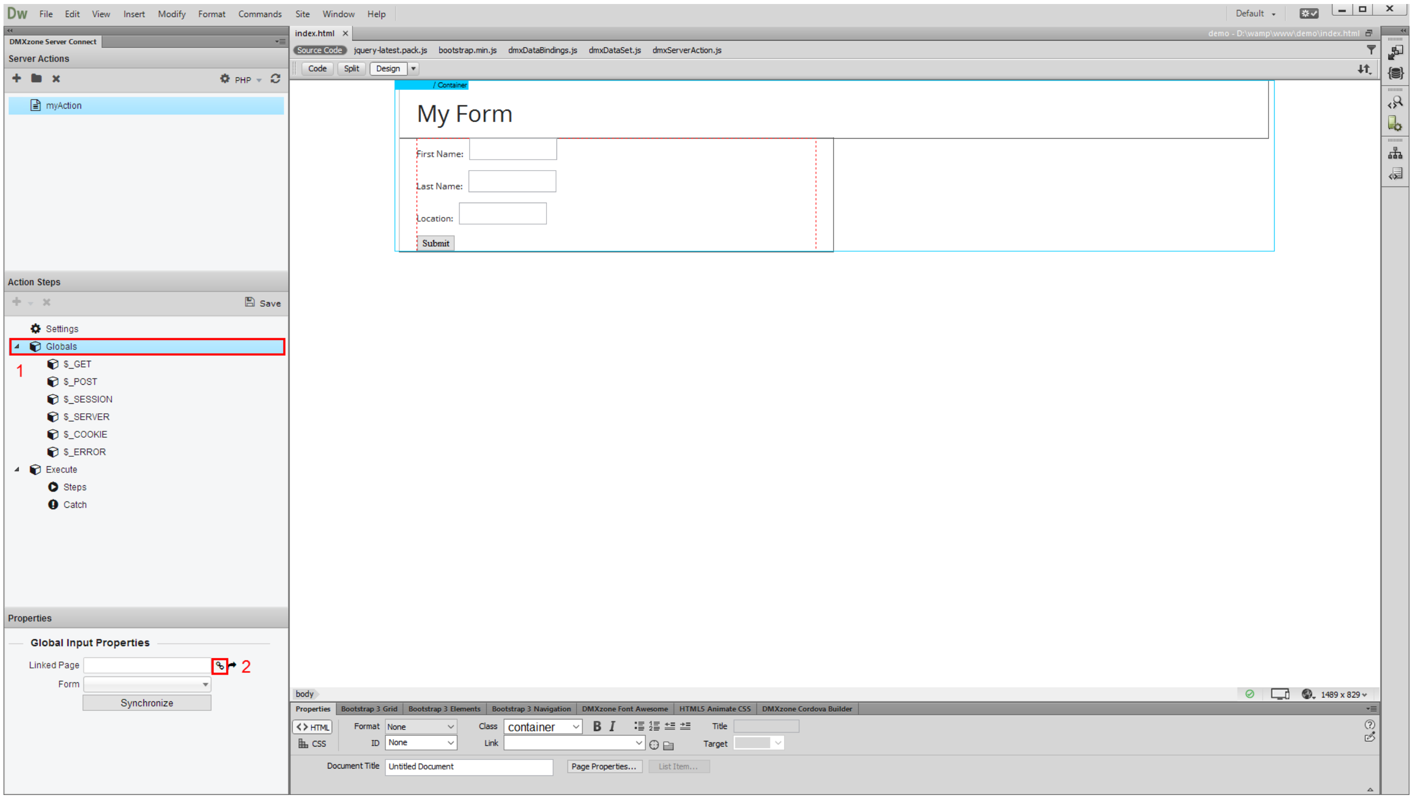

Dw File Edit View Insert Modify Format Commands Site Window Help
DMXzone Server Connect
Server Actions
PHP
Action Steps
Save
Properties
index.html
Source Code jquery-latest.pack.js bootstrap.min.js dmxDataBindings.js dmxDataSet.js dmxServerAction.js
Code Split Design
Container
My Form
First Name:
Last Name:
Location:
Submit
body
1489 x 829
Properties Bootstrap 3 Grid Bootstrap 3 Elements Bootstrap 3 Navigation DMXzone Font Awesome HTML5 Animate CSS DMXzone Cordova Builder
HTML
CSS
Format None
ID None
Class container
Link
Title
Target
Document Title Untitled Document
Page Properties...
List Item...

7. You can change the **name for the server action** but we leave it to its default **(1)**. Now under the available action steps expand the **global list (2)**.

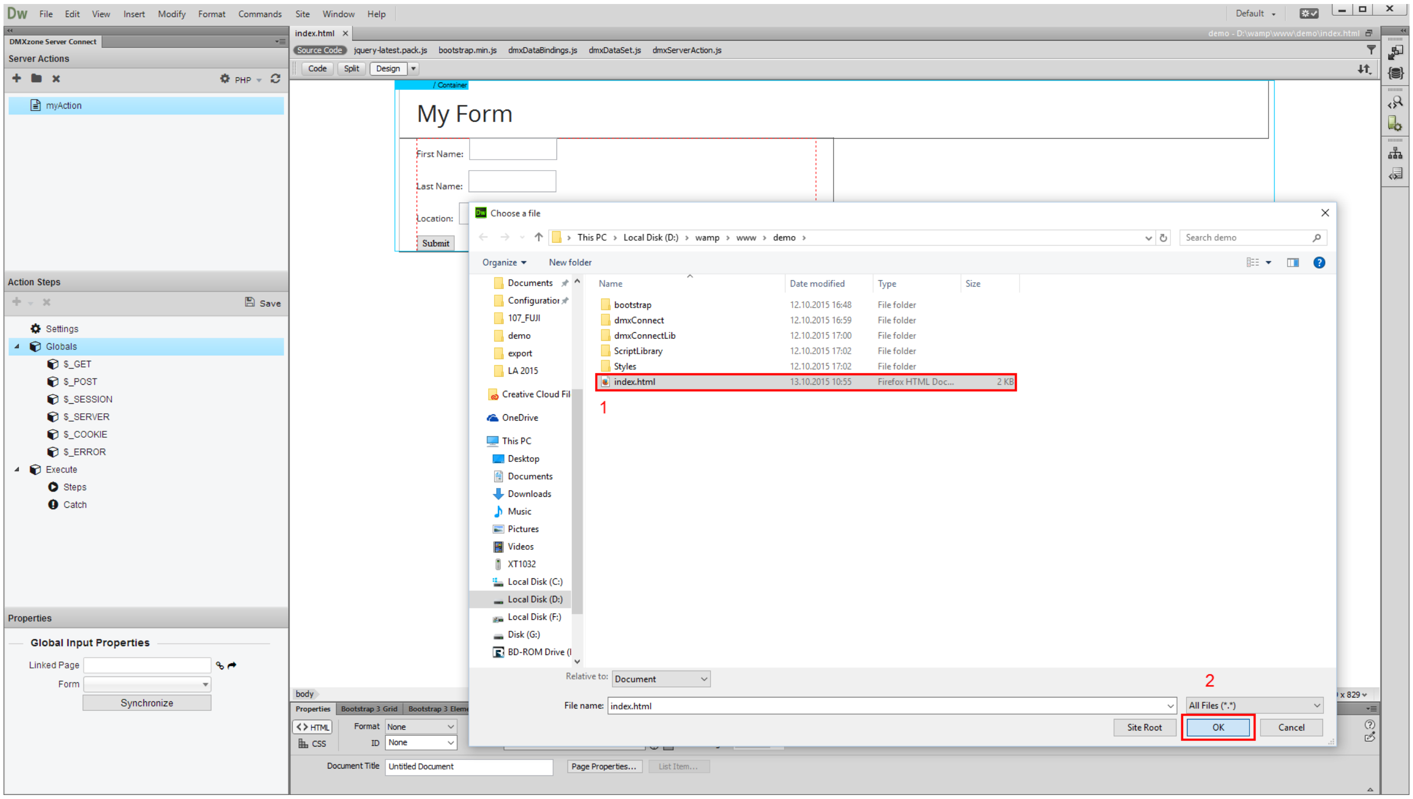


New Action File created successfully!

8. Here you can define your global form, session, server or cookie variables. In order to be able to use the form variables from the page, that we've just created, you need to define the input variables. You can either define the variables manually or bind to your page, containing the HTML form. Let's link our page to the global variables. First select globals (1) then click the **browse to page button (2)** located in the properties inspector.



9. Select the **page (1)**, which contains your HTML form and click the **OK button (2)**.



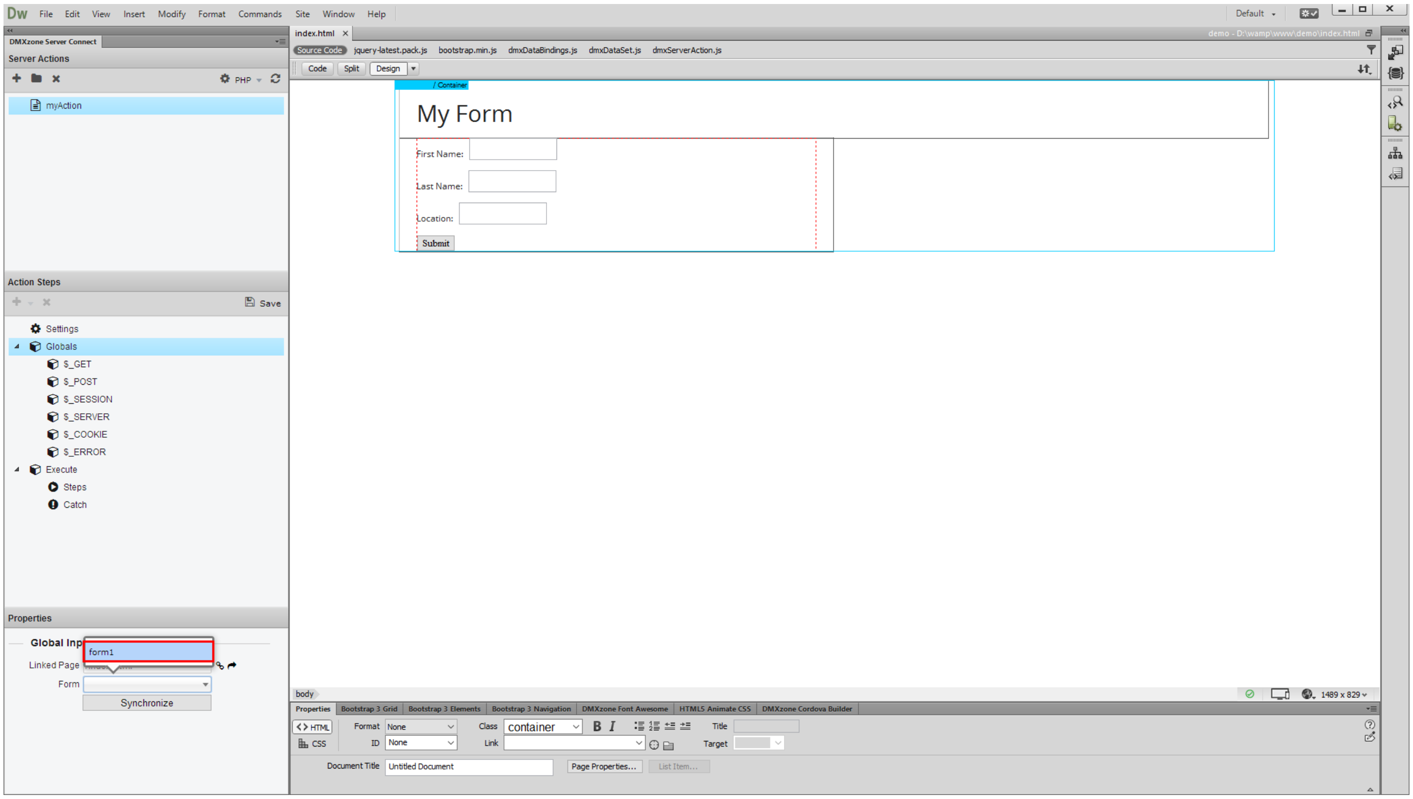
Name	Date modified	Type	Size
bootstrap	12.10.2015 16:48	File folder	
dmxConnect	12.10.2015 16:59	File folder	
dmxConnectLib	12.10.2015 17:00	File folder	
ScriptLibrary	12.10.2015 17:02	File folder	
Styles	12.10.2015 17:02	File folder	
index.html	13.10.2015 10:55	Firefox HTML Doc...	2 KB

Relative to: Document

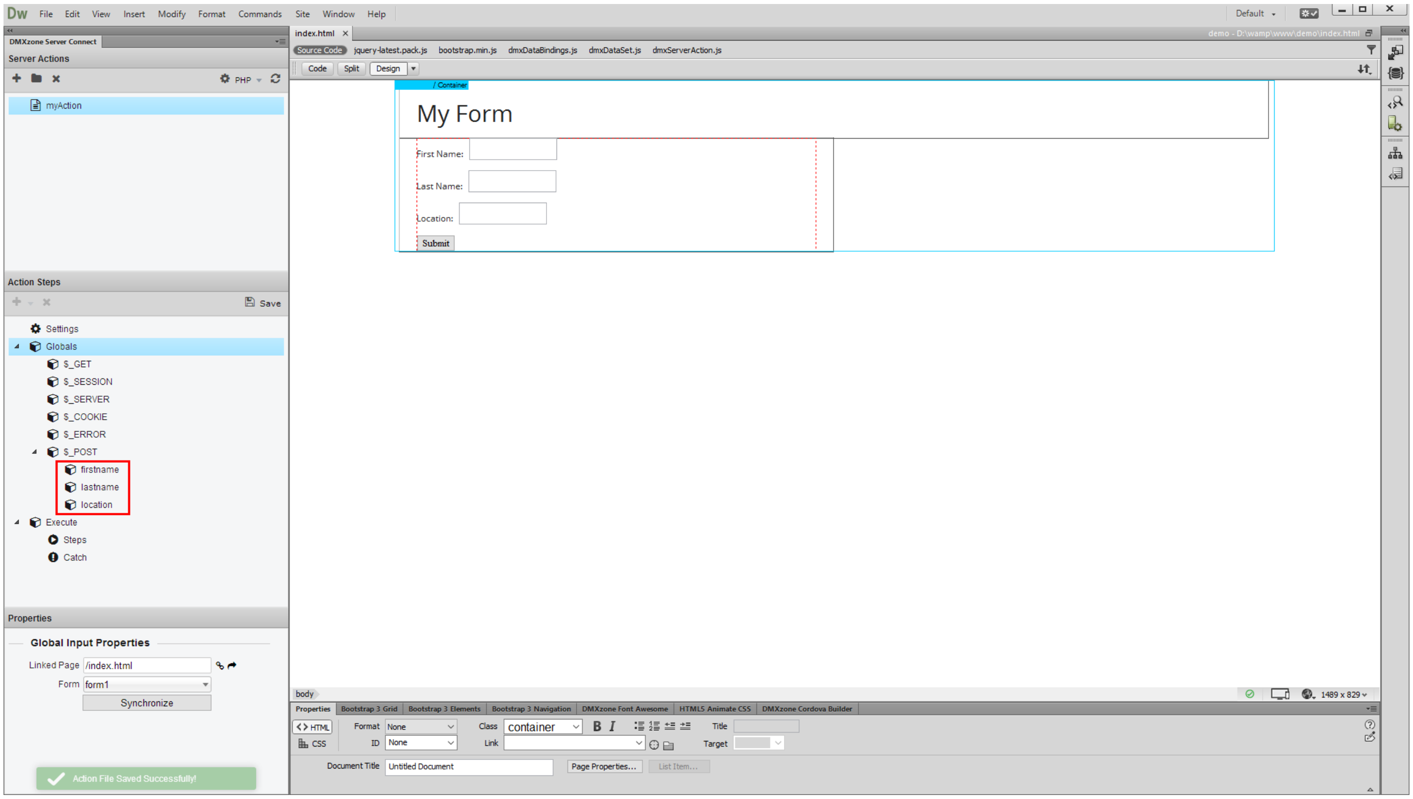
File name: index.html

All Files (*.*)

10. Once you've linked the page you can **select your HTML form** in the form dropdown. Just select it and the global input variables will be synced automatically.



11. Now you can see your form inputs under the global post variables. You can use them in your server action steps later.



myAction

My Form

First Name:

Last Name:

Location:

Submit

Settings
Globals
$_GET
$_SESSION
$_SERVER
$_COOKIE
$_ERROR
$_POST
firstname
lastname
location
Execute
Steps
Catch

Global Input Properties

Linked Page /index.html

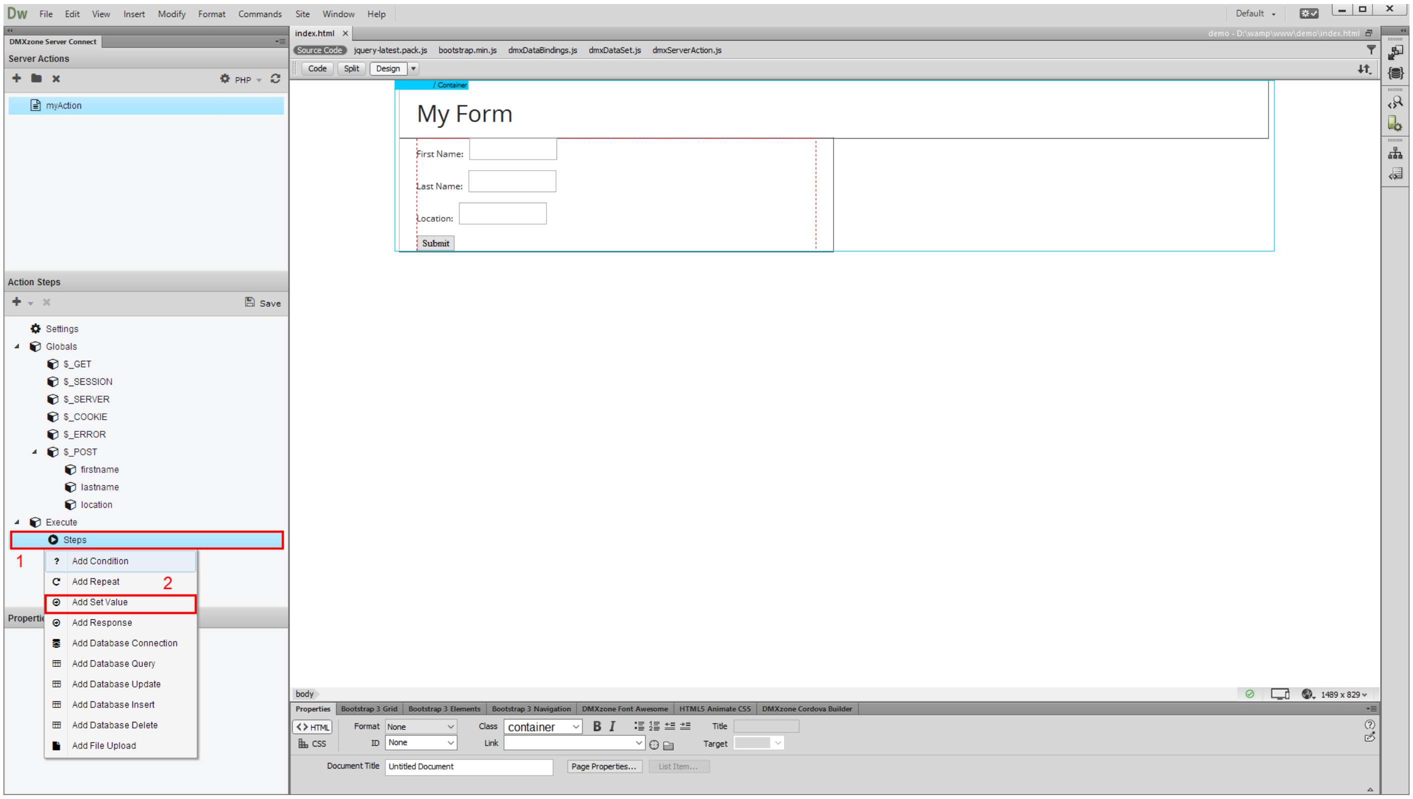
Form form1

Synchronize

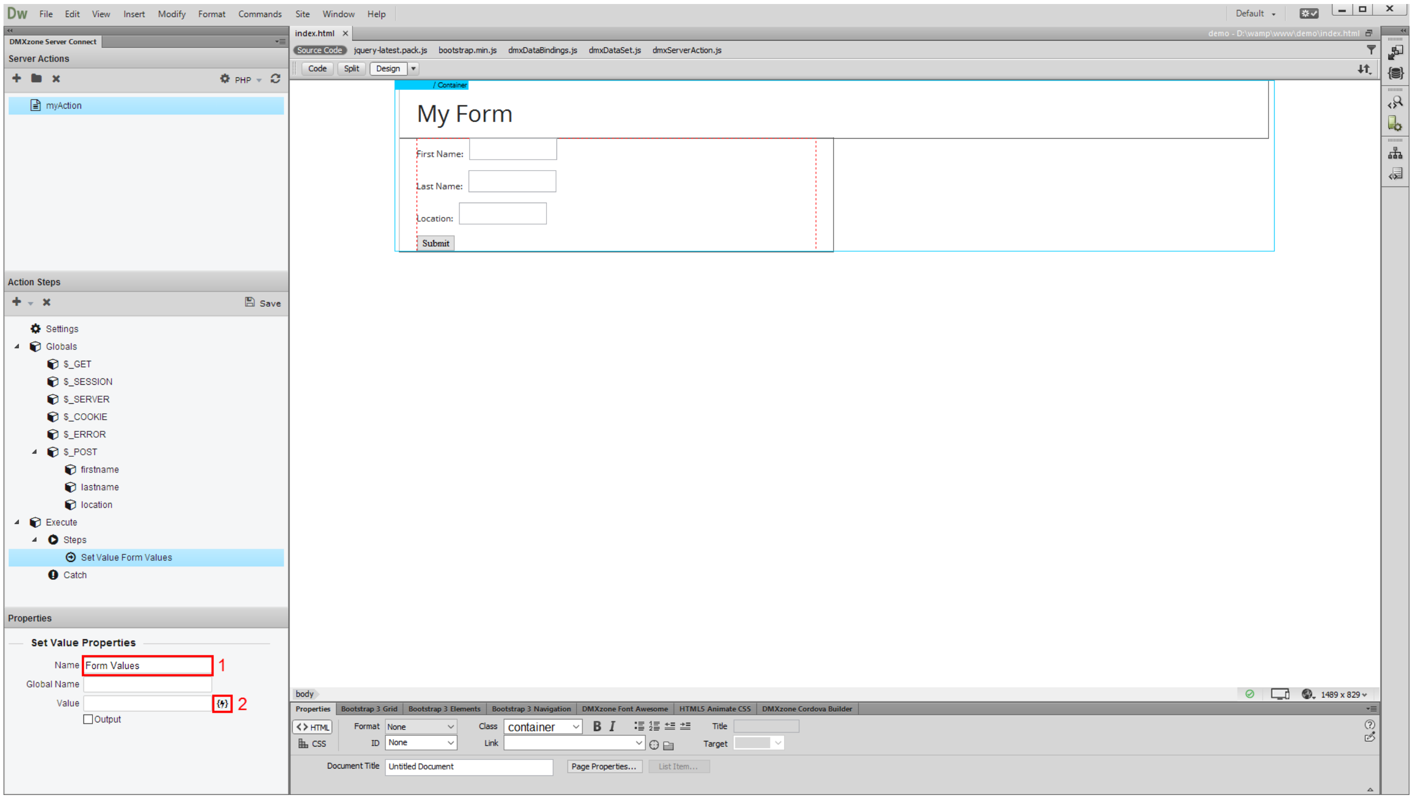
Action File Saved Successfully!

Format None
Class container
ID None
Link
Title
Target
Document Title Untitled Document

12. Let's create a really simple server action step, which will output the form inputs content, when we submit the form. Right click on **steps (1)** and select – **add value (2)**.



13. Add **a name** for the set value step **(1)** and select **the value** that you want to output **(2)**.

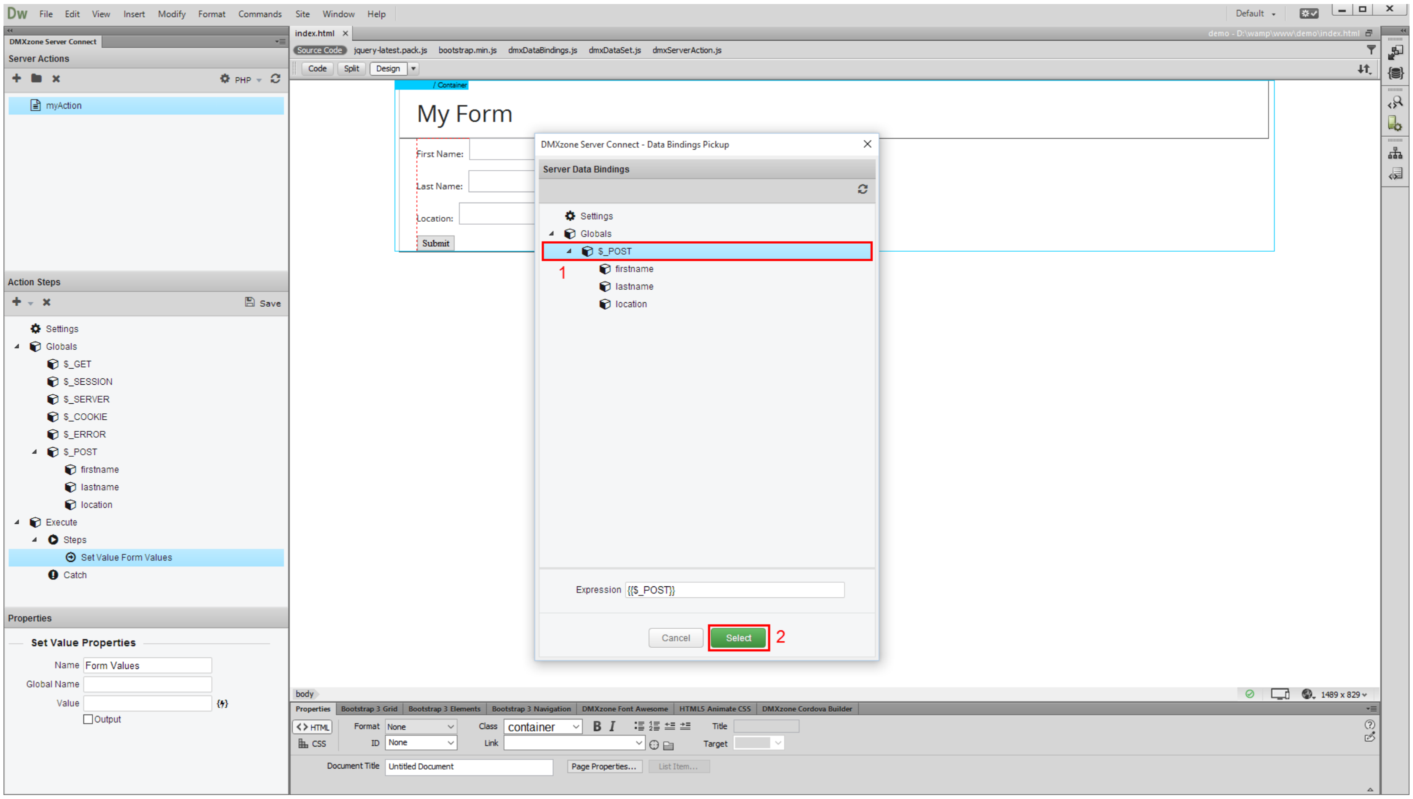


Set Value Properties

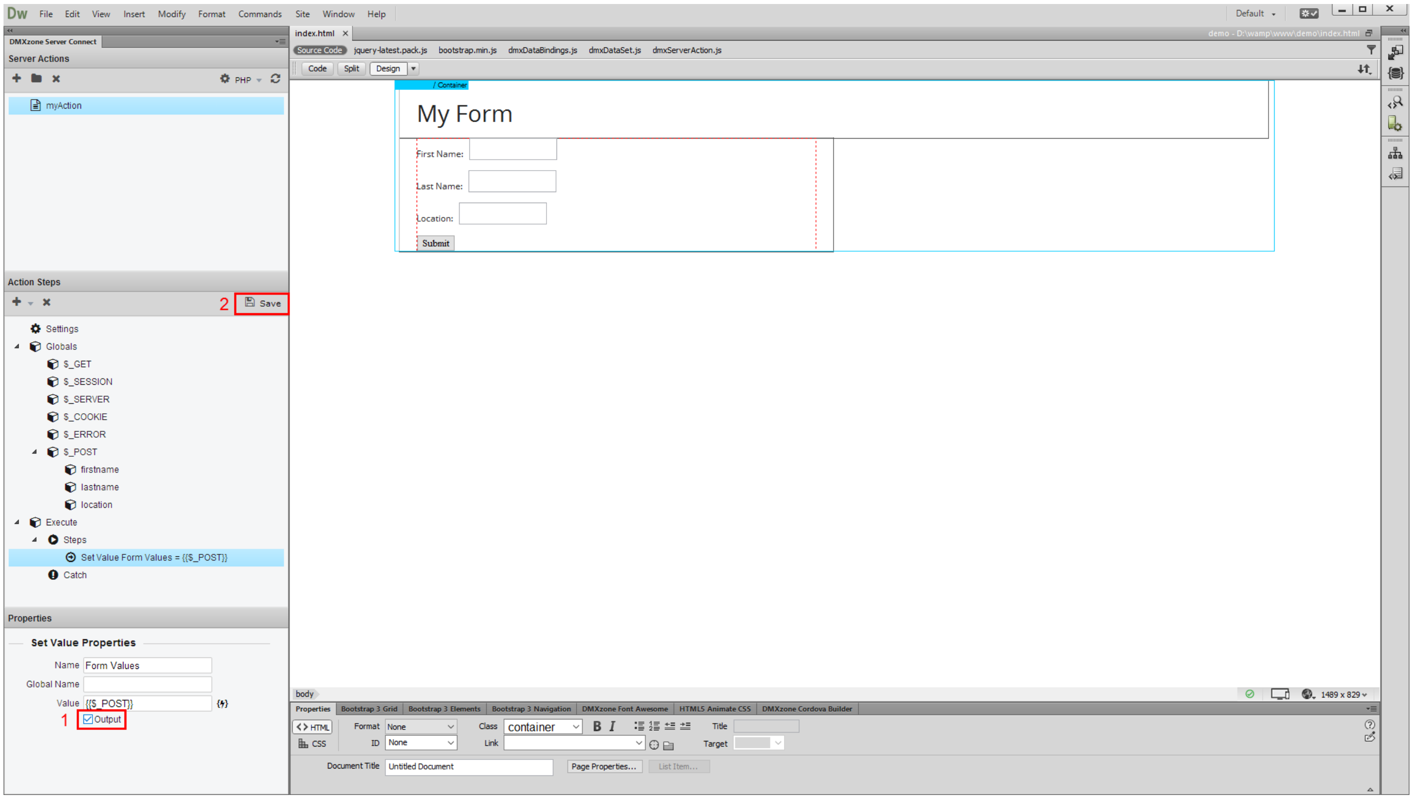
Name: Form Values
Global Name:
Value:
□ Output

Format: None
ID: None
Class: container
Document Title: Untitled Document

14. In the server data bindings panel select **the post global variable (1)**. This contains all of the form inputs. Click the **select button (2)**, when you're done.



15. Then **check the output option (1)**. It adds the data, generated by the step to the output of the DMXzone Server Connect Actions handler. Click the **save button (2)**.

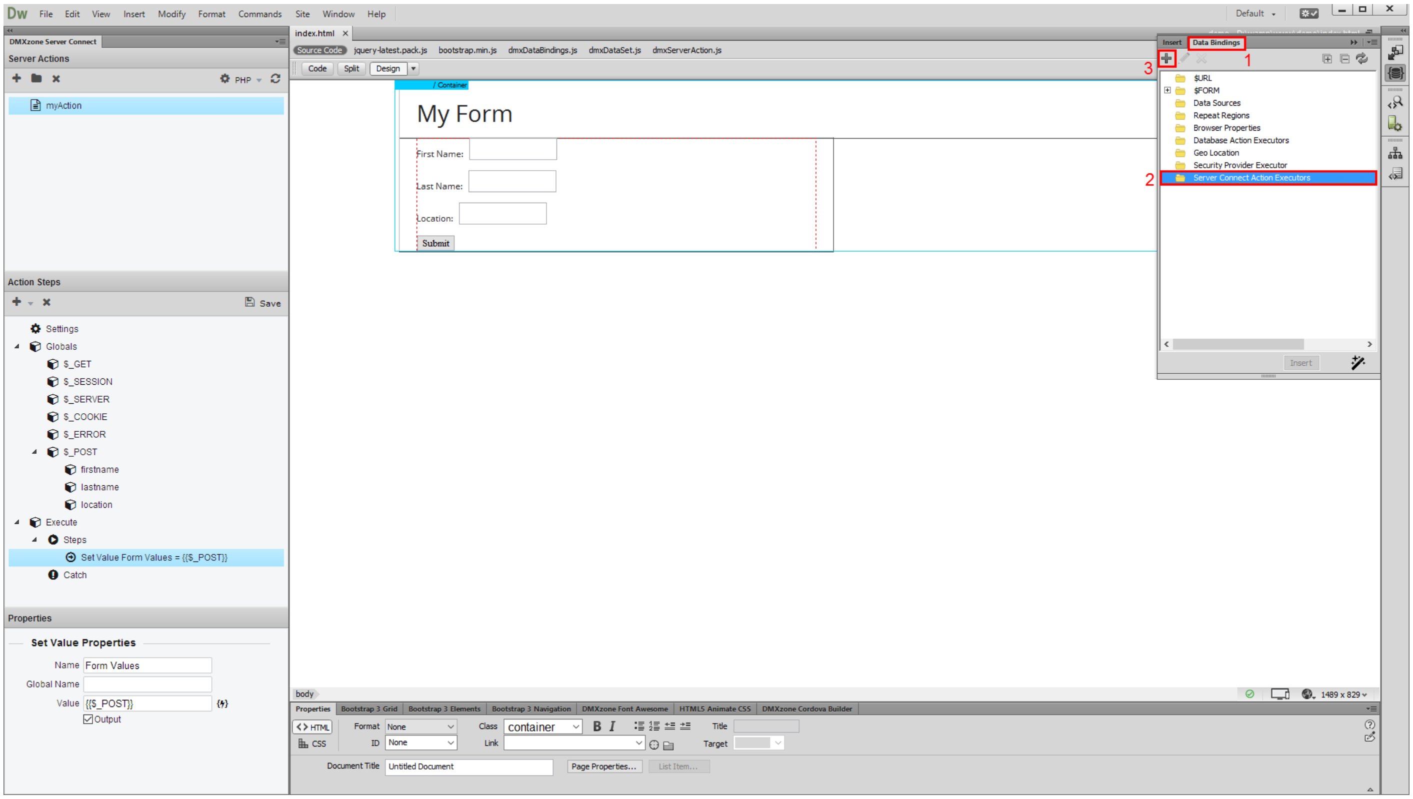


Dw File Edit View Insert Modify Format Commands Site Window Help
DMXzone Server Connect
Server Actions
PHP
myAction
index.html
Source Code jquery-latest.pack.js bootstrap.min.js dmxDataBindings.js dmxDataSet.js dmxServerAction.js
Code Split Design
My Form
First Name:
Last Name:
Location:
Submit
Action Steps
2
Save
Settings
Globals
$_GET
$_SESSION
$_SERVER
$_COOKIE
$_ERROR
$_POST
firstname
lastname
location
Execute
Steps
Set Value Form Values = {{$_POST}}
Catch
Properties
Set Value Properties
1
body
1489 x 829
Properties Bootstrap 3 Grid Bootstrap 3 Elements Bootstrap 3 Navigation DMXzone Font Awesome HTML5 Animate CSS DMXzone Cordova Builder
HTML CSS
Page Properties... List Item...

Name: Form Values
Global Name:
Value: {{$_POST}}
☑ Output

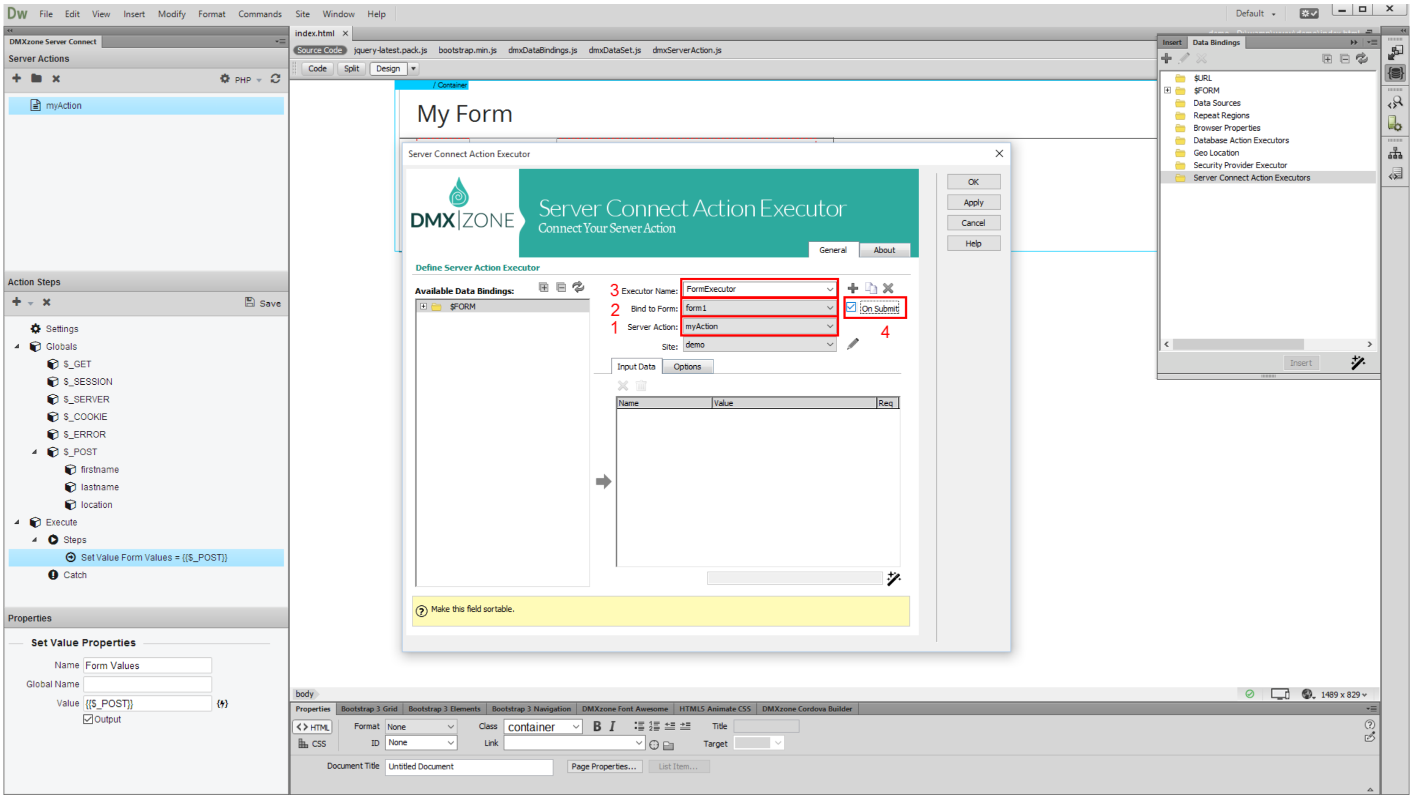
Format: None
ID: None
Class: container
Document Title: Untitled Document

16. When you are done defining the server action, you need to create a server action executor, so the action can be called on form submit. In order to do that, open the **HTML5 Data Bindings Panel (1),** select **server connect action executors (2)** and **click add new button (3)**.



Name: Form Values
Global Name:
Value: {{$_POST}}
☑ Output

17. Choose your **server action from** the dropdown **(1)**, select **the form**, you want to bind the executor to **(2)** and add **a name for this executor (3)**. Check the **on submit checkbox (4)**, in order to run the executor on form submit. And click the OK button, when you are done.



Server Connect Action Executor

Define Server Action Executor

3 Executor Name: FormExecutor

2 Bind to Form: form1

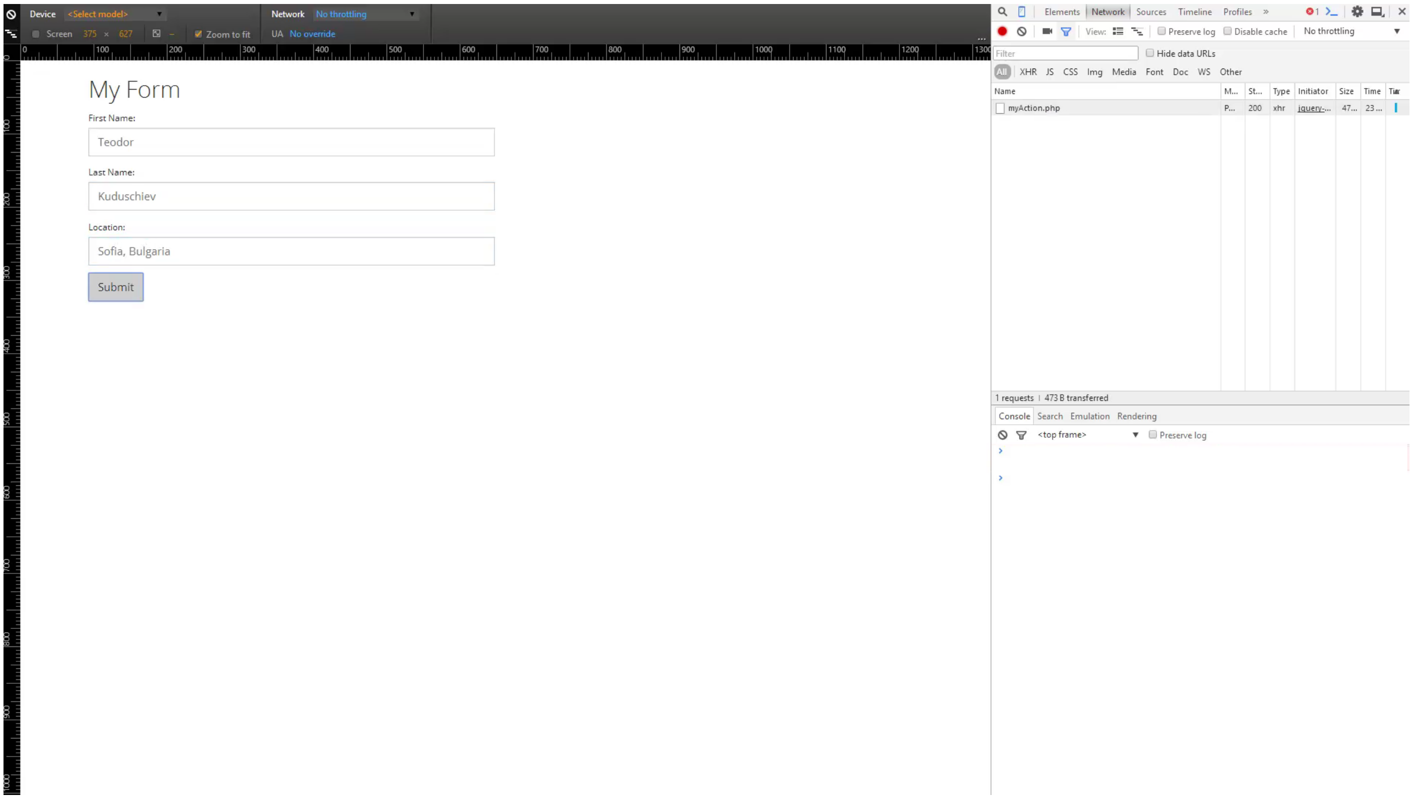
1 Server Action: myAction

Site: demo

4 ☑ On Submit

Make this field sortable.

18. Now, let’s preview the results in our browser. Open the page in the Chrome Browser. Press f12 in order to activate the developer tools. Here, we can see the output results, from our server action, as soon as the form is submitted. You can see the output from our HTML Form inputs in JSON format. That’s how easy it is to add your form fields to the global input variables of DMXzone Server Connect.



My Form

First Name: Teodor

Last Name: Kuduschiev

Location: Sofia, Bulgaria

Submit

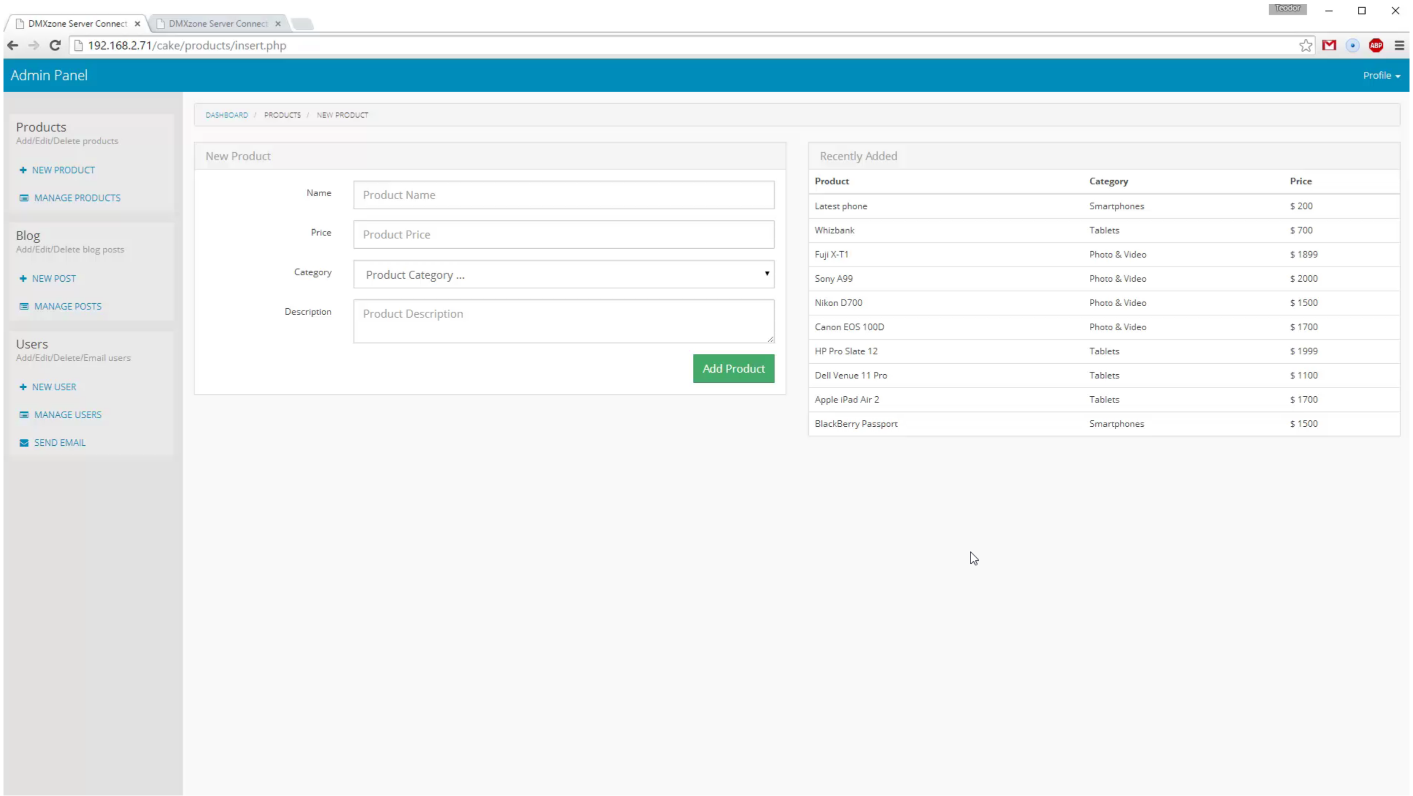
Name	M...	St...	Type	Initiator	Size	Time
myAction.php	P...	200	xhr	jquery-...	47...	23 ...

1 requests | 473 B transferred

Advanced: Insert Records into a Database

In this tutorial we will show you how to how to insert records in your database, using DMXzone Server Connect. Note, that the insert record functionality requires the HTML5 Data Bindings, DMXzone Database Connector PHP/ASP and DMXzone Database Updater PHP/ASP extensions.

*You can also check out the video.



New Product

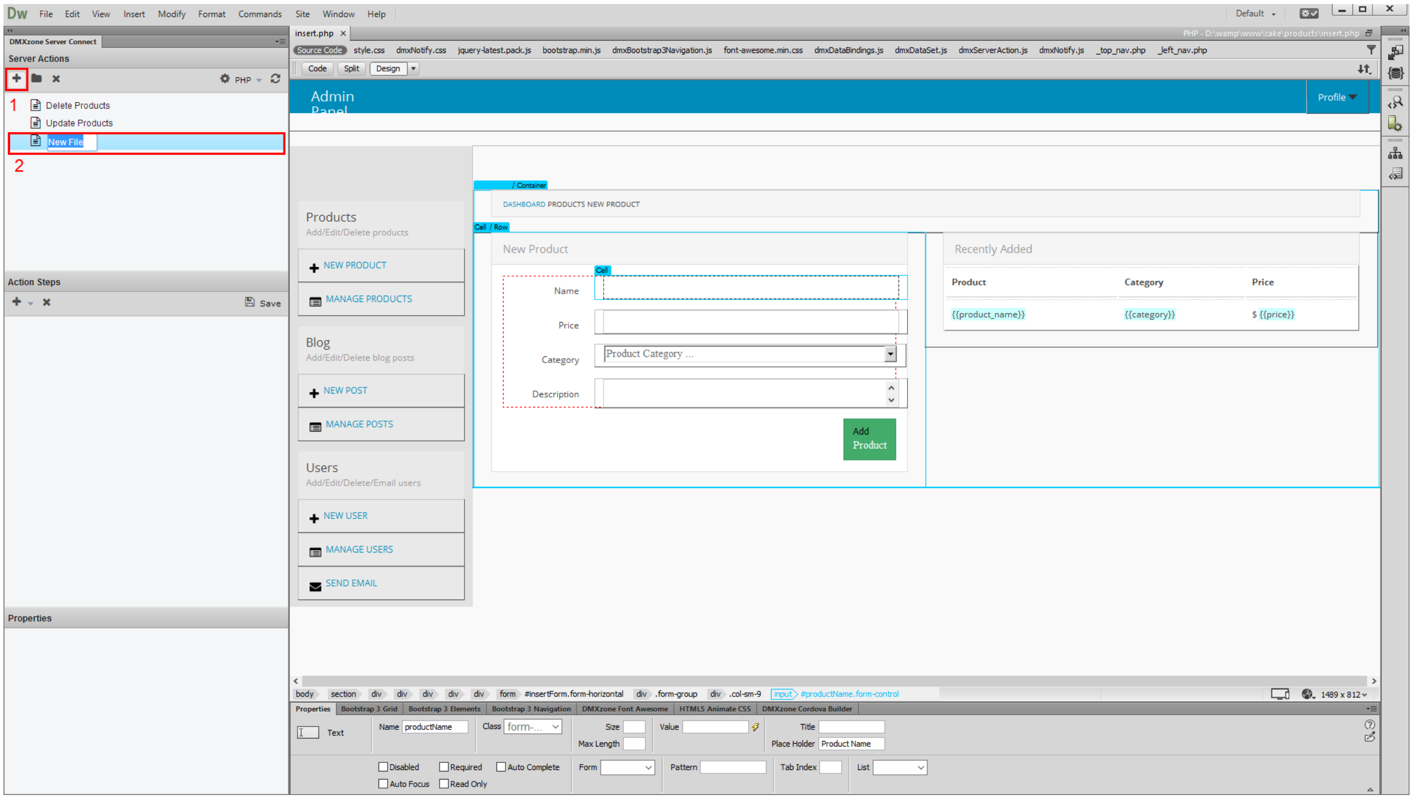
Name	Product Name
Price	Product Price
Category	Product Category ...
Description	Product Description

Recently Added

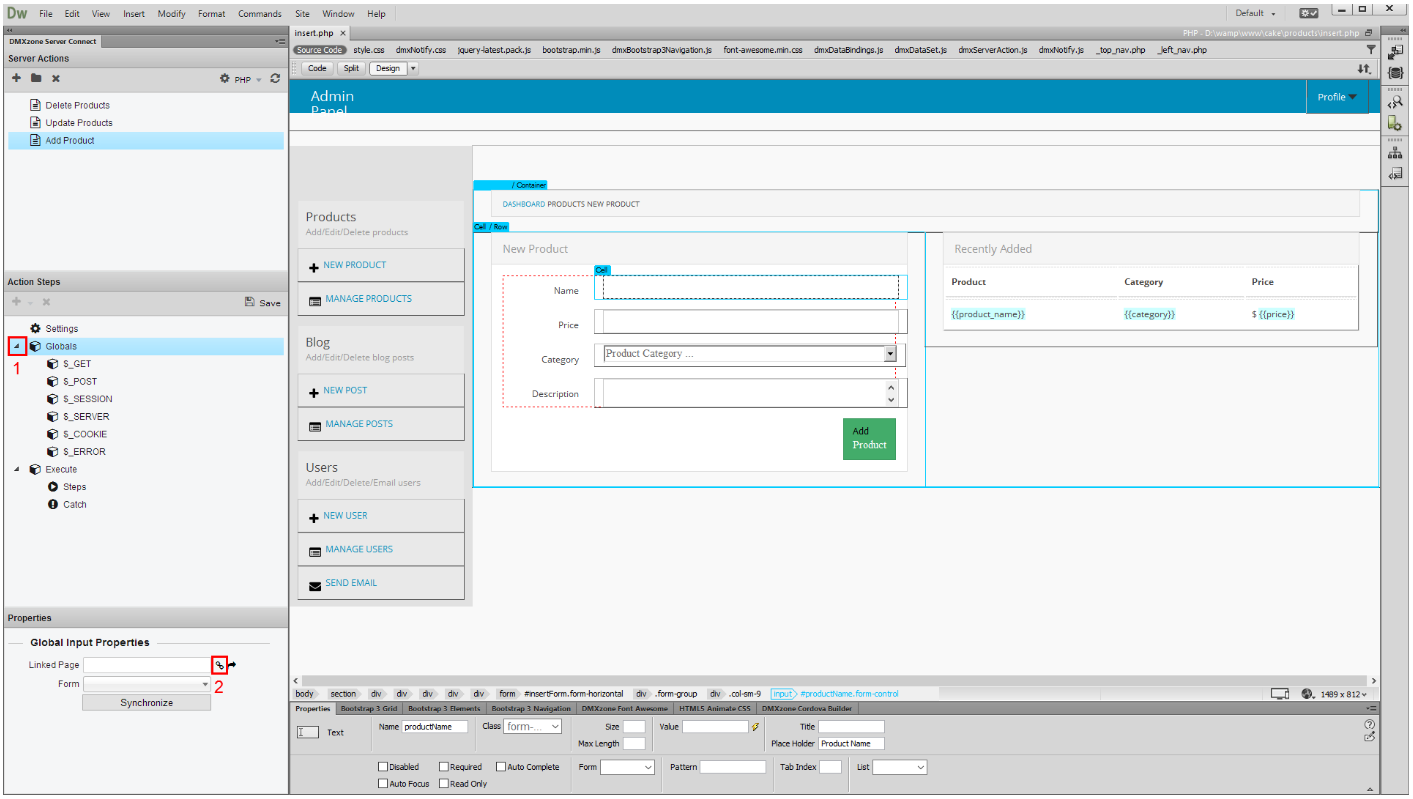
Product	Category	Price
Latest phone	Smartphones	$ 200
Whizbank	Tablets	$ 700
Fuji X-T1	Photo & Video	$ 1899
Sony A99	Photo & Video	$ 2000
Nikon D700	Photo & Video	$ 1500
Canon EOS 100D	Photo & Video	$ 1700
HP Pro Slate 12	Tablets	$ 1999
Dell Venue 11 Pro	Tablets	$ 1100
Apple iPad Air 2	Tablets	$ 1700
BlackBerry Passport	Smartphones	$ 1500

How to do it

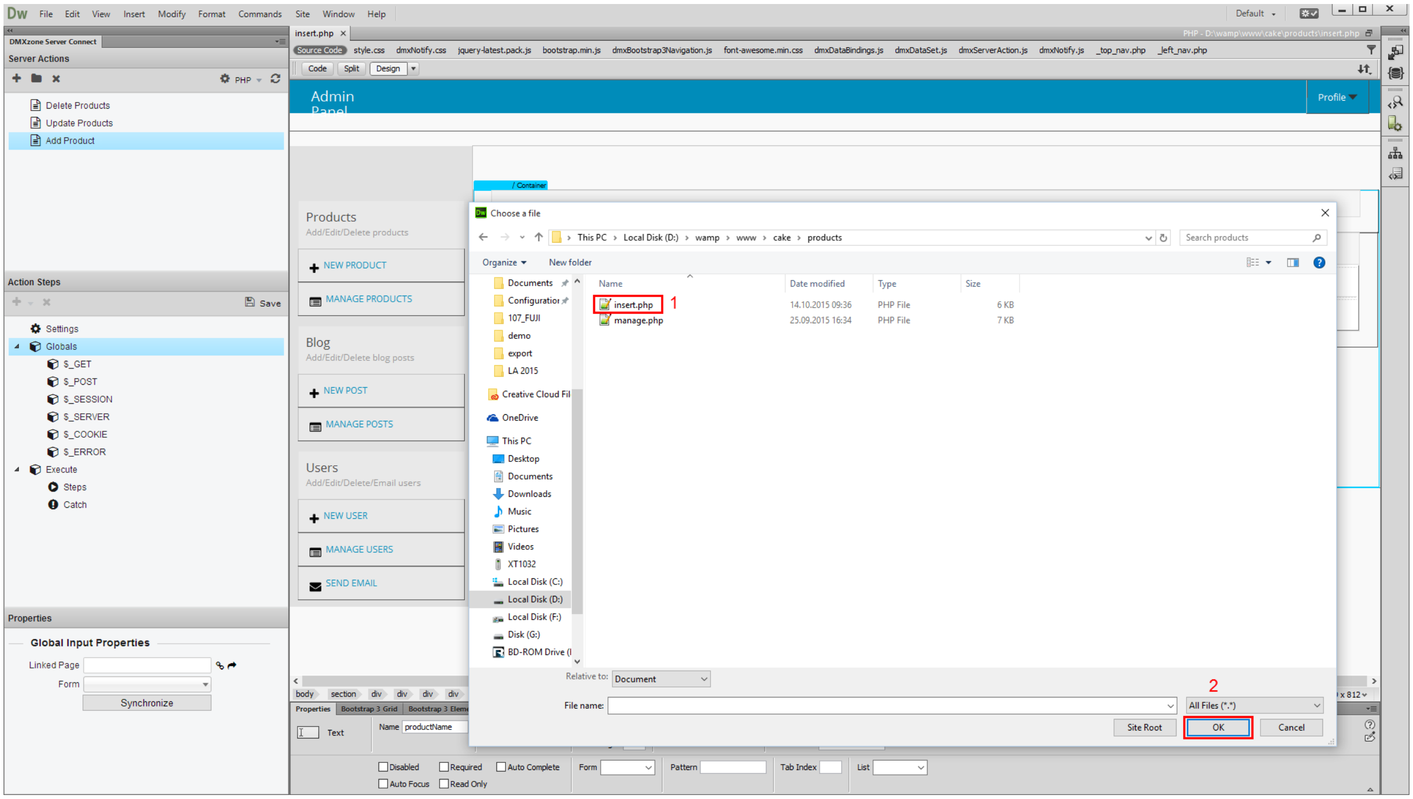
1. We created a simple HTML form, containing 4 fields and a submit button on our page. We are going to use it in order to insert information about products in our database. Click **add new server action button (1)** and **add a name** for your action **(2).**



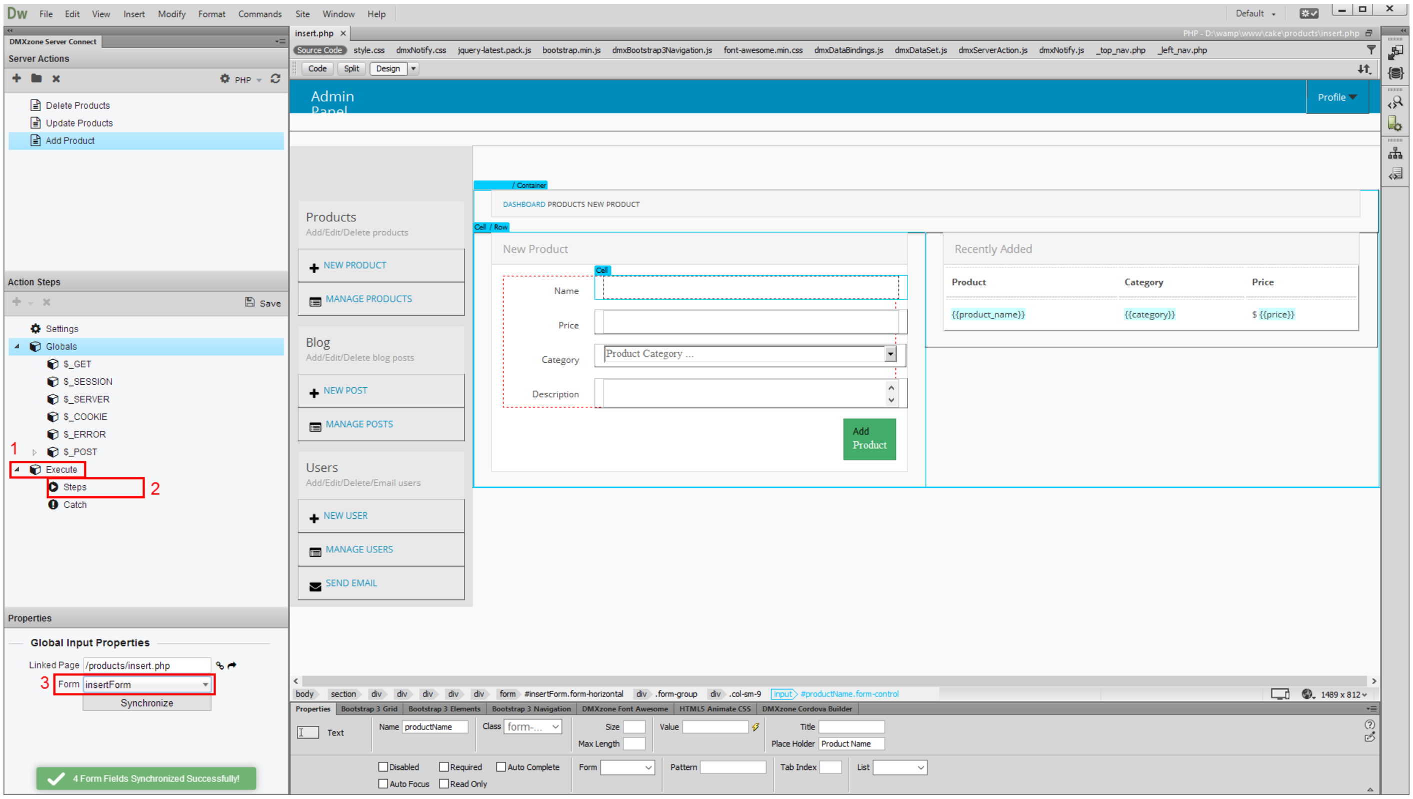
2. Select the **server action** that you just created **(1)**. Now, let's bind our form inputs to the global variables for this action. Expand globals **(2)** and in the property inspector **browse to the page**, containing your form (3).



3. Select the **page (1)** that contains your form and **click OK (2)**.



4. Select your **form** from the dropdown **(1)** and in the **actions steps list (2)**, under execute, **right click step (3).**

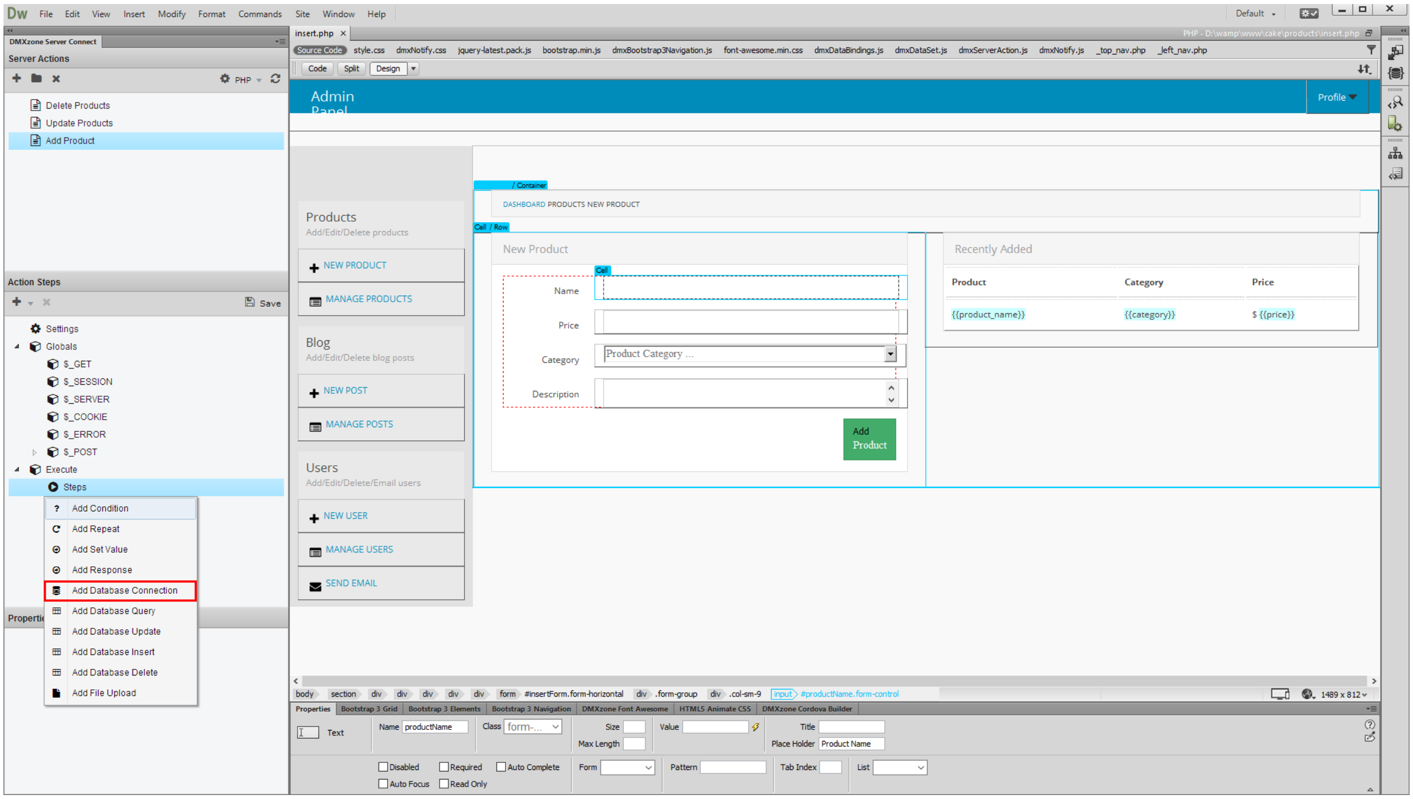


Linked Page /products/insert.php

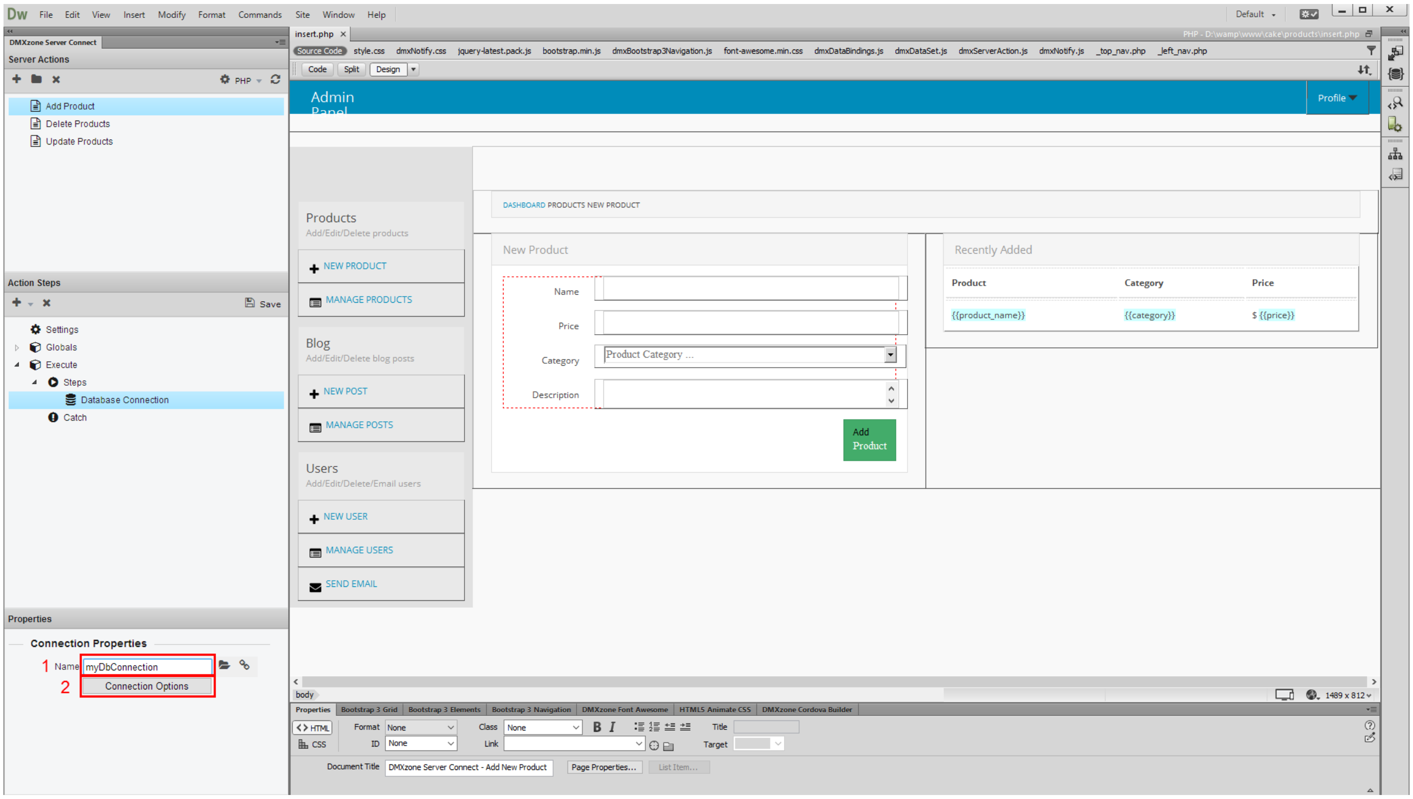
Form insertForm

4 Form Fields Synchronized Successfully!

5. Select **add database connection** from the menu.



6. Add **a name for your connection (1)** or if you have already defined one - just select it from the dropdown. In our tutorial we will show you how to setup one. Now click the **connection options button (2)**.

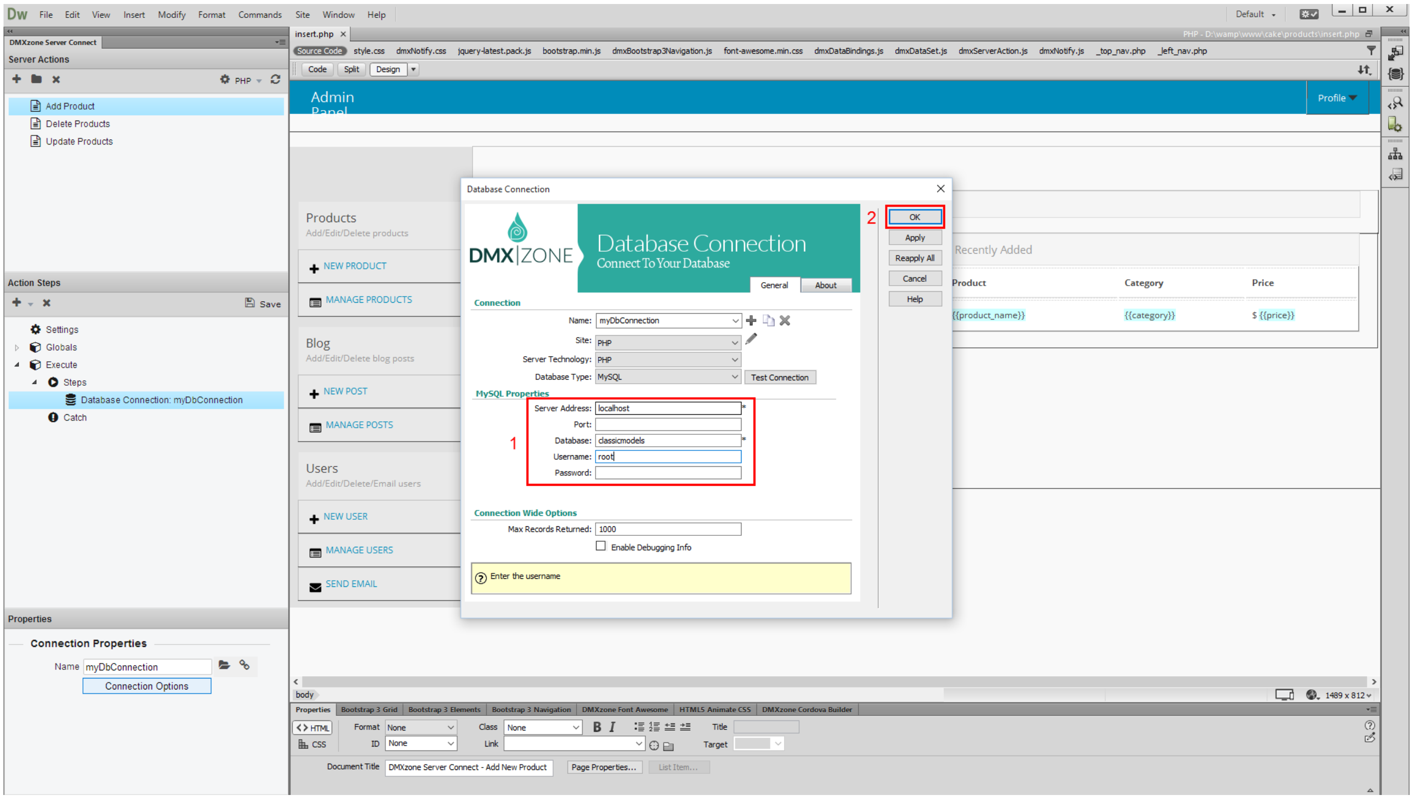


1 Name myDbConnection

2 Connection Options

Document Title: DMXzone Server Connect - Add New Product

7. Here, you need to enter your **MySQL server address, database, username and password (1). Click OK (2)** when you're done.



Database Connection

Database Connection
Connect To Your Database

Connection

Name: myDbConnection
Site: PHP
Server Technology: PHP
Database Type: MySQL

MySQL Properties

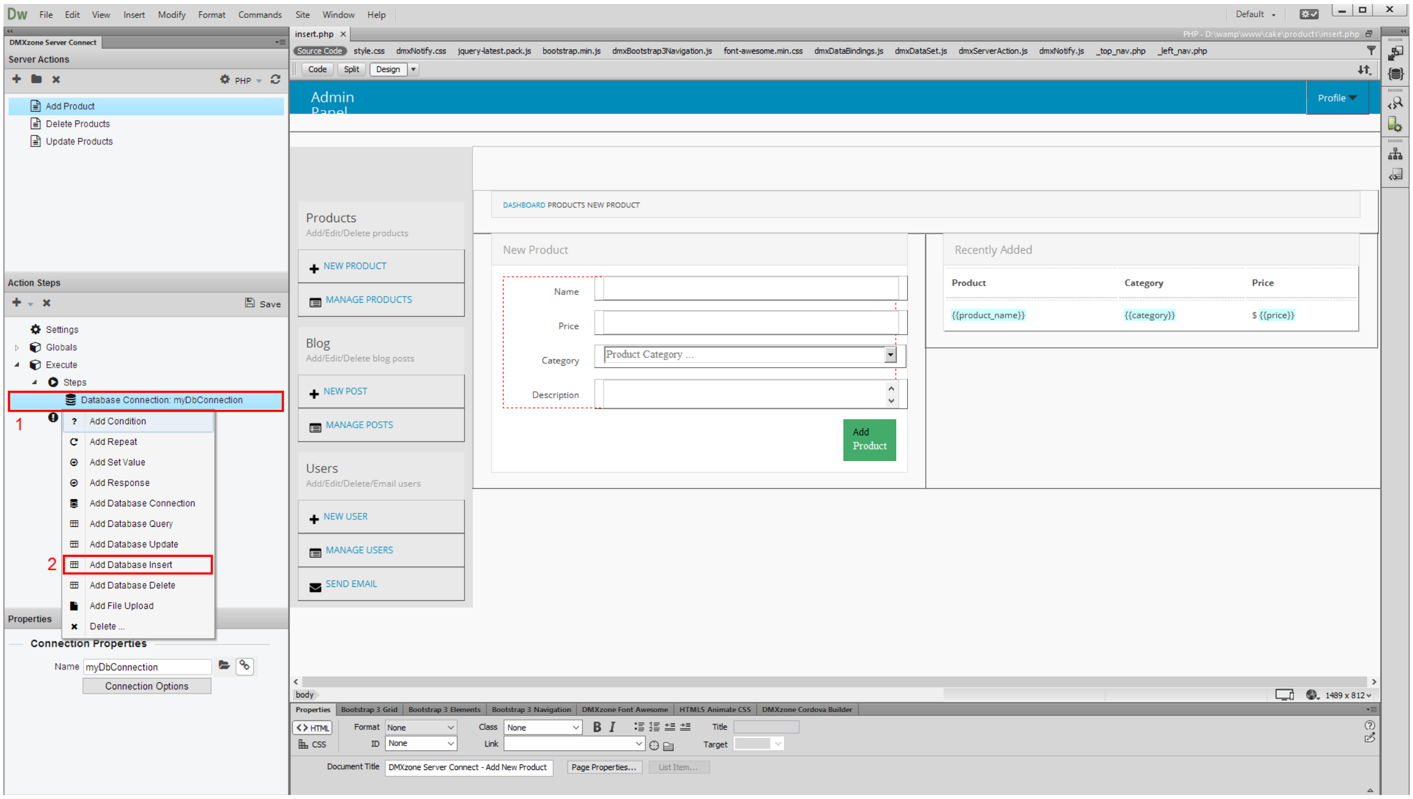
Server Address: localhost *
Port:
Database: classicmodels *
Username: root
Password:

Connection Wide Options

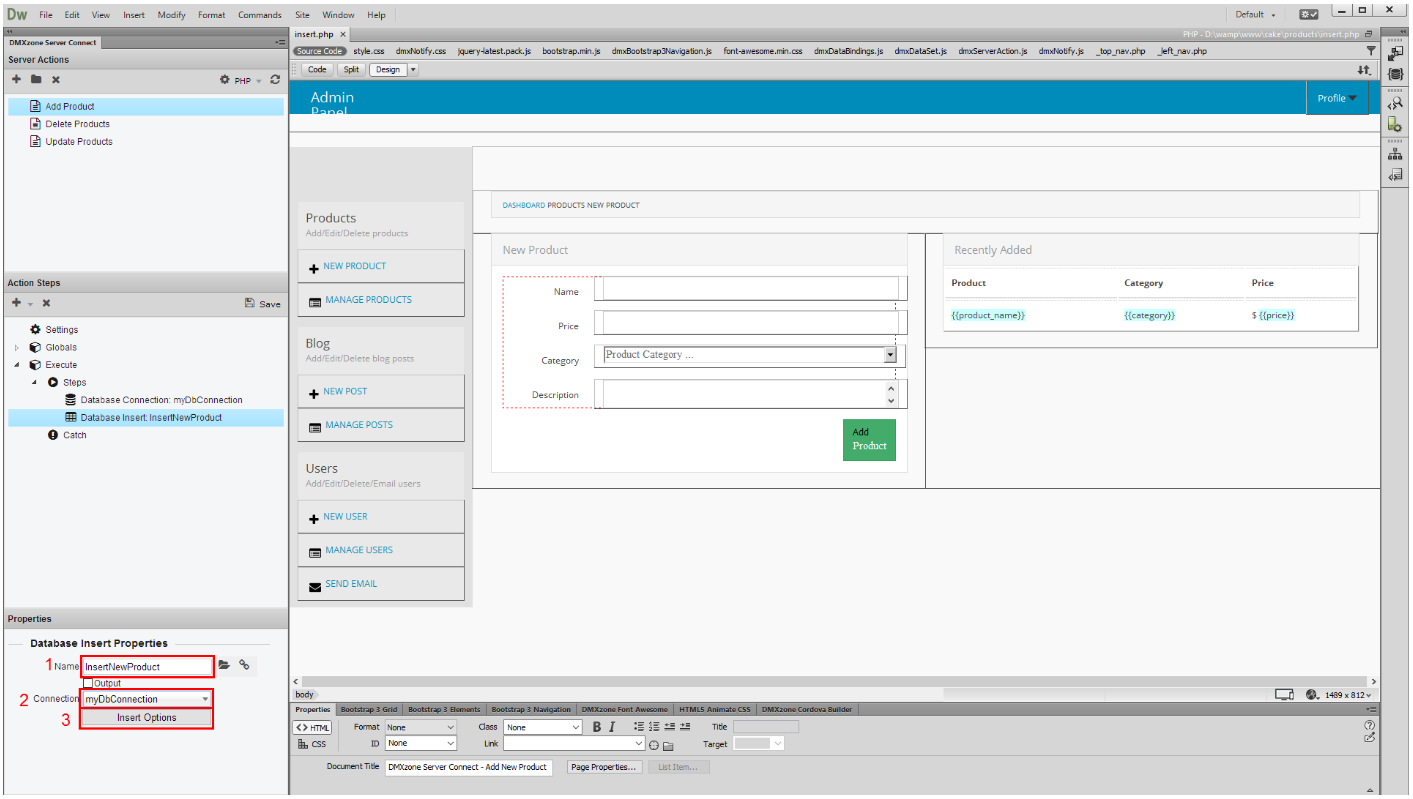
Max Records Returned: 1000
□ Enable Debugging Info

Enter the username

8. Now, we need to add the next step in the server action steps list. **Right click the database connection** that you've just created **(1)** and select – **add database insert (2)**.



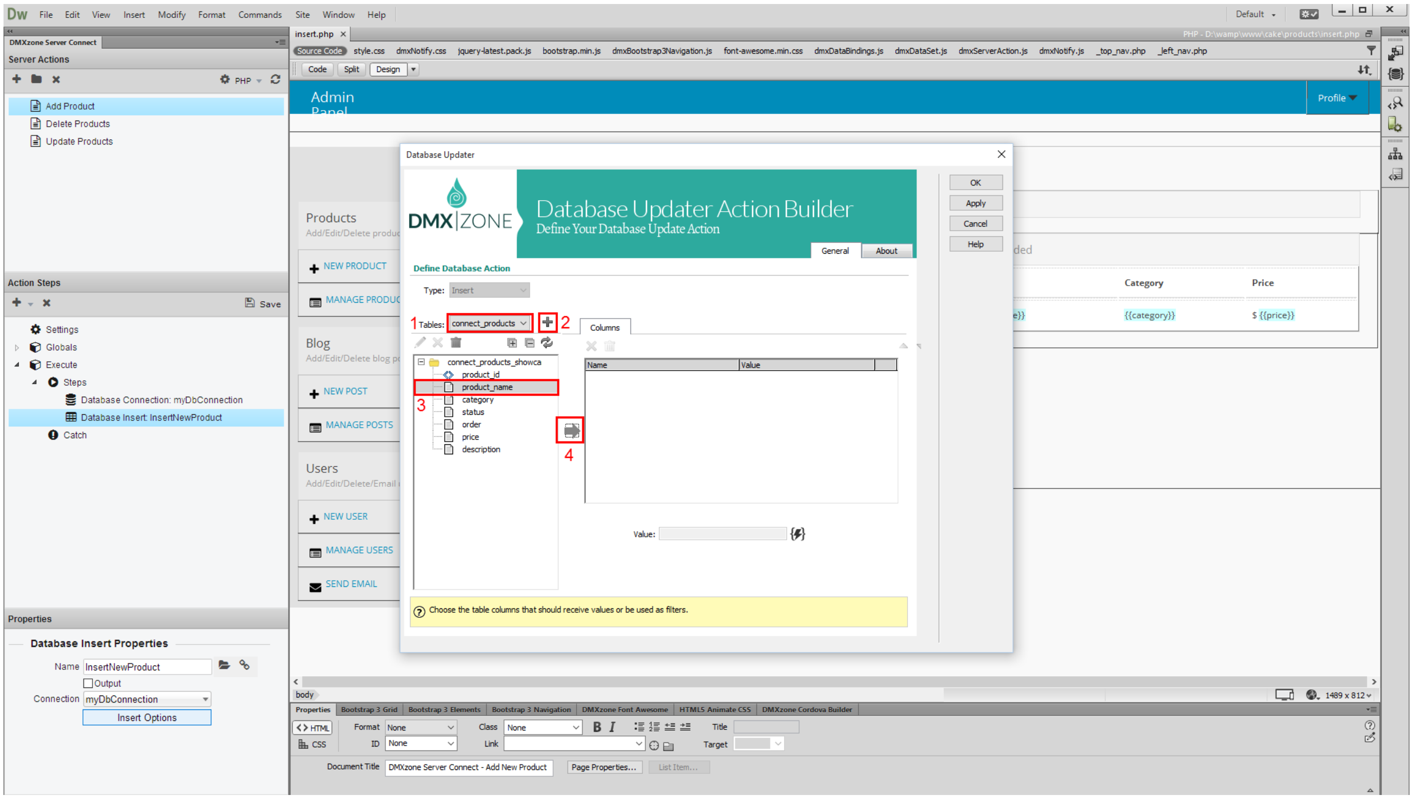
9. In the properties inspector **add a name** for this database insert step **(1)**, select your **database connection**, from the dropdown **(2)** and **click the insert options** in order to setup the database insert **(3)**.



Database Insert Properties

1 Name: InsertNewProduct
□ Output
2 Connection: myDbConnection
3 Insert Options

10. First, **select (1)** and **add the database table (2)**, where you want to insert the data. Then, **select the columns (3)**, that you want to **insert data into (4)**.



Type: Insert

Tables: connect_products

Value:

Choose the table columns that should receive values or be used as filters.

Name: InsertNewProduct

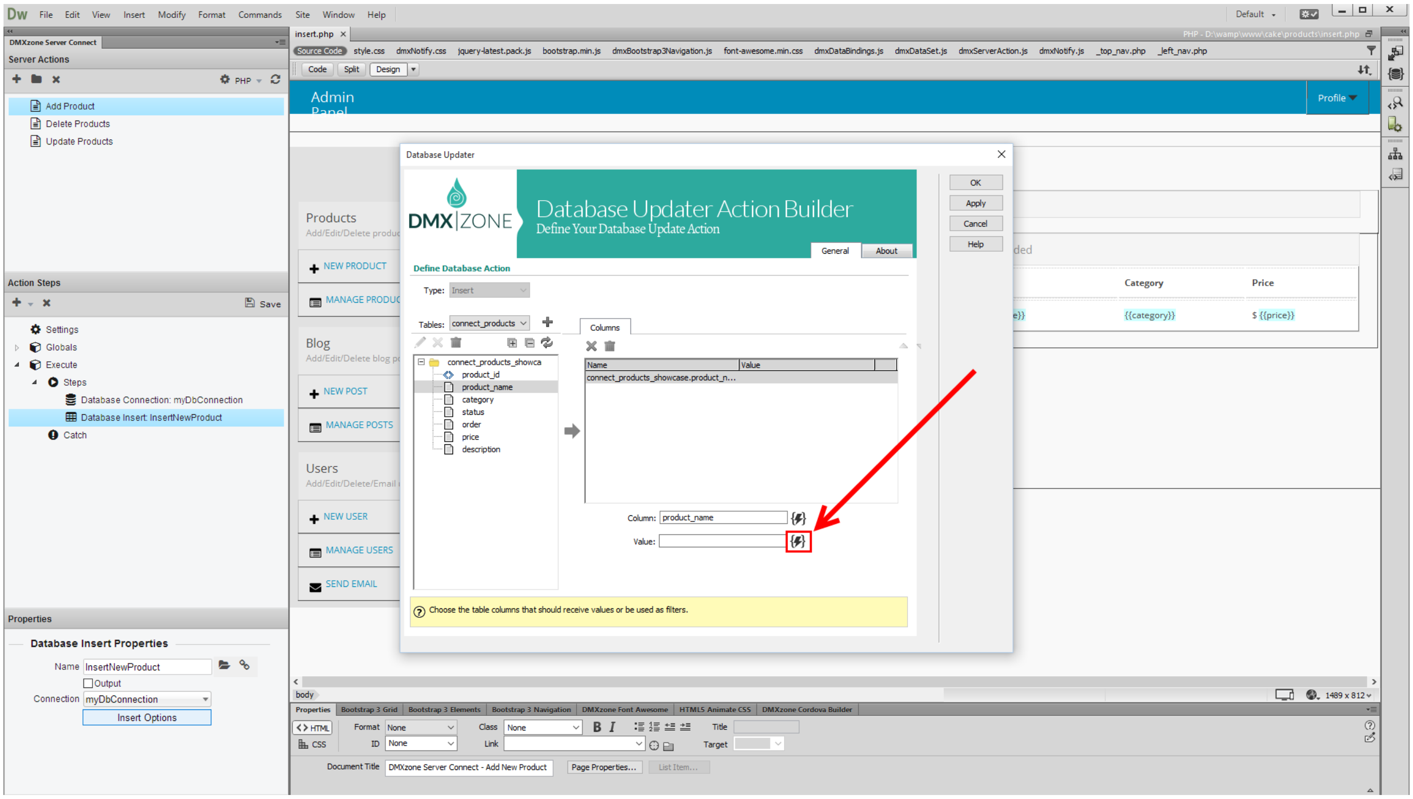
☐ Output

Connection: myDbConnection

Insert Options

Document Title: DMXzone Server Connect - Add New Product

11. Then, select the value that you want to insert into the selected column. Click the **server data bindings button.**



Type: Insert

Tables: connect_products

Name	Value
connect_products_showcase.product_n...	

Column: product_name

Value:

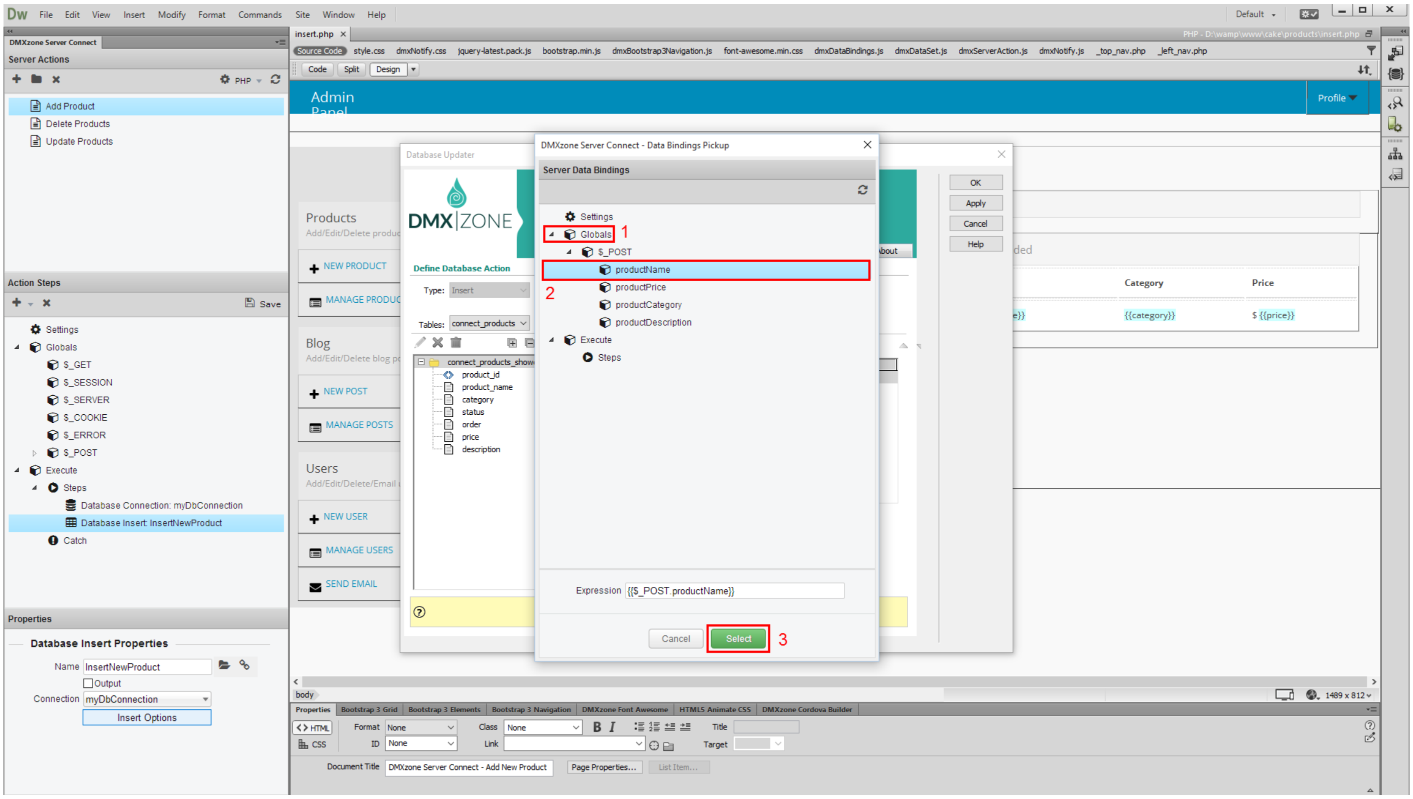
Choose the table columns that should receive values or be used as filters.

Name: InsertNewProduct

Connection: myDbConnection

Document Title: DMXzone Server Connect - Add New Product

12. **Expand globals (1)** and under post, select **the form input (2)** that you want to bind to this column. In our tutorial, this is the product name form input. When you're done, click the **select button (3)**.



Expression {{$_POST.productName}}

Name InsertNewProduct

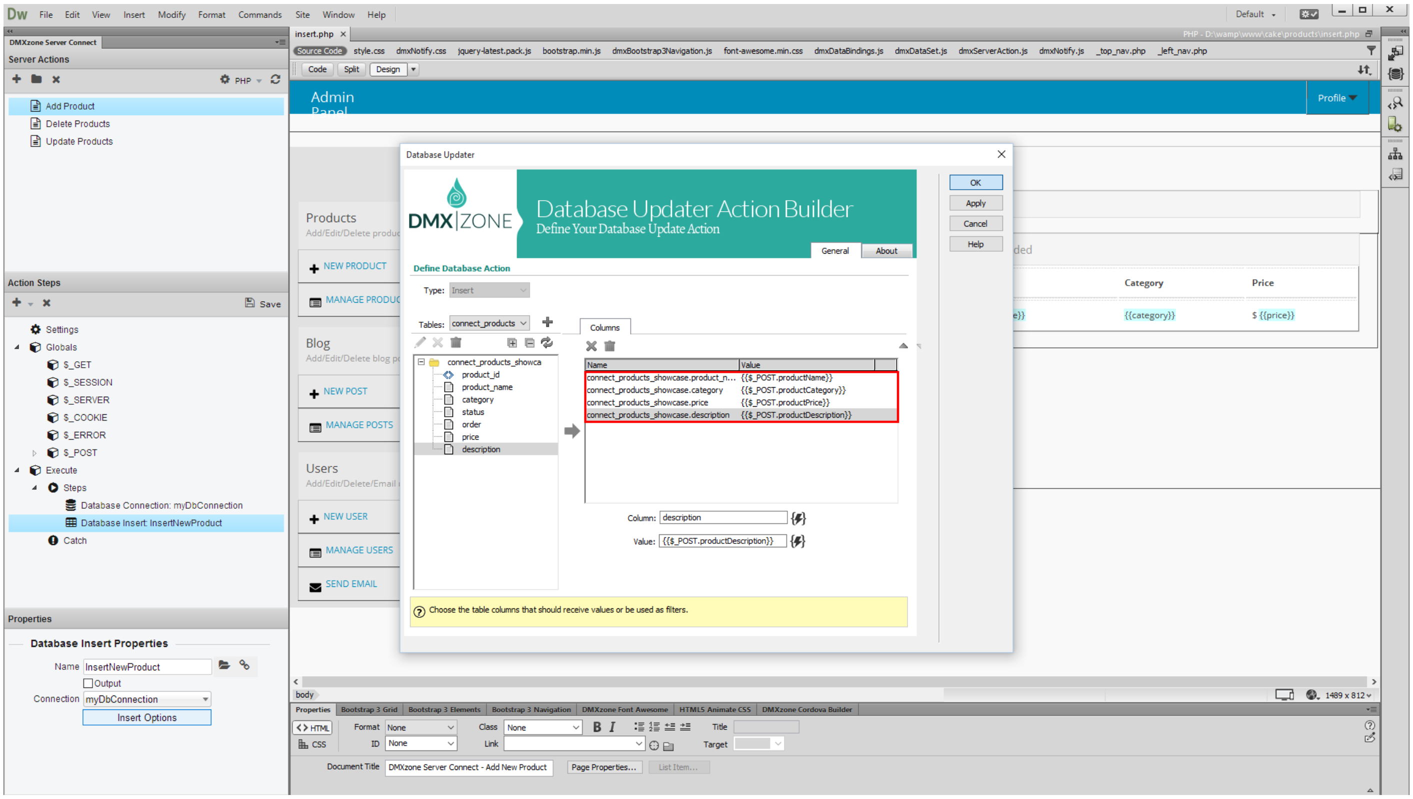
Connection myDbConnection

Type: Insert

Tables: connect_products

Document Title DMXzone Server Connect - Add New Product

13. We do the same for the rest of the columns, binding the proper form inputs to them. Click OK when you're done.



Database Updater

Database Updater Action Builder
Define Your Database Update Action

Define Database Action

Type: Insert

Tables: connect_products

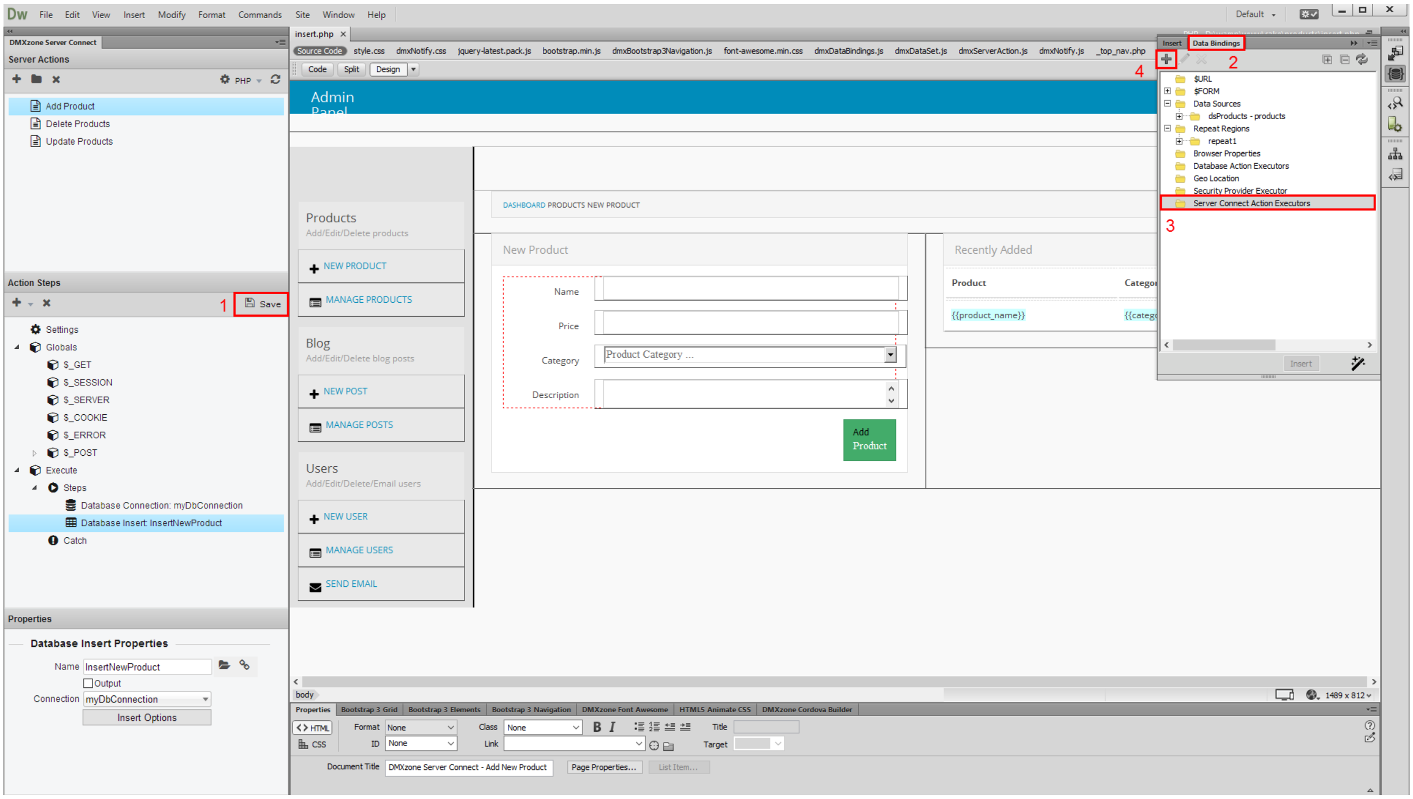
Name	Value
connect_products_showcase.product_n...	{{$_POST.productName}}
connect_products_showcase.category	{{$_POST.productCategory}}
connect_products_showcase.price	{{$_POST.productPrice}}
connect_products_showcase.description	{{$_POST.productDescription}}

Column: description

Value: {{$_POST.productDescription}}

Choose the table columns that should receive values or be used as filters.

14. Make sure to **save your action steps (1)**. Open the **HTML5 Data Bindings panel (2)**, select **server connect action executors (3)** and click **add new button (4)**.



Database Insert Properties

Name: InsertNewProduct

☐ Output

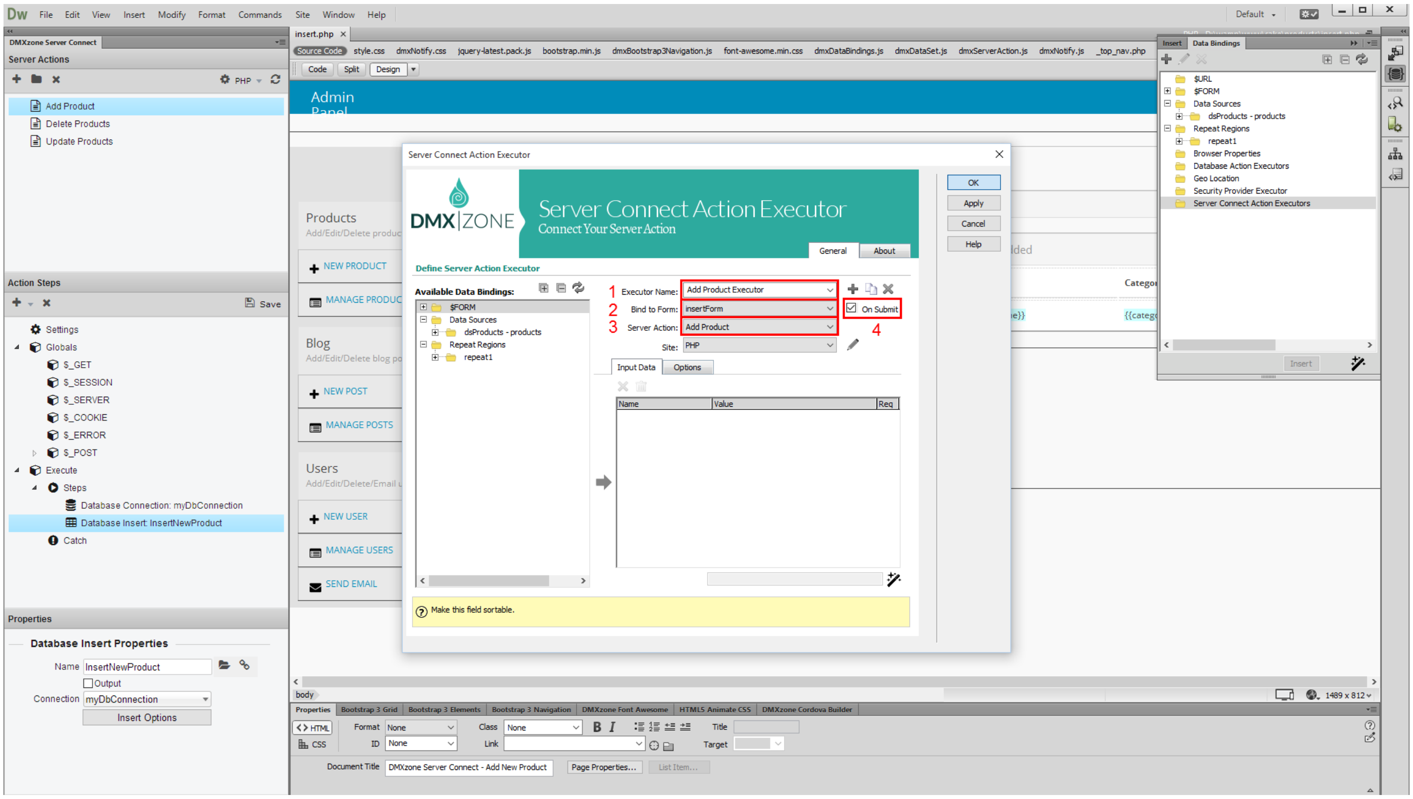
Connection: myDbConnection

Insert Options

Format: None   Class: None   Title:
ID: None   Link: Target:

Document Title: DMXzone Server Connect - Add New Product

15. Select your **server action (1)**, from the dropdown, **bind this server action executor** to the form on your page **(2)** and **enter a name for the executor (3)**. We'd like to run it, when the form is submitted so we **check the On Submit checkbox (4)**. Click the OK Button, when you are done.



Server Connect Action Executor

Define Server Action Executor

1 Executor Name: Add Product Executor

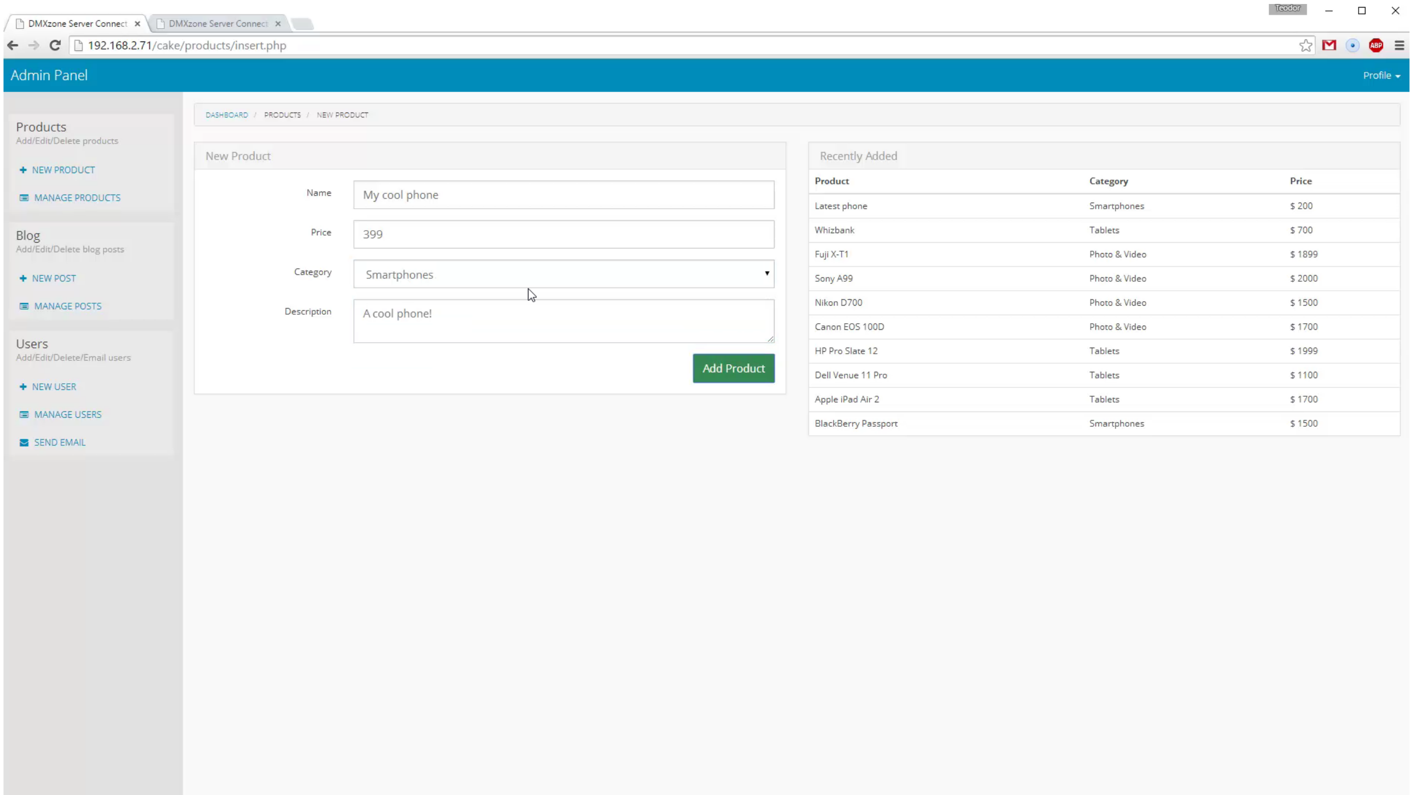
2 Bind to Form: insertForm

3 Server Action: Add Product

Site: PHP

4 ☑ On Submit

16. Let’s preview the results.



New Product

Name: My cool phone

Price: 399

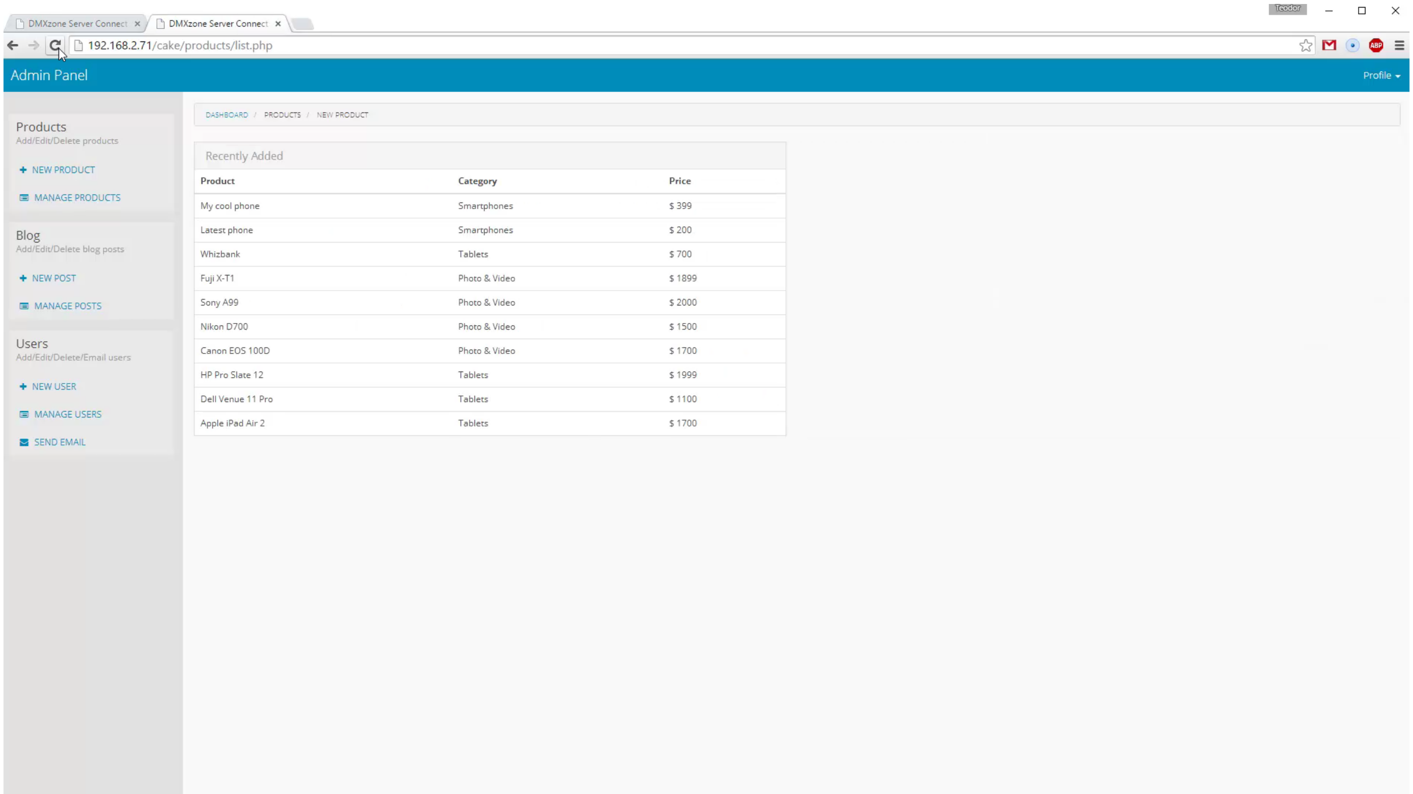
Category: Smartphones

Description: A cool phone!

Recently Added

Product	Category	Price
Latest phone	Smartphones	$ 200
Whizbank	Tablets	$ 700
Fuji X-T1	Photo & Video	$ 1899
Sony A99	Photo & Video	$ 2000
Nikon D700	Photo & Video	$ 1500
Canon EOS 100D	Photo & Video	$ 1700
HP Pro Slate 12	Tablets	$ 1999
Dell Venue 11 Pro	Tablets	$ 1100
Apple iPad Air 2	Tablets	$ 1700
BlackBerry Passport	Smartphones	$ 1500

17. So, we just enter the name, price, description and category and click the submit button. The form submits are using AJAX so the page does not reload. You can see the new product on our latest products page. That's how easy it is to insert database records, using DMXzone Server Connect.



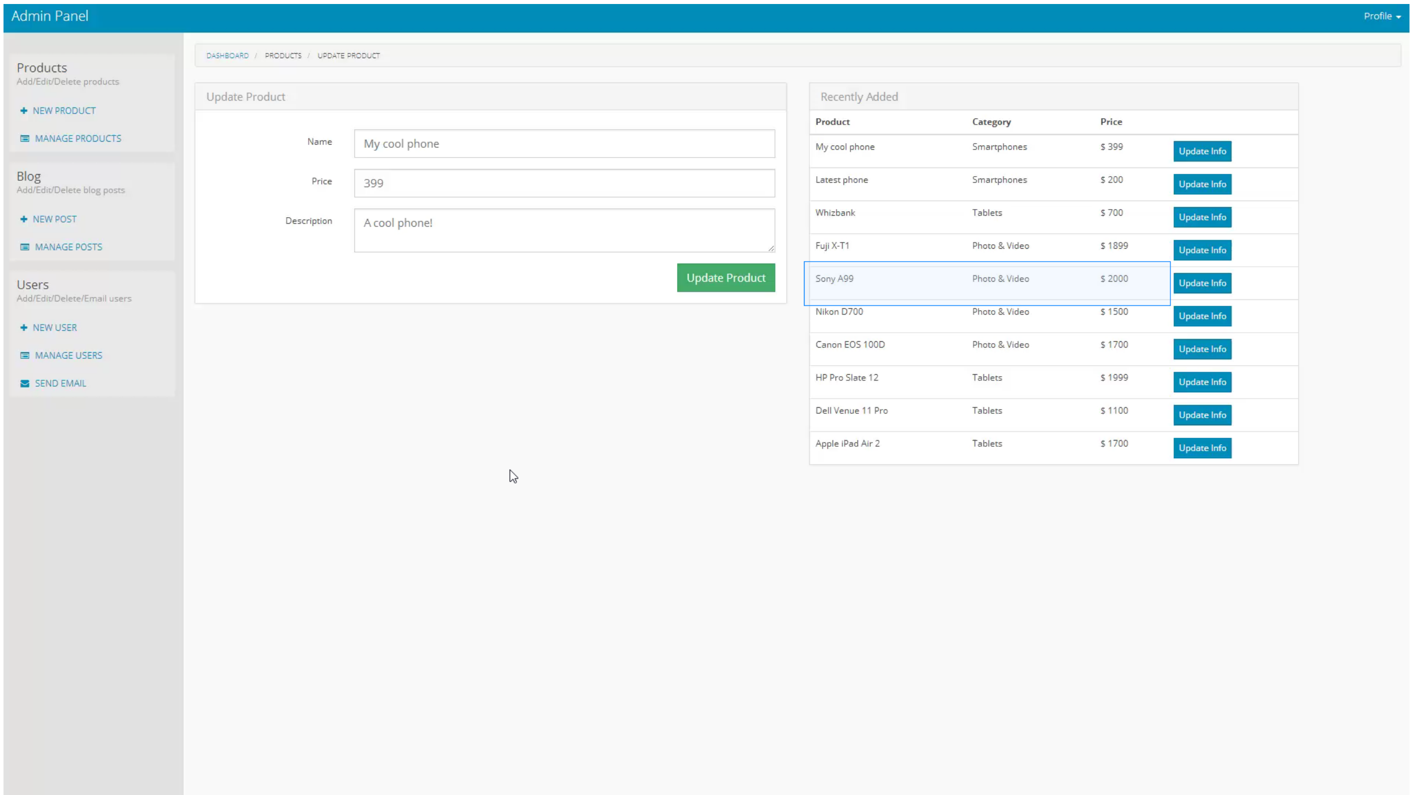
Recently Added

Product	Category	Price
My cool phone	Smartphones	$ 399
Latest phone	Smartphones	$ 200
Whizbank	Tablets	$ 700
Fuji X-T1	Photo & Video	$ 1899
Sony A99	Photo & Video	$ 2000
Nikon D700	Photo & Video	$ 1500
Canon EOS 100D	Photo & Video	$ 1700
HP Pro Slate 12	Tablets	$ 1999
Dell Venue 11 Pro	Tablets	$ 1100
Apple iPad Air 2	Tablets	$ 1700

Advanced: Update Database Records

In this tutorial we will show you how update database records, using DMXzone Server Connect. In order to do that you'll also need HTML5 Data Bindings, DMXzone Database Connector PHP/ASP and DMXzone Database Updater PHP/ASP extensions.

*You can also check out the video.



Update Product

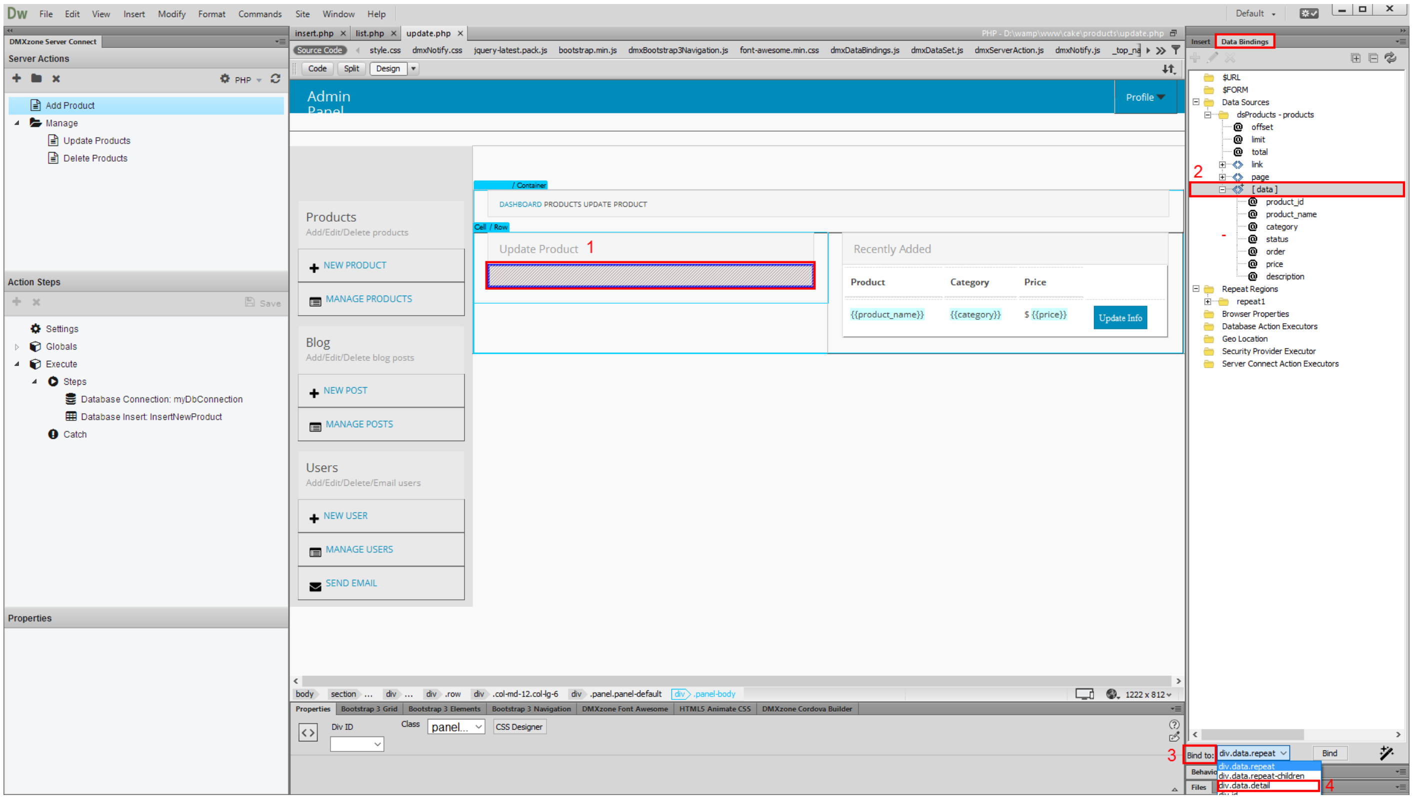
Name	My cool phone
Price	399
Description	A cool phone!

Recently Added

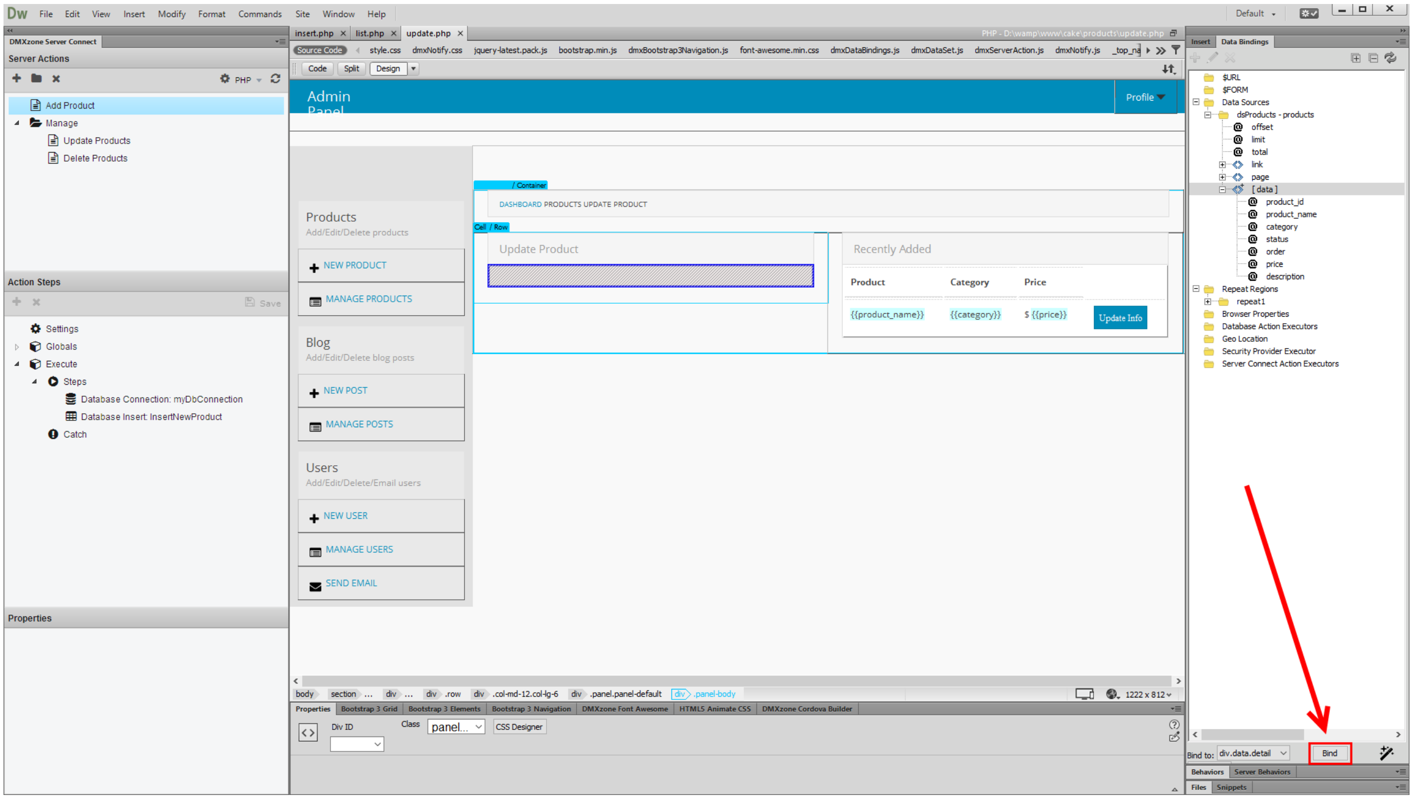
Product	Category	Price	
My cool phone	Smartphones	$ 399	Update Info
Latest phone	Smartphones	$ 200	Update Info
Whizbank	Tablets	$ 700	Update Info
Fuji X-T1	Photo & Video	$ 1899	Update Info
Sony A99	Photo & Video	$ 2000	Update Info
Nikon D700	Photo & Video	$ 1500	Update Info
Canon EOS 100D	Photo & Video	$ 1700	Update Info
HP Pro Slate 12	Tablets	$ 1999	Update Info
Dell Venue 11 Pro	Tablets	$ 1100	Update Info
Apple iPad Air 2	Tablets	$ 1700	Update Info

How to do it

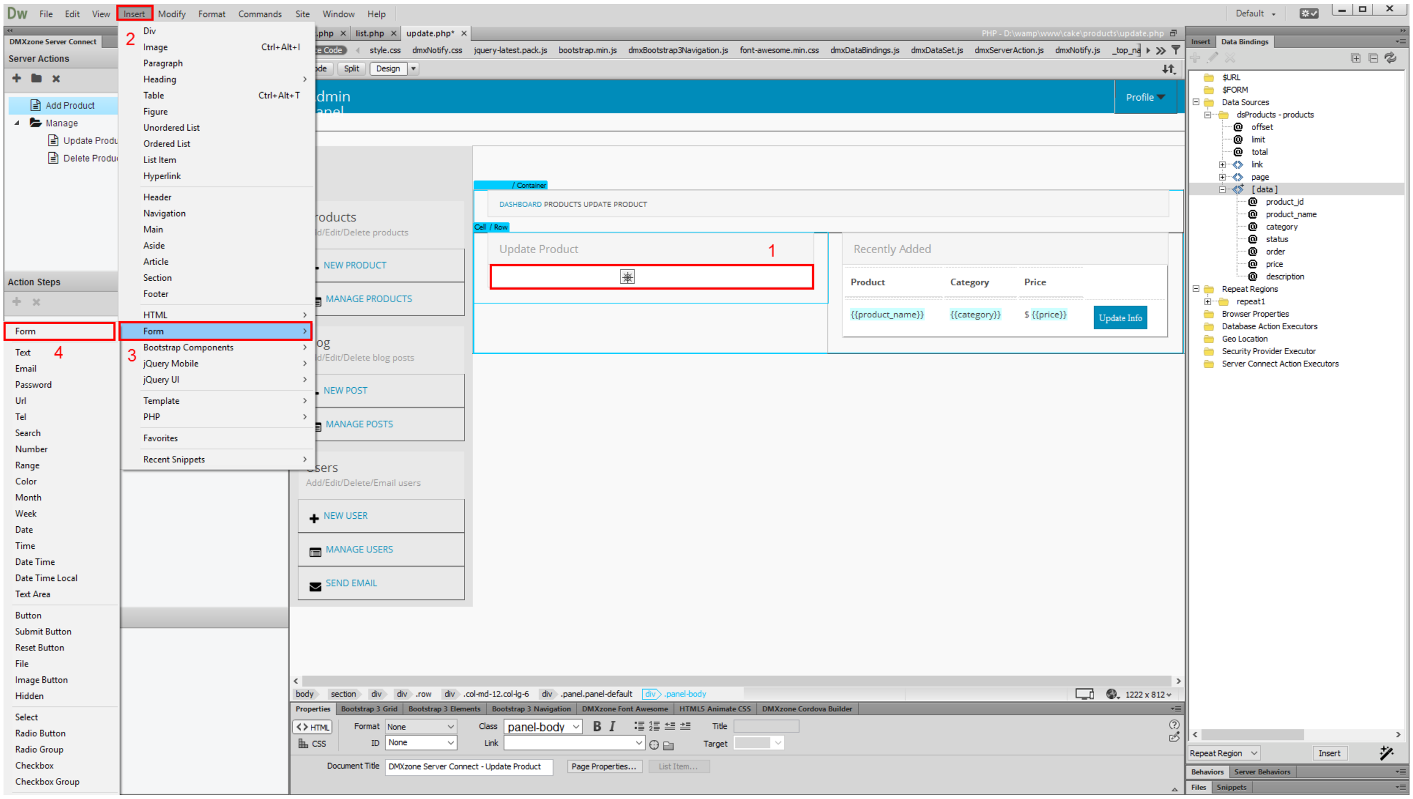
1. On this page we inserted a repeat region that lists our products. There is also a simple button, inside the repeat region. We will use this button to select the record that we want to update. First, we need to setup a detail region, where the update form will be placed. We select the **div container** on our page **(1)**, where we plan to insert the form. In the HTML5 Data Bindings panel (2), we select the repeating element of our data source (3) and from the **bind To menu** (4) we select **div data detail (5)**. This turns our div container into a detail region for our repeat region.



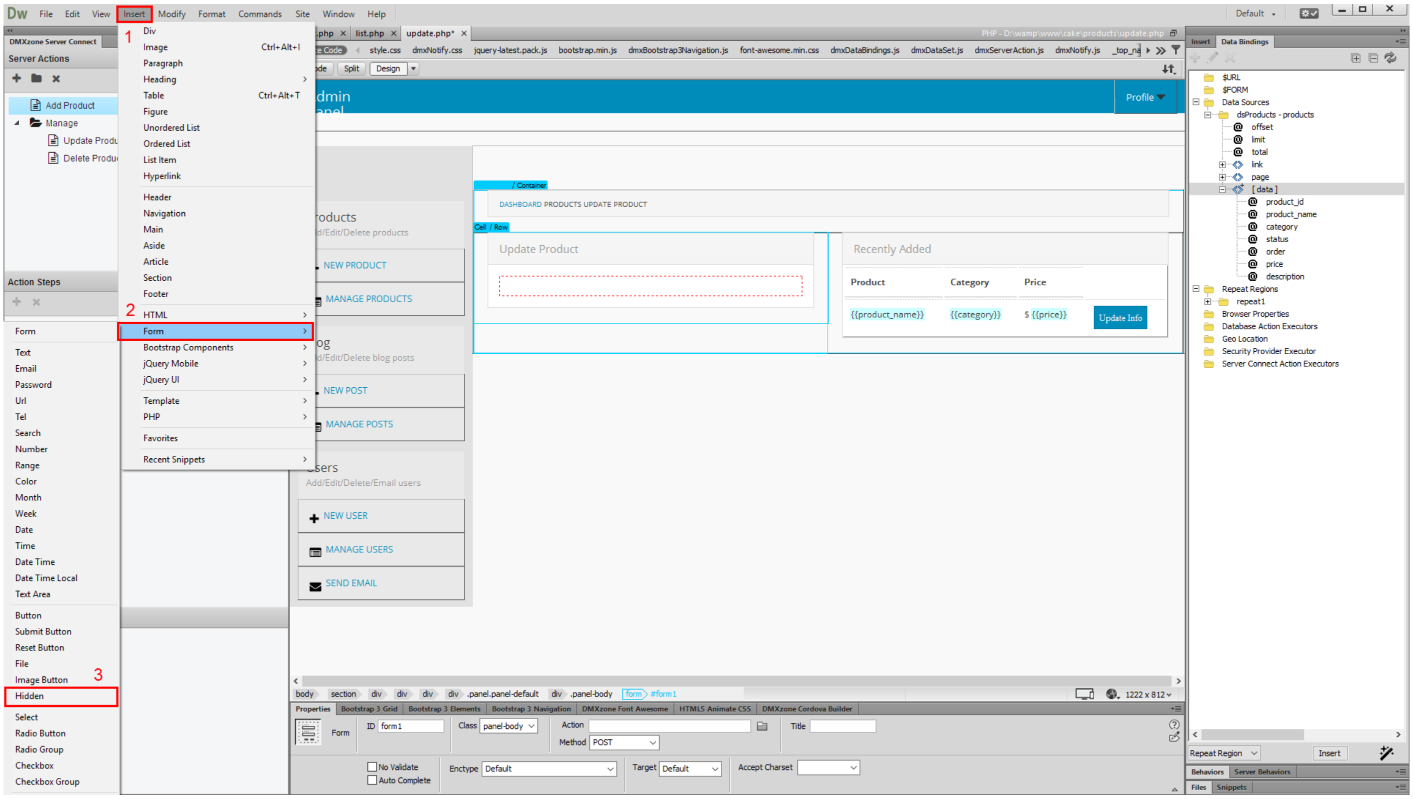
2. Click the **bind button** in order to apply this.



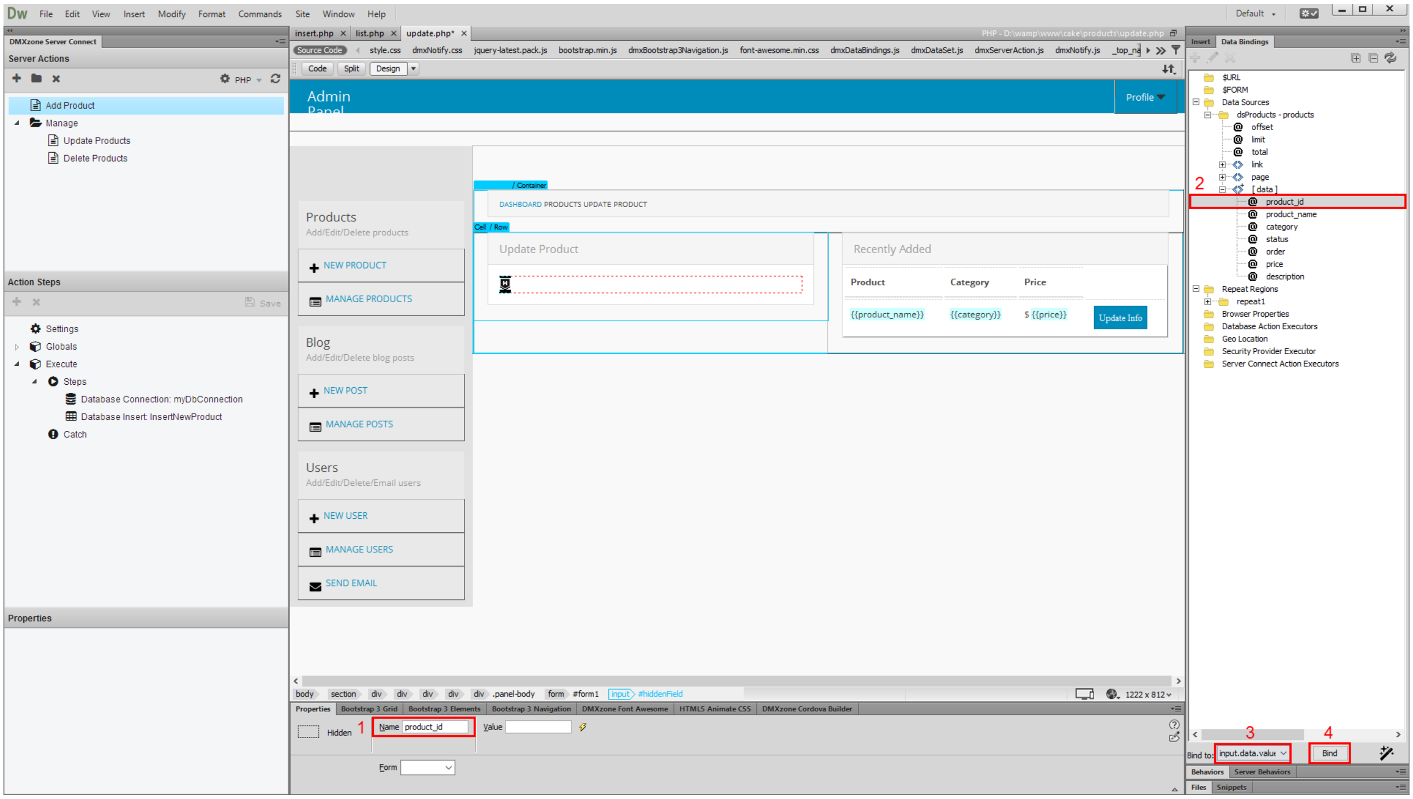
3. Now, click **inside the detail region (1)** that you've just created. From the **insert menu (2)**, open the **form sub menu (3)**, and click **form (4)**.



4. Let’s add a hidden field inside the form. From the **insert menu (1)**, open the **form sub menu (2)**, and click **hidden field (3)**.



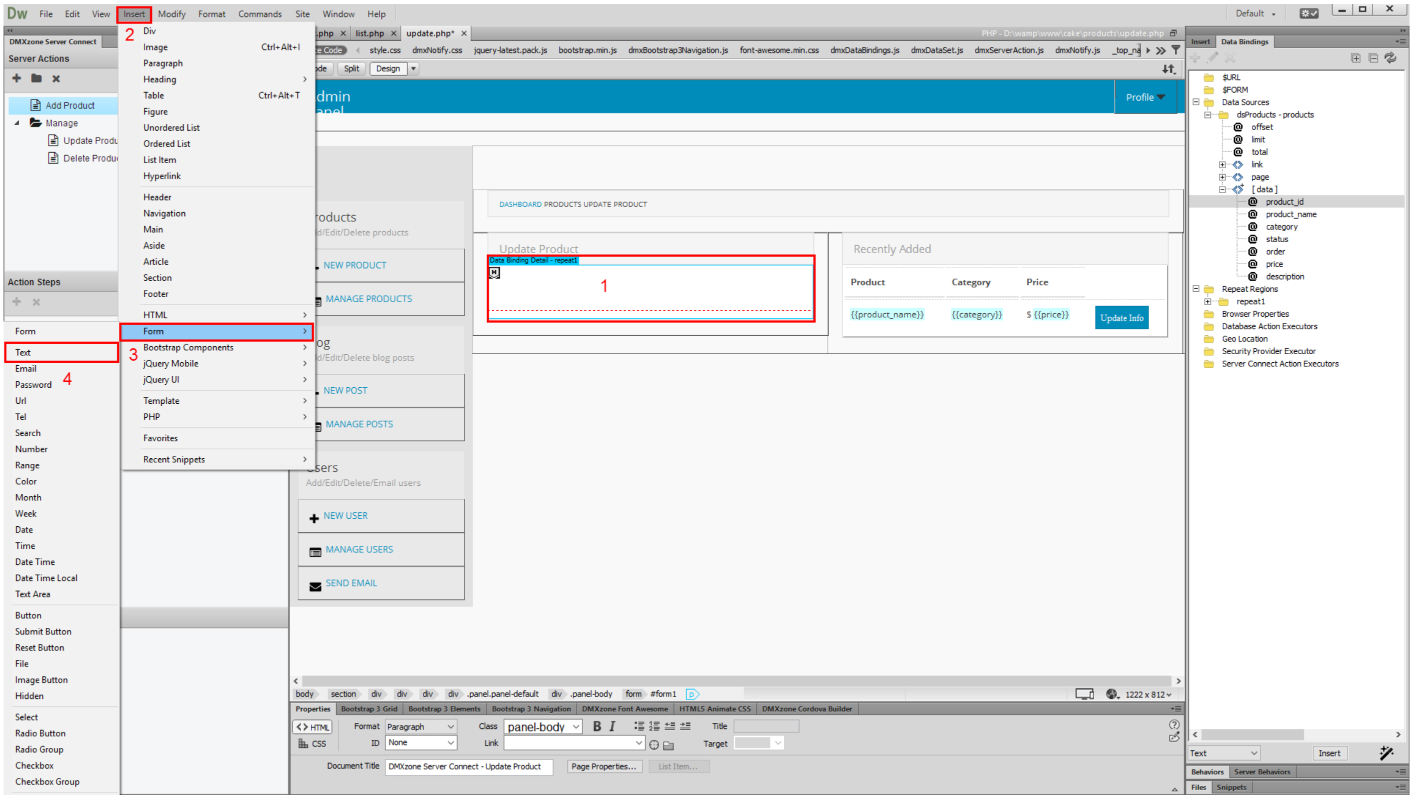
5. We **change the default name** of this hidden field to product ID **(1)**. We will use this hidden field in order to filter the records for the update action. Then, with the hidden field still selected, we click the **product ID binding** from our data source **(2)**. We **bind (4)** this to the **input.data.value (3)**.



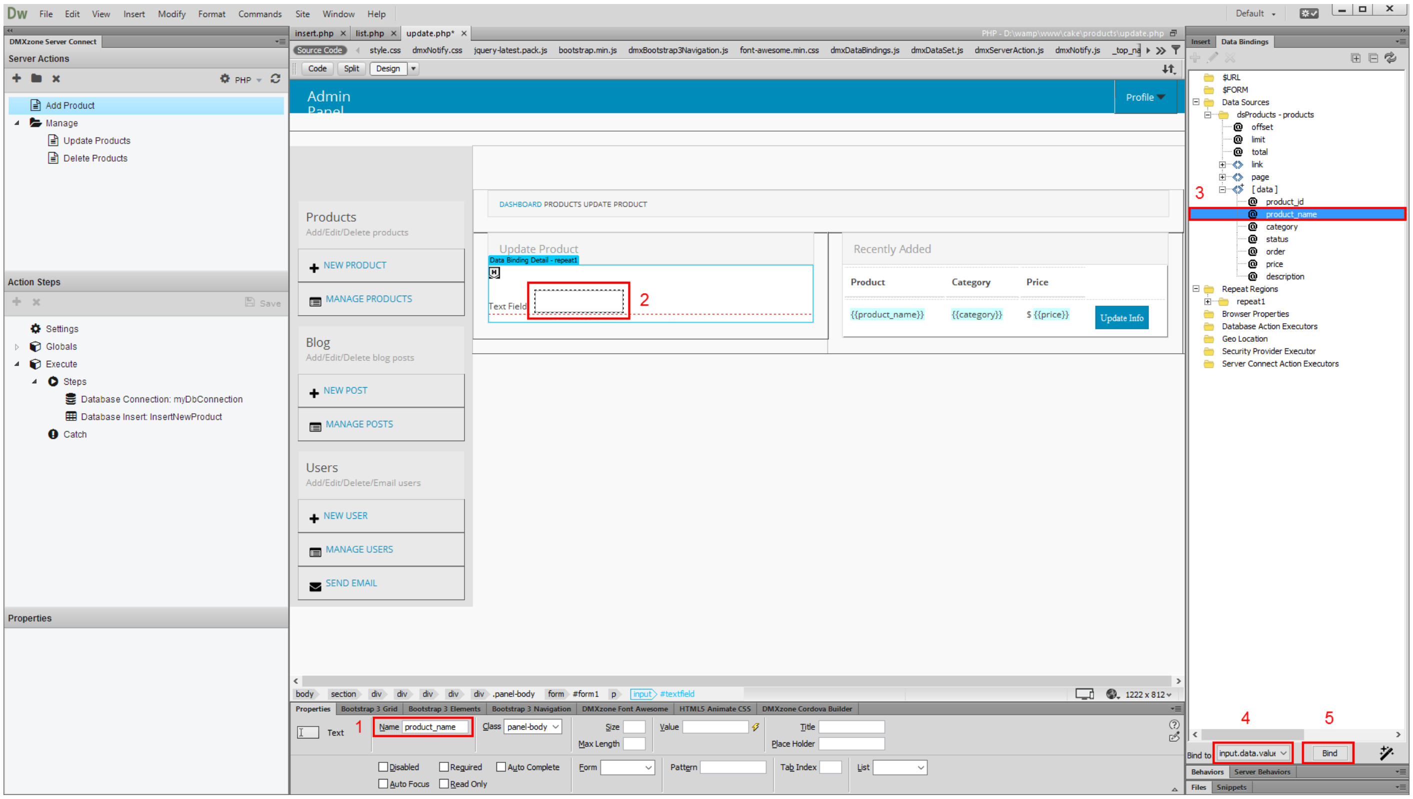
Name: product_id

Bind to: input.data.value

6. Now, let’s add the rest of the form fields that we will use in order to update the products info. Click **inside the form (1)** and open the **insert menu (2)**. Under the **form sub menu (3)**, select **text (4)**. This adds a text field inside the detail region.

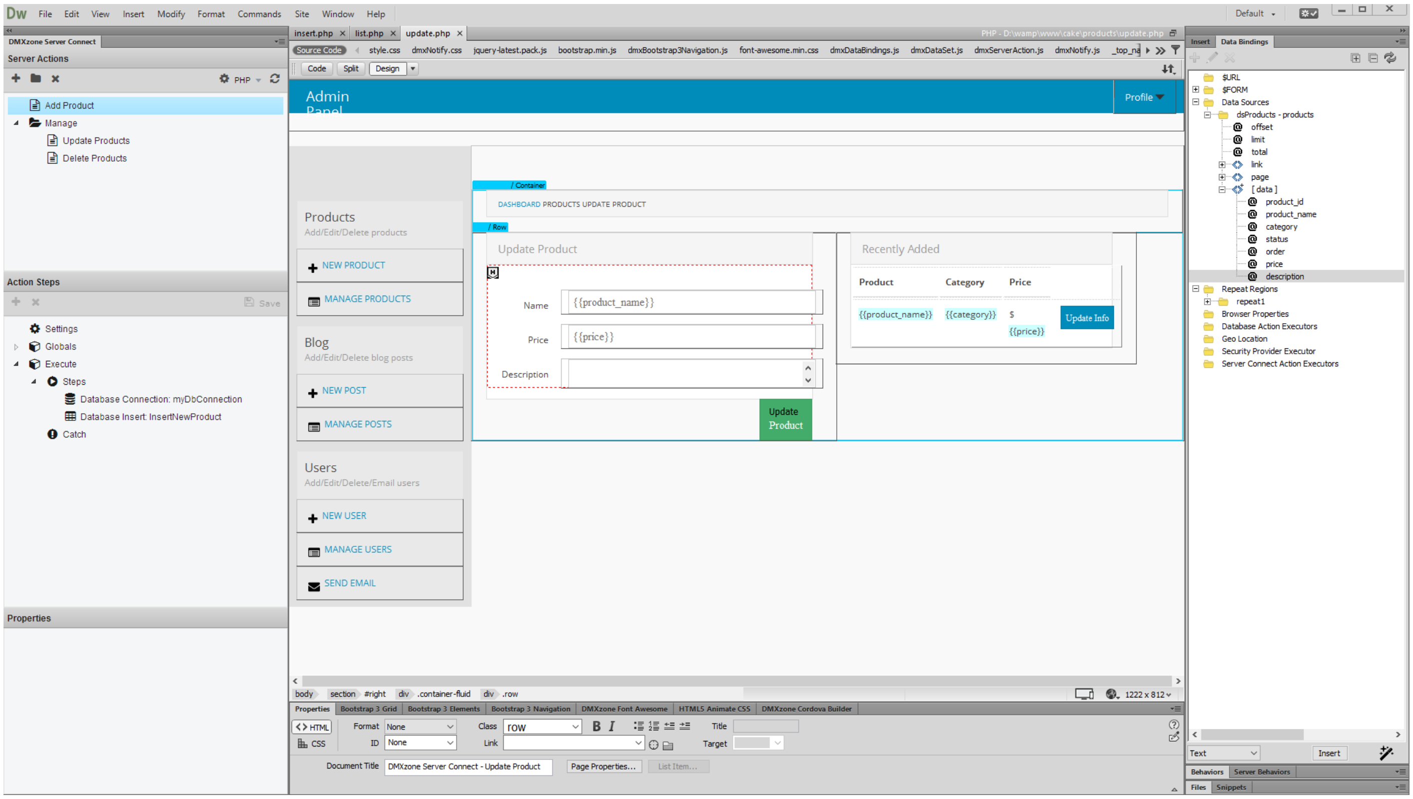


7. We **change its name** to product name **(1)**. We will use this field to display and update the name of the selected product. With the **field still selected (2)**, click the **product name binding (2)** from our data source, and **bind (5)** it to **input.data.value** (4).

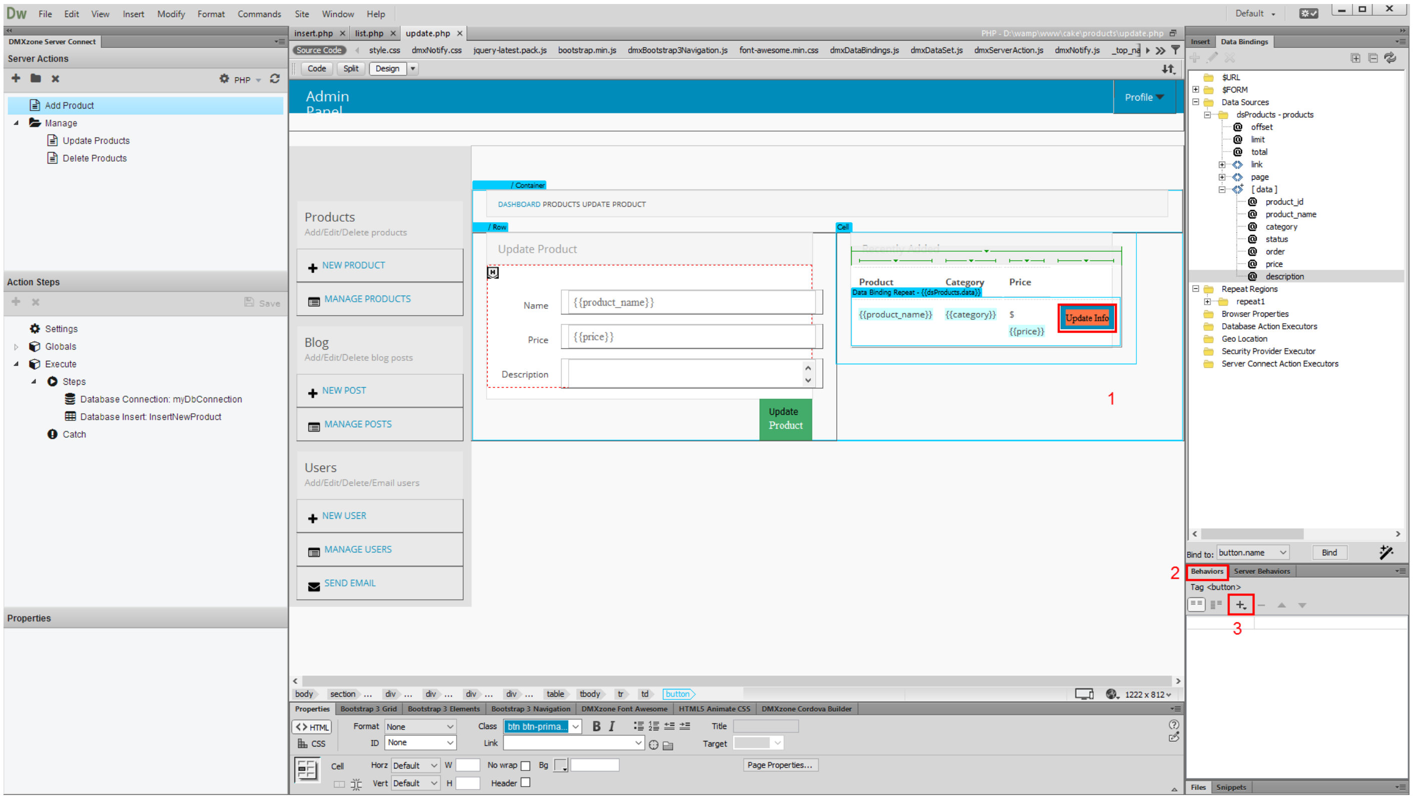


Name: product_name
Class: panel-body
Bind to: input.data.value

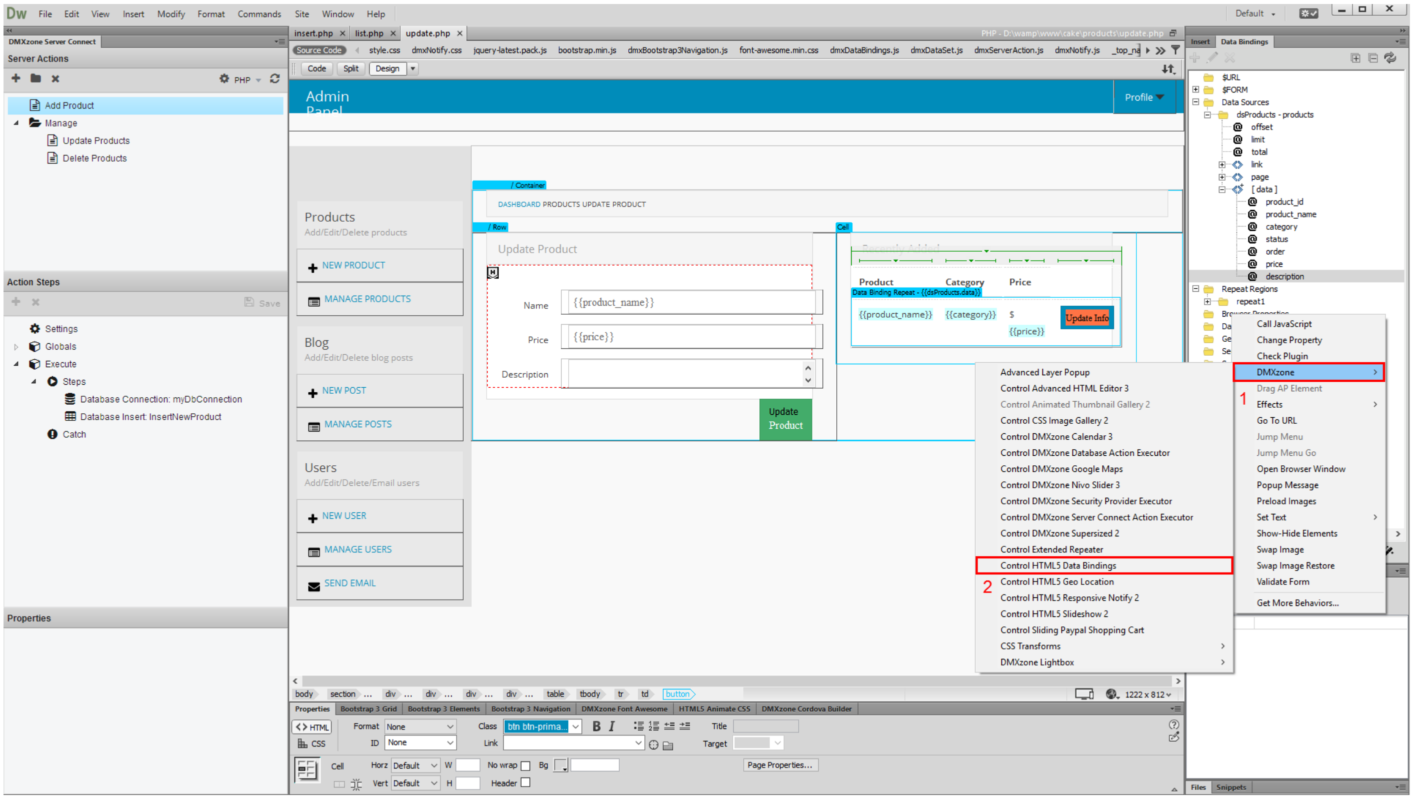
8. We do the same with the rest of the fields that we need in this form. When we are done adding the text fields inside the form, we also add a submit button, so we can submit the form data.


Dw File Edit View Insert Modify Format Commands Site Window Help
DMXzone Server Connect
Server Actions
PHP
Add Product
Manage
Update Products
Delete Products
Action Steps
Save
Settings
Globals
Execute
Steps
Database Connection: myDbConnection
Database Insert: InsertNewProduct
Catch
Properties
insert.php list.php update.php
Source Code style.css dmxNotify.css jquery-latest.pack.js bootstrap.min.js dmxBootstrap3Navigation.js font-awesome.min.css dmxDataBindings.js dmxDataSet.js dmxServerAction.js dmxNotify.js
Code Split Design
Admin Panel
Profile
Products
Add/Edit/Delete products
NEW PRODUCT
MANAGE PRODUCTS
Blog
Add/Edit/Delete blog posts
NEW POST
MANAGE POSTS
Users
Add/Edit/Delete/Email users
NEW USER
MANAGE USERS
SEND EMAIL
DASHBOARD PRODUCTS UPDATE PRODUCT
Update Product
Name {{product_name}}
Price {{price}}
Description
Update Product
Recently Added
Product Category Price
{{product_name}} {{category}} $ {{price}}
Update Info
body section #right div .container-fluid div .row
1222 x 812
Properties Bootstrap 3 Grid Bootstrap 3 Elements Bootstrap 3 Navigation DMXzone Font Awesome HTML5 Animate CSS DMXzone Cordova Builder
HTML CSS Format None ID None Class row Link Title Target
Document Title DMXzone Server Connect - Update Product
Page Properties... List Item...
Default
Insert Data Bindings
$URL
$FORM
Data Sources
dsProducts - products
offset
limit
total
link
page
[data]
product_id
product_name
category
status
order
price
description
Repeat Regions
repeat1
Browser Properties
Database Action Executors
Geo Location
Security Provider Executor
Server Connect Action Executors
Text Insert
Behaviors Server Behaviors
Files Snippets

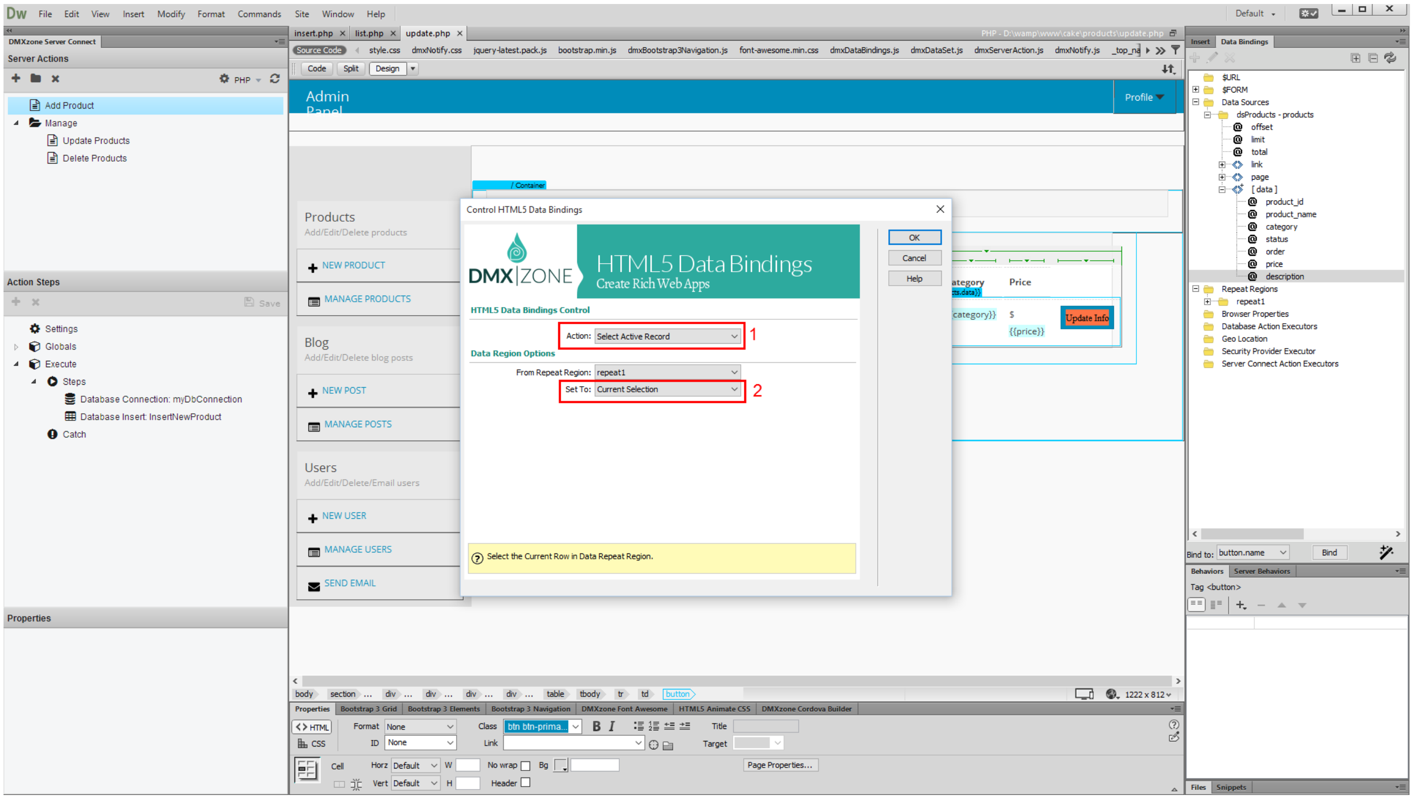
9. Now, click the **button (1)** that is located inside the repeat region. Open the **behaviors panel (2)** and **add a new behavior** for this button **(3)**.



10. Under **DMXzone (1)**, select **control HTML5 Data Bindings (2)**.


Call JavaScript
Change Property
Check Plugin
DMXzone
Drag AP Element
Effects
Go To URL
Jump Menu
Jump Menu Go
Open Browser Window
Popup Message
Preload Images
Set Text
Show-Hide Elements
Swap Image
Swap Image Restore
Validate Form
Get More Behaviors...
1
Advanced Layer Popup
Control Advanced HTML Editor 3
Control Animated Thumbnail Gallery 2
Control CSS Image Gallery 2
Control DMXzone Calendar 3
Control DMXzone Database Action Executor
Control DMXzone Google Maps
Control DMXzone Nivo Slider 3
Control DMXzone Security Provider Executor
Control DMXzone Server Connect Action Executor
Control DMXzone Supersized 2
Control Extended Repeater
Control HTML5 Data Bindings
Control HTML5 Geo Location
Control HTML5 Responsive Notify 2
Control HTML5 Slideshow 2
Control Sliding Paypal Shopping Cart
CSS Transforms
DMXzone Lightbox
2
Update Product
Name
{{product_name}}
Price
{{price}}
Description
Update Product
Product
Category
Price
{{product_name}}
{{category}}
{{price}}
Update Info

11. From the action dropdown, choose - **select active record (1)** and **current selection** from the set to dropdown **(2)**. Click the OK button, when you are done. Now, when you click the update info button in the repeat region, it will populate the info of the selected record inside the detail region.



Control HTML5 Data Bindings

HTML5 Data Bindings Control

Action: Select Active Record

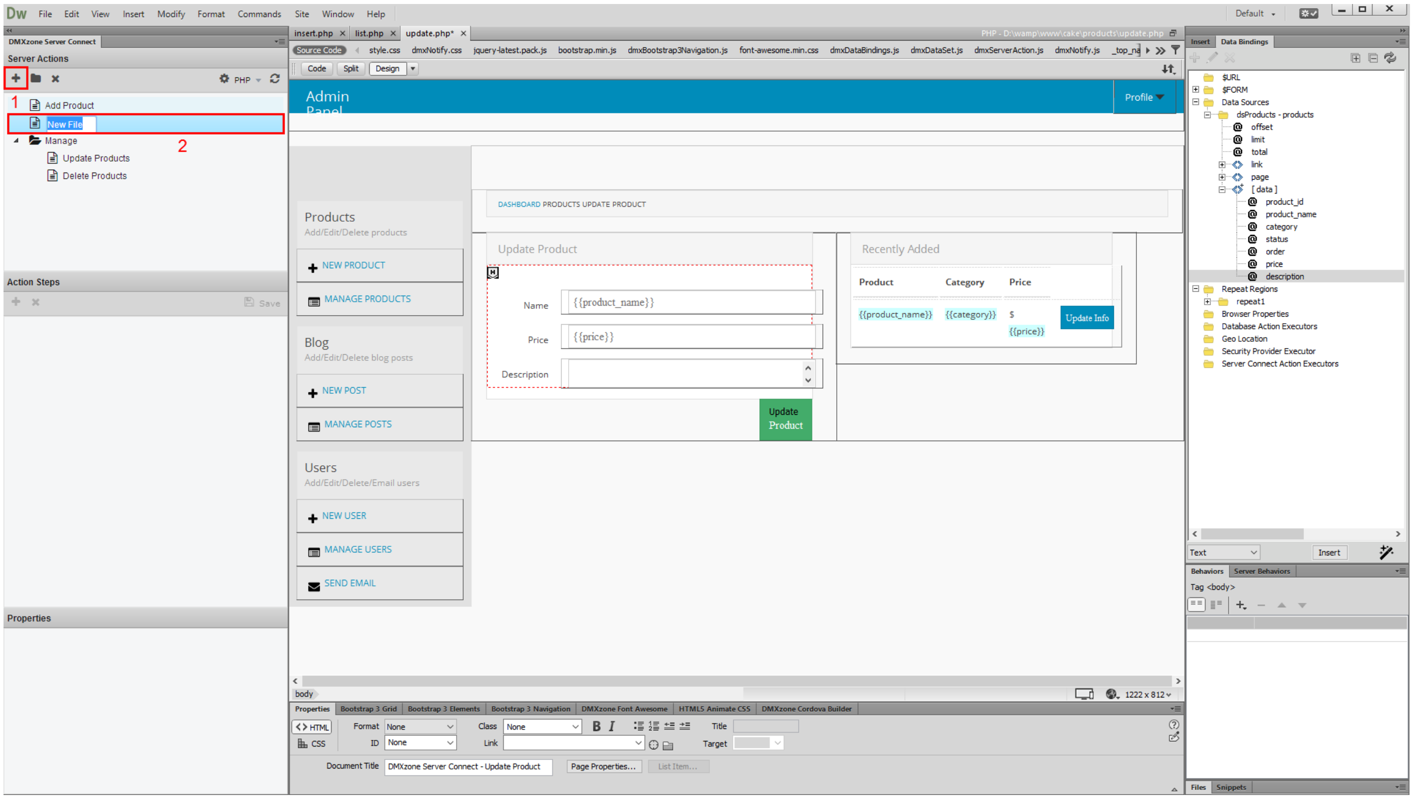
Data Region Options

From Repeat Region: repeat1

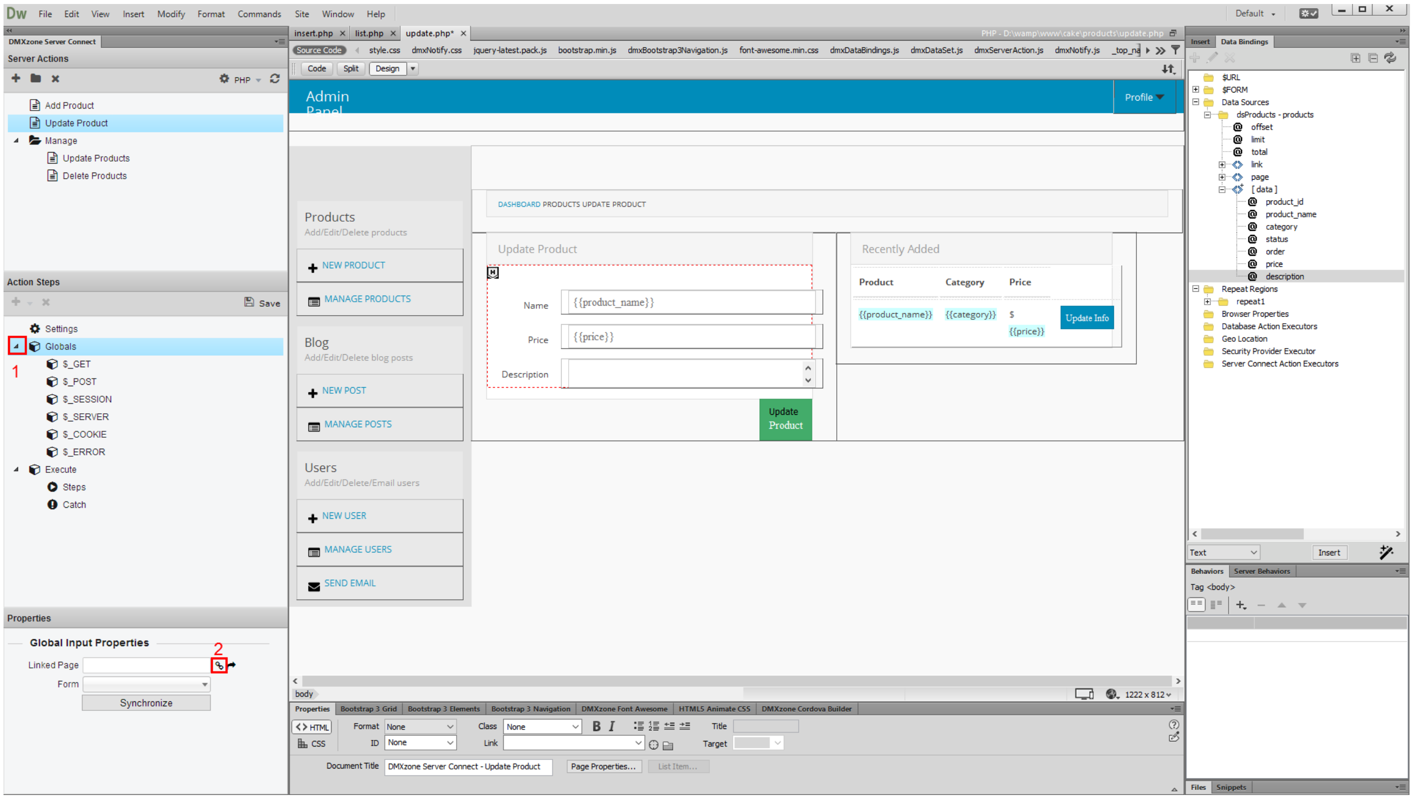
Set To: Current Selection

Select the Current Row in Data Repeat Region.

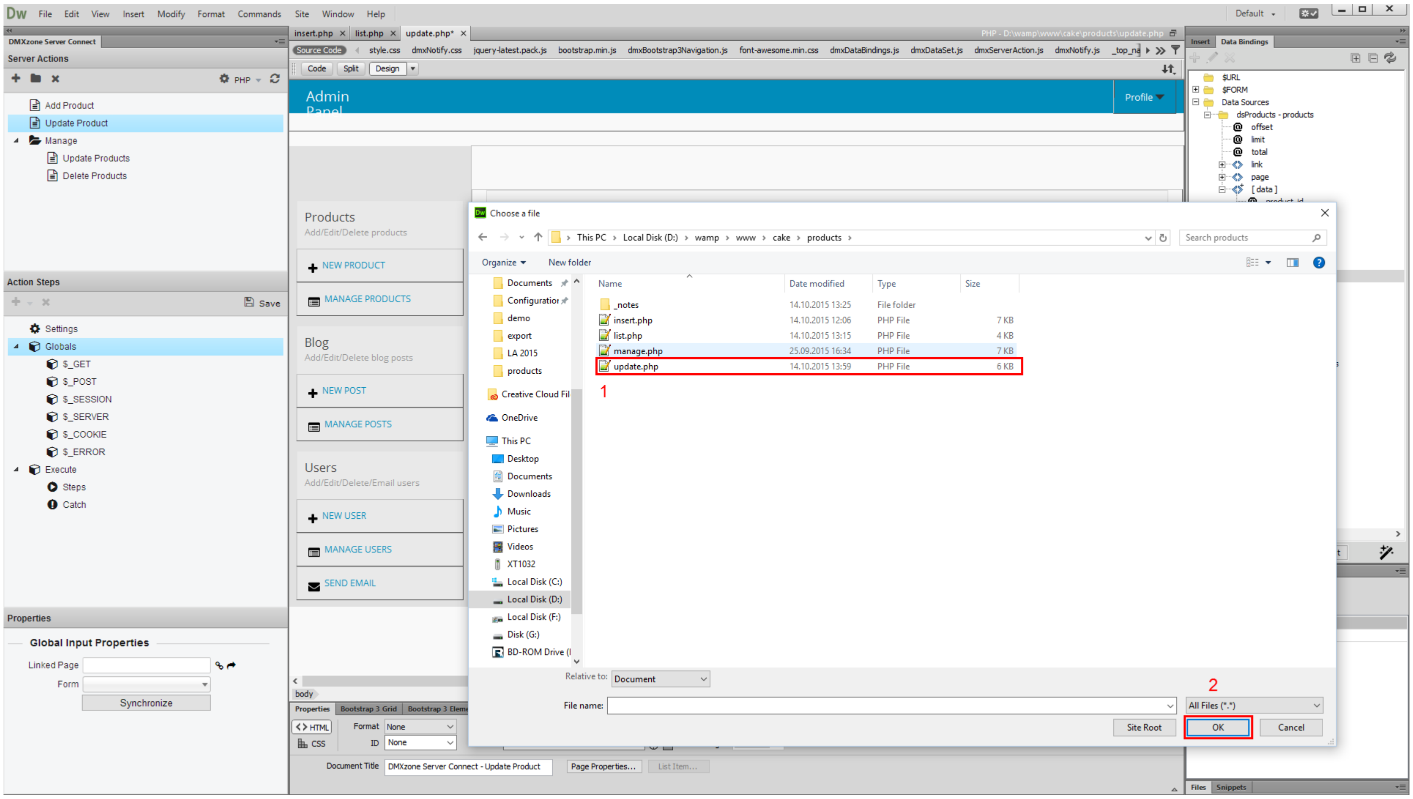
12. Now, let’s create the update record server action. Click **add new server action button (1)** and **add a name** for this action **(2)**.



13. Expand the **global variables (1)** in the action steps list and **link your update record page (2)** to them.



14. Select your **page (1)** and **click OK (2)**.



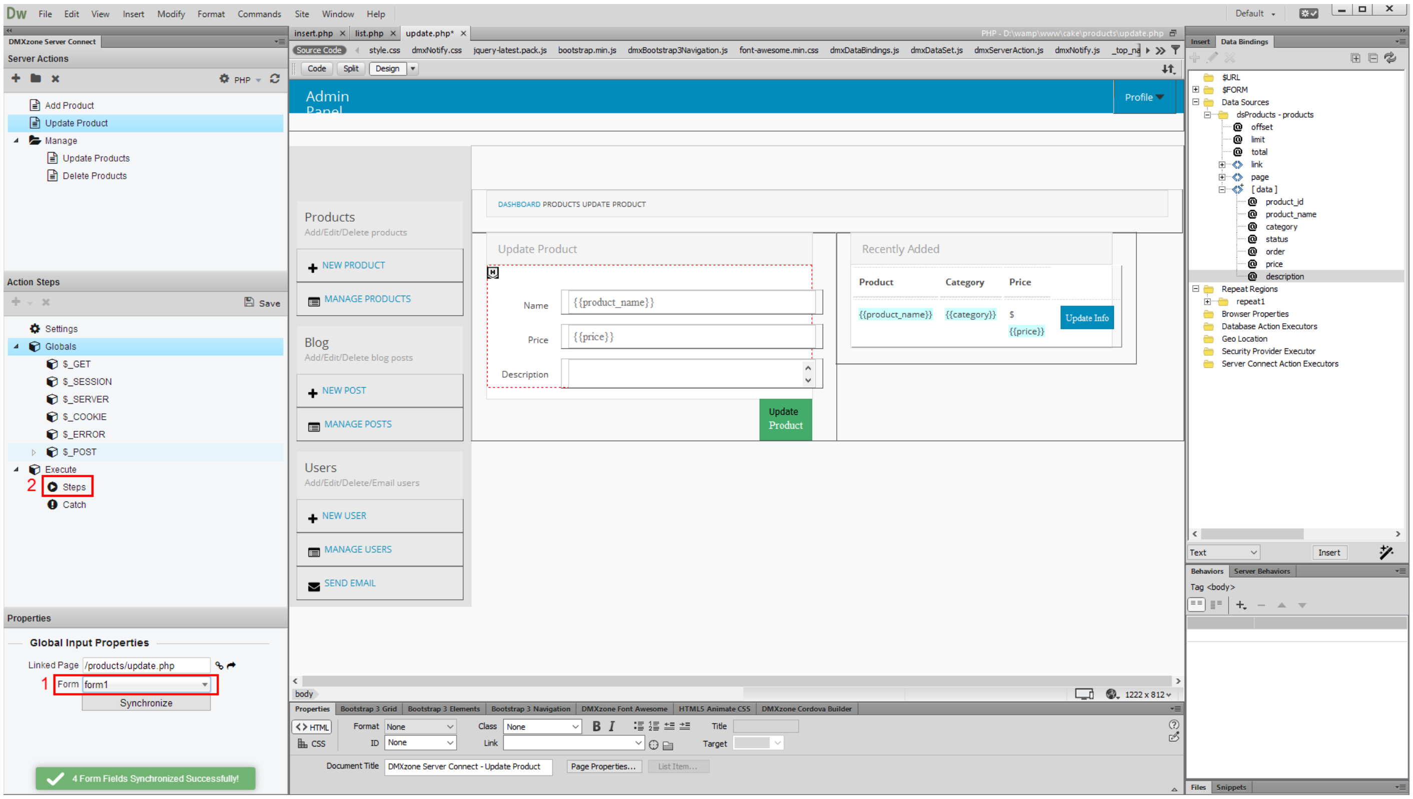
Name	Date modified	Type	Size
_notes	14.10.2015 13:25	File folder	
insert.php	14.10.2015 12:06	PHP File	7 KB
list.php	14.10.2015 13:15	PHP File	4 KB
manage.php	25.09.2015 16:34	PHP File	7 KB
update.php	14.10.2015 13:59	PHP File	6 KB

Relative to: Document

File name:

All Files (*.*)

15. From the global input properties, select the **update record form (1)** and **right click steps**, under the execute list **(2)**.

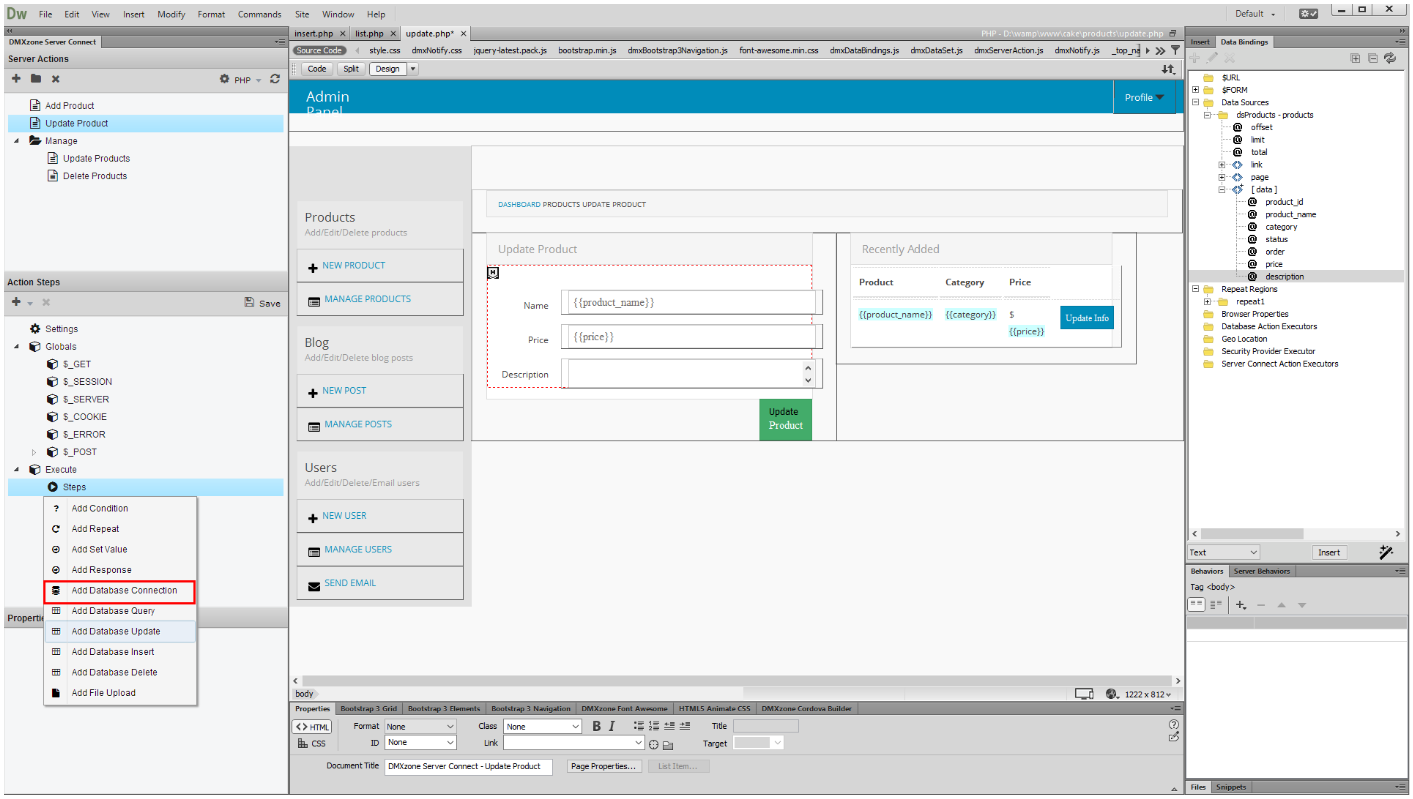


Linked Page /products/update.php

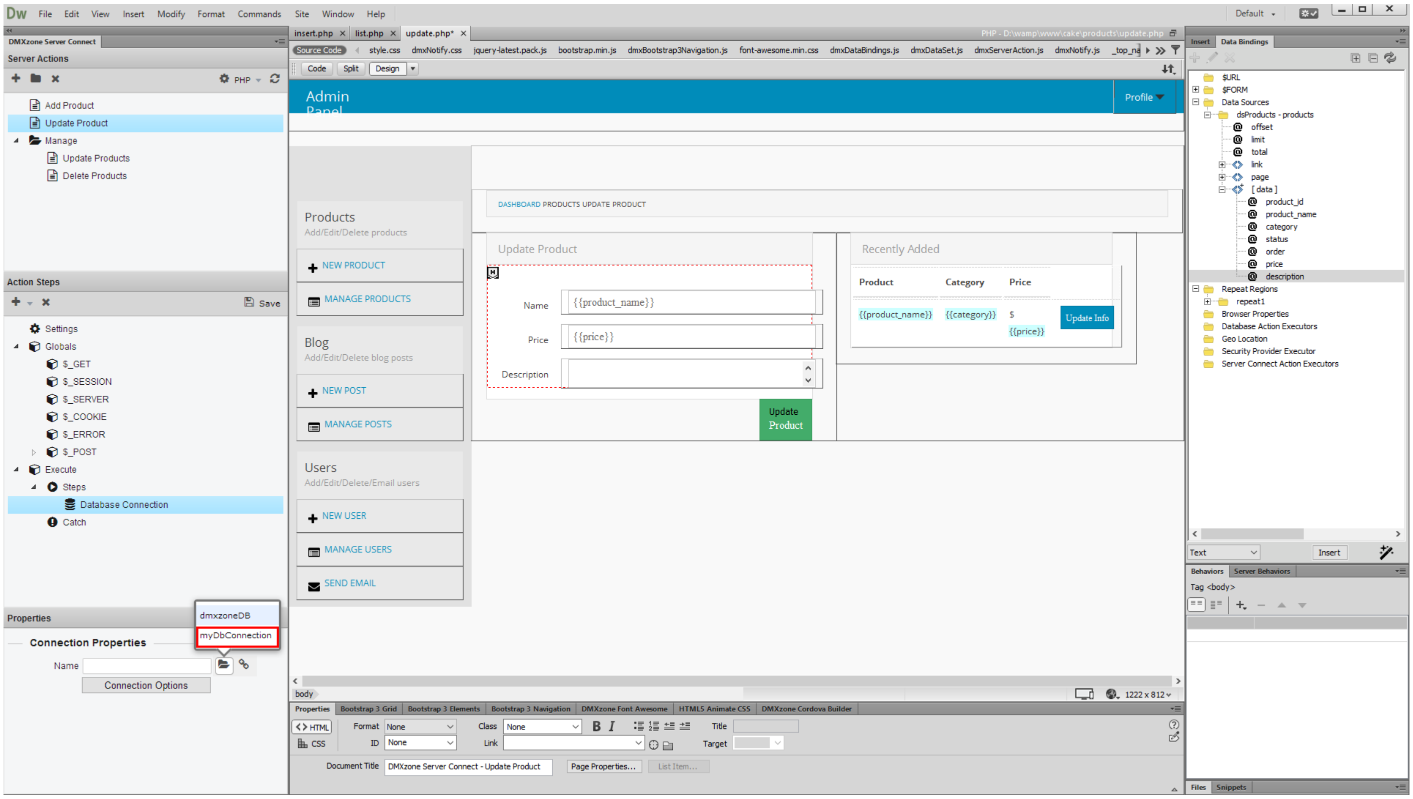
Form form1

4 Form Fields Synchronized Successfully!

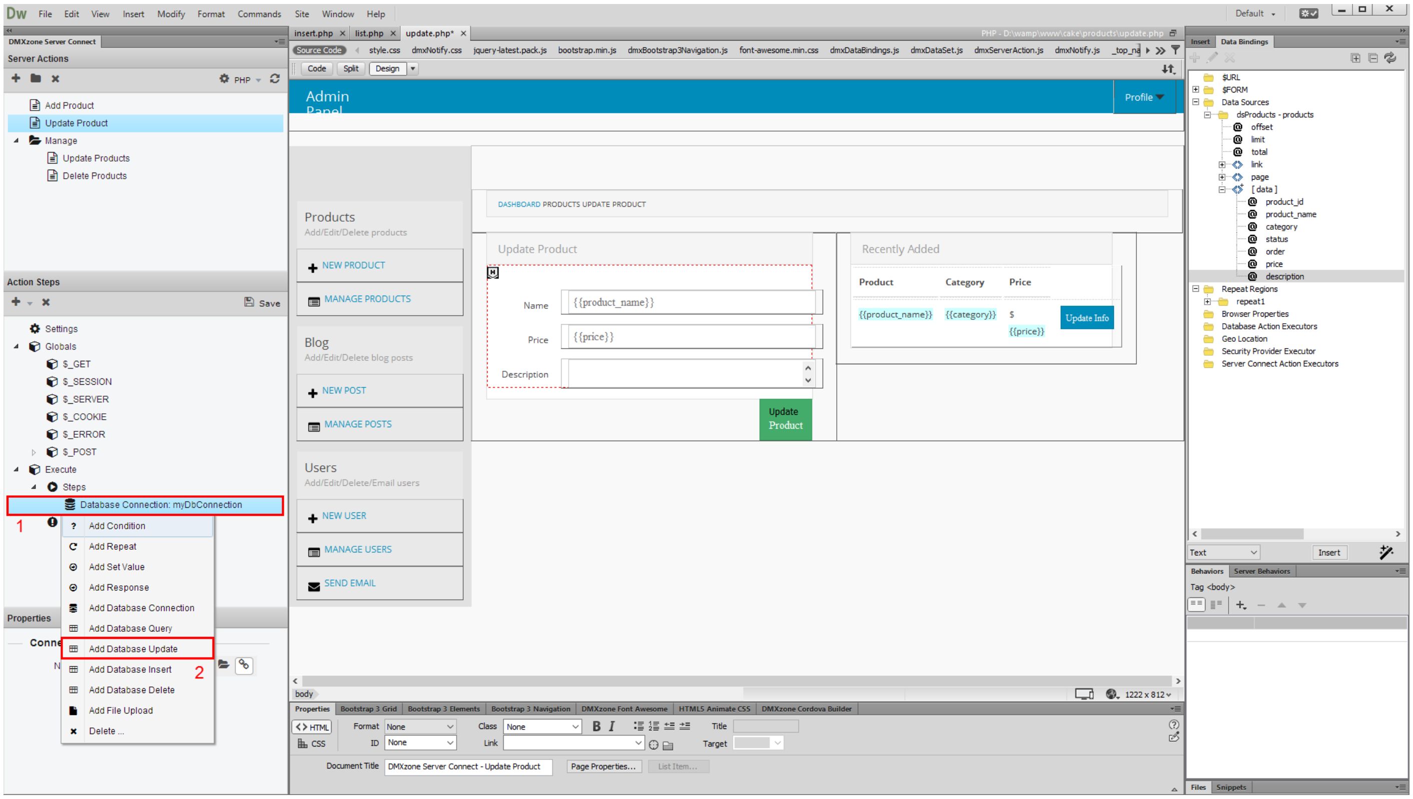
16. We need to add a **database connection**.



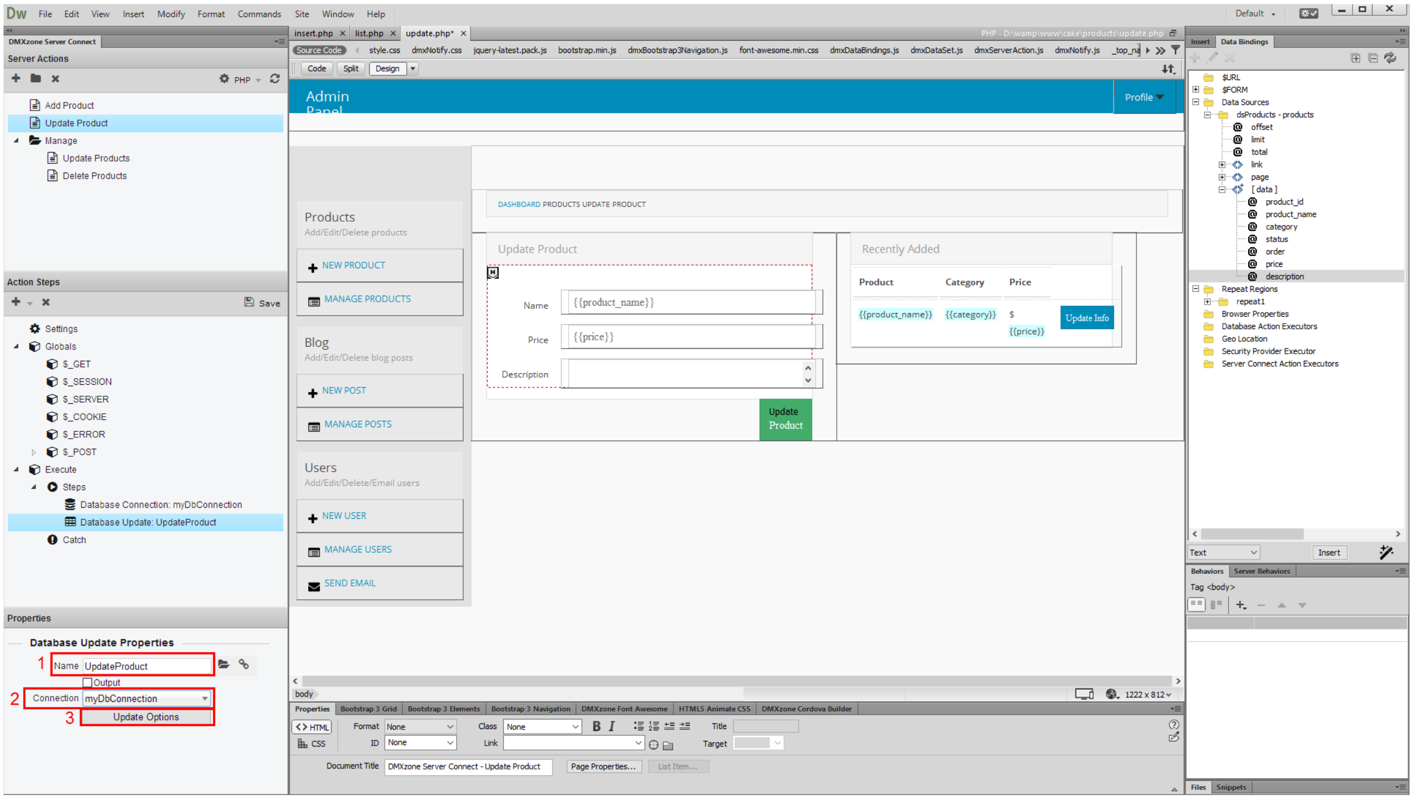
17. We already defined a database connection for the repeat region on the page, so we **just select it from the dropdown**.



18. Right click the **database connection** in the steps list **(1)** and select **add database update (2)**.



19. Add **a name for the database update step (1)** and select your **connection** from the dropdown **(2)**. When you're done, click the **update options button (3)**.

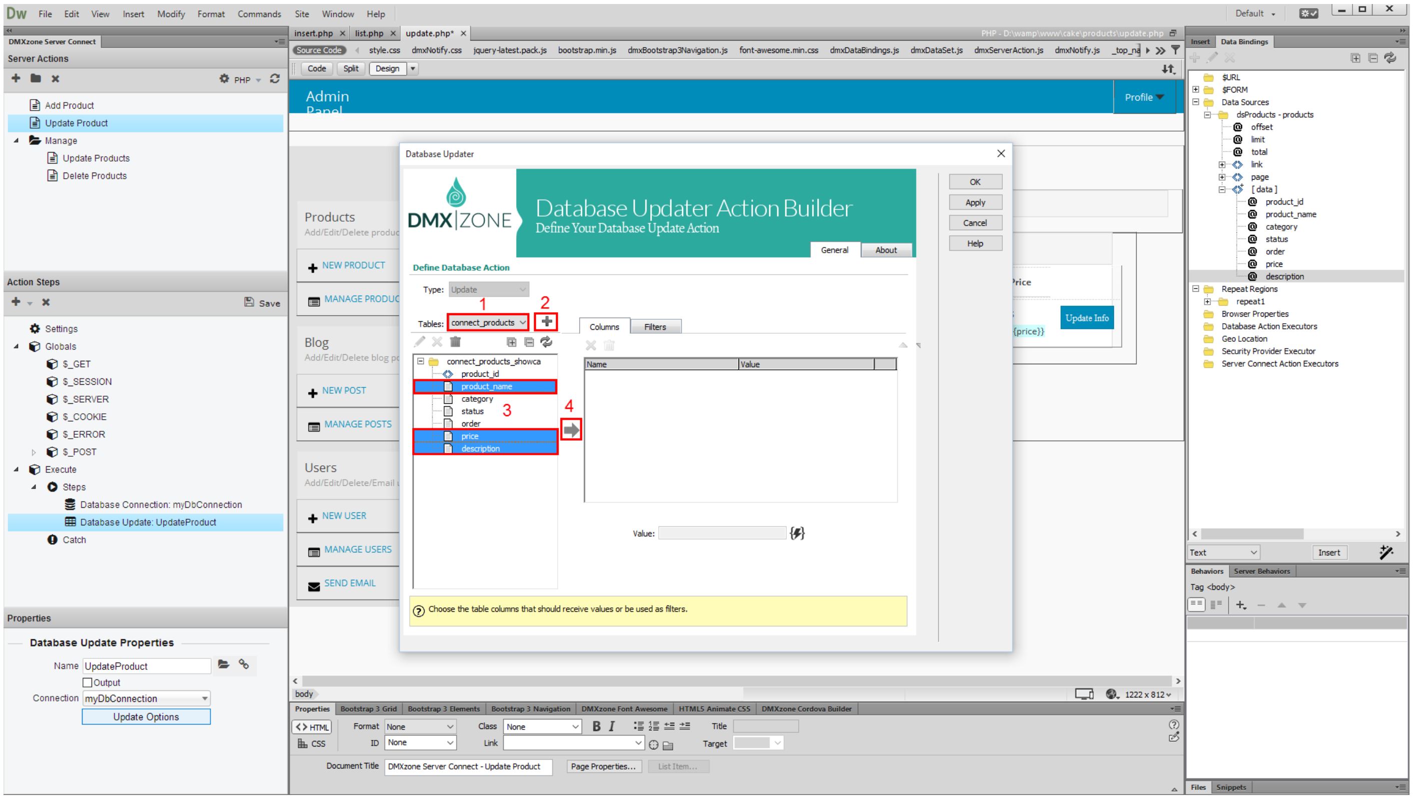


Database Update Properties

Name	UpdateProduct
□ Output	
Connection	myDbConnection

Update Options

20. Select the **database table (1)**, that you want to update and **add it** to the tables list **(2)**. **Choose (3)** and **add the table columns** that you need to edit **(4)**.



Database Updater

Database Updater Action Builder
Define Your Database Update Action

Define Database Action

Type: Update

Tables: connect_products

connect_products_showca
product_id
product_name
category
status
order
price
description

Name | Value

Value:

Choose the table columns that should receive values or be used as filters.

Database Update Properties

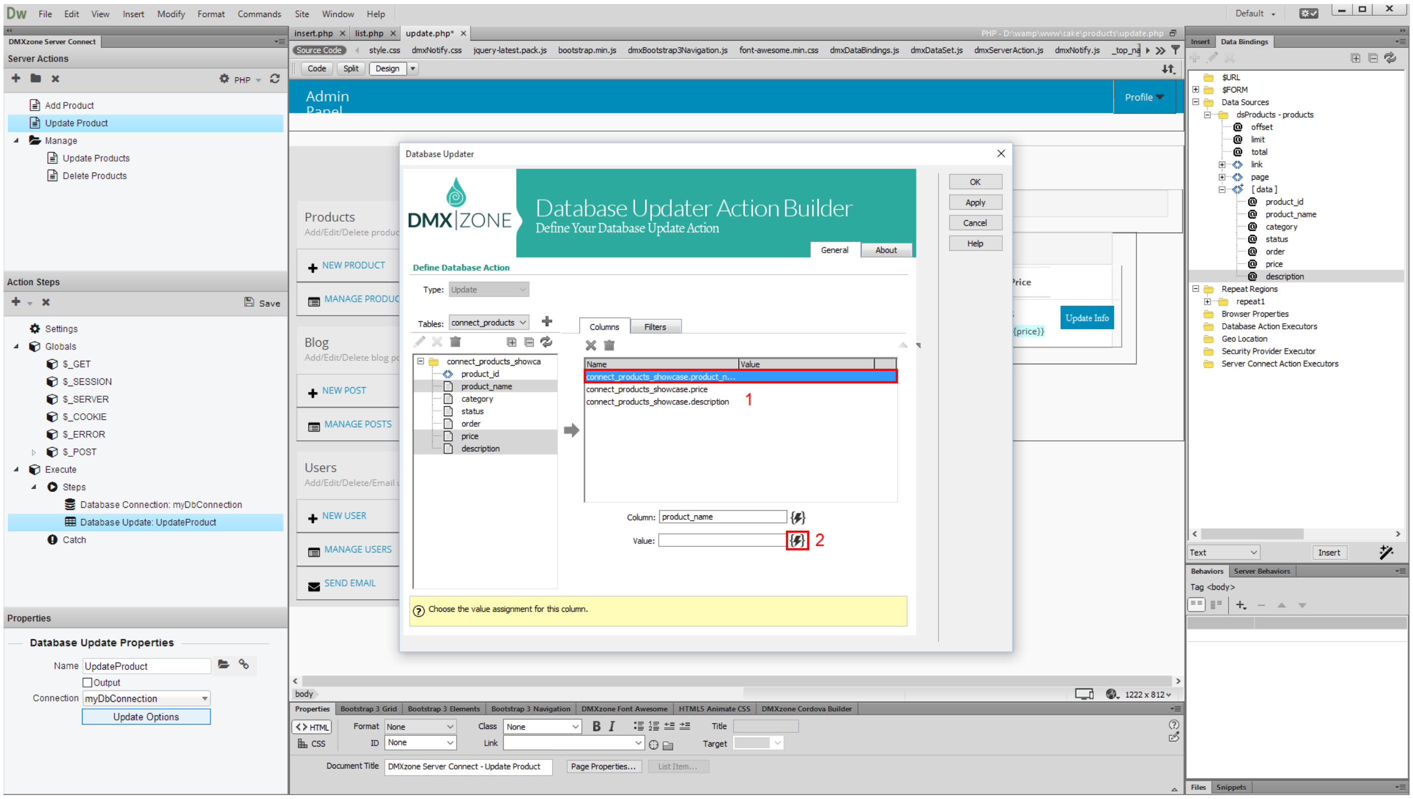
Name UpdateProduct

□ Output

Connection myDbConnection

Update Options

21. Now, bind the corresponding form inputs to each of them. Select the **first column (1)** and click the **server data bindings picker (2)**.



Database Updater

Database Updater Action Builder
Define Your Database Update Action

Define Database Action

Type: Update

Tables: connect_products

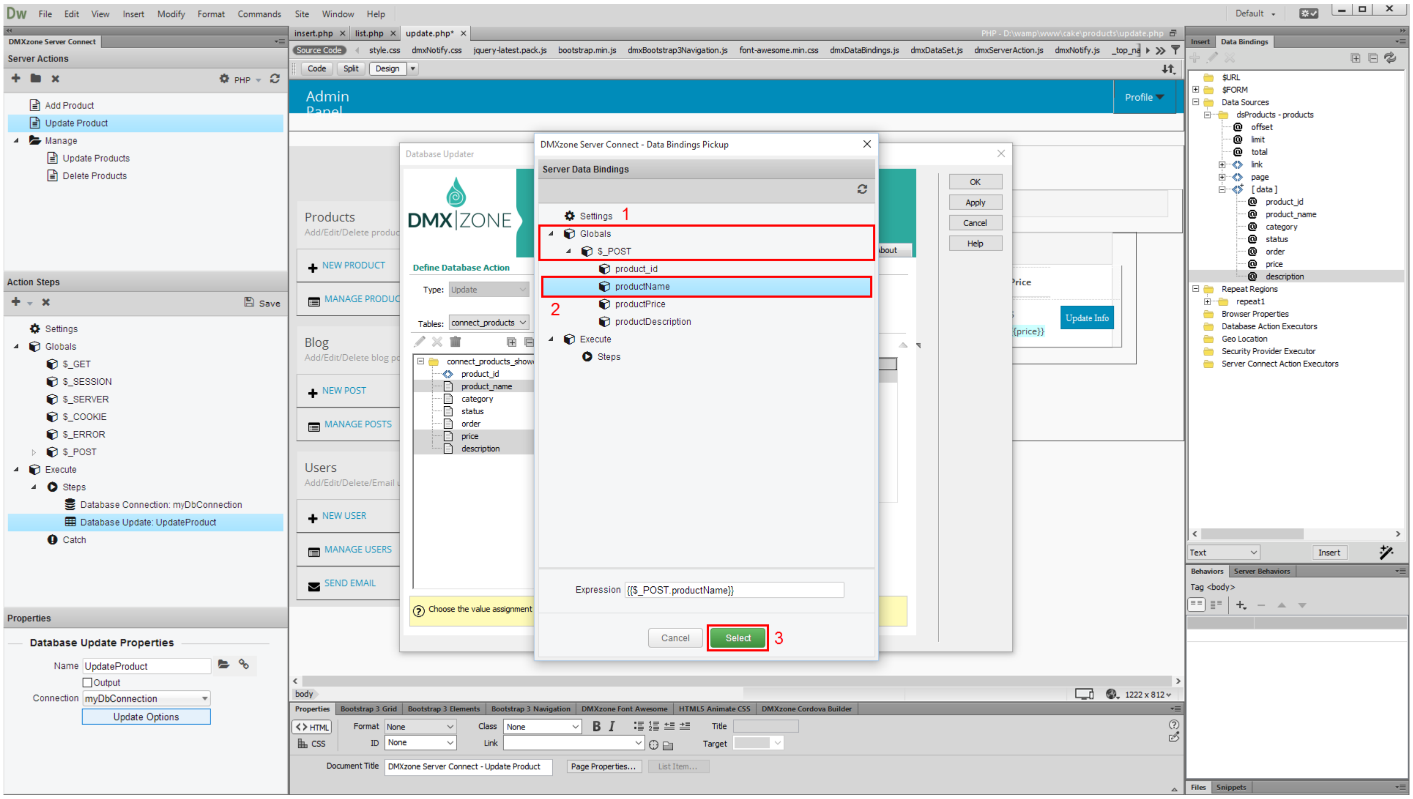
Name	Value
connect_products_showcase.product_n...	
connect_products_showcase.price	
connect_products_showcase.description	

Column: product_name

Value:

Choose the value assignment for this column.

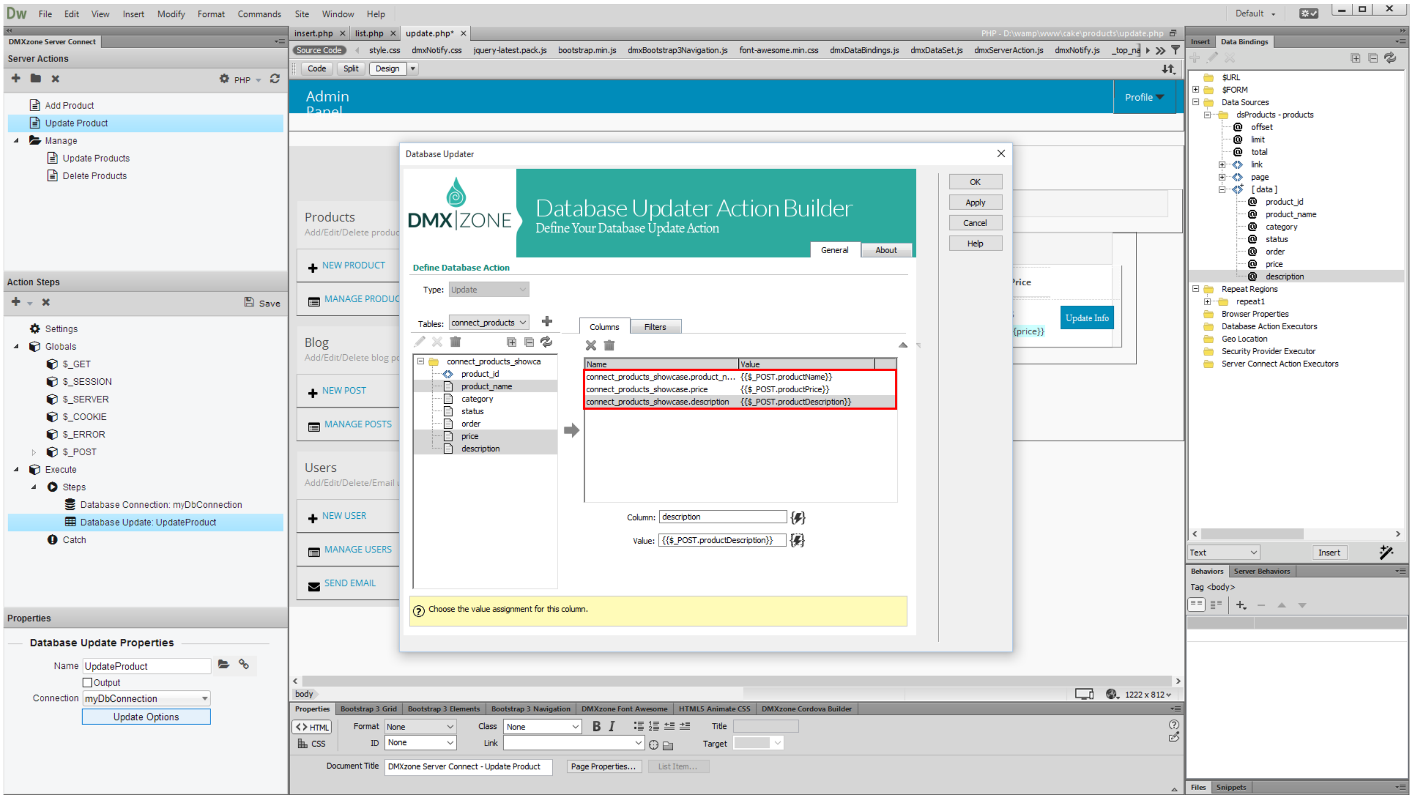
22. Under globals, **expand post variables (1)**, select the **form input** that corresponds to the selected database column **(2)** and click the **select button (3)**.



Expression {{$_POST.productName}}

Name UpdateProduct
Connection myDbConnection

23. We do the same with the rest of the database table columns.



Database Updater

Database Updater Action Builder
Define Your Database Update Action

Define Database Action

Type: Update

Tables: connect_products

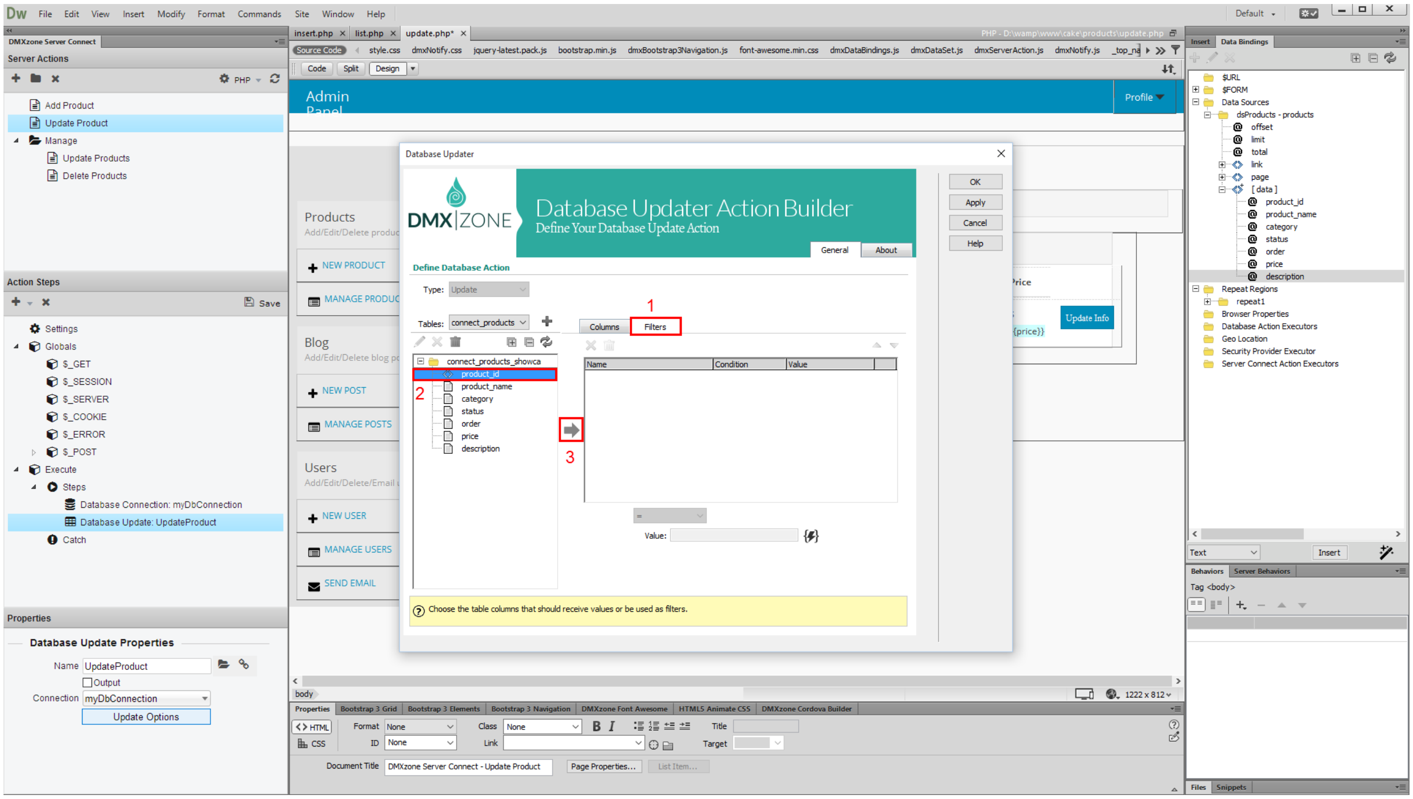
Name	Value
connect_products_showcase.product_n...	{{$_POST.productName}}
connect_products_showcase.price	{{$_POST.productPrice}}
connect_products_showcase.description	{{$_POST.productDescription}}

Column: description

Value: {{$_POST.productDescription}}

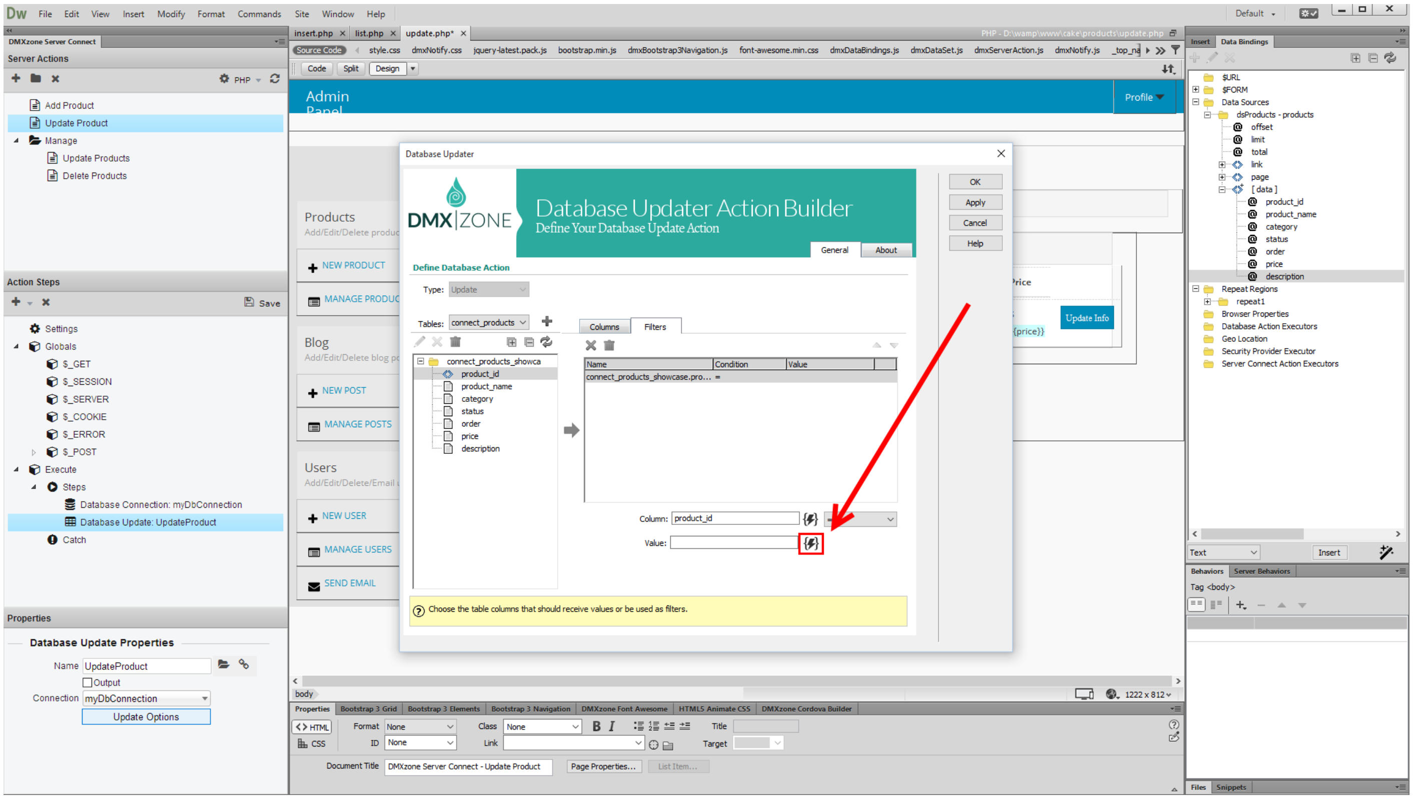
Choose the value assignment for this column.

24. In the **filter tab (1)**, select the **column (2)**, that you want to filter your product by. In our tutorial this is the product ID. Click the **add button (3)**.



Choose the table columns that should receive values or be used as filters.

25. The column value must be equal to the value of the hidden field, located in the detail region. Click the **server data bindings picker**.



Type: Update

Tables: connect_products

connect_products_showcase.pro... =

Column: product_id

Value:

Choose the table columns that should receive values or be used as filters.

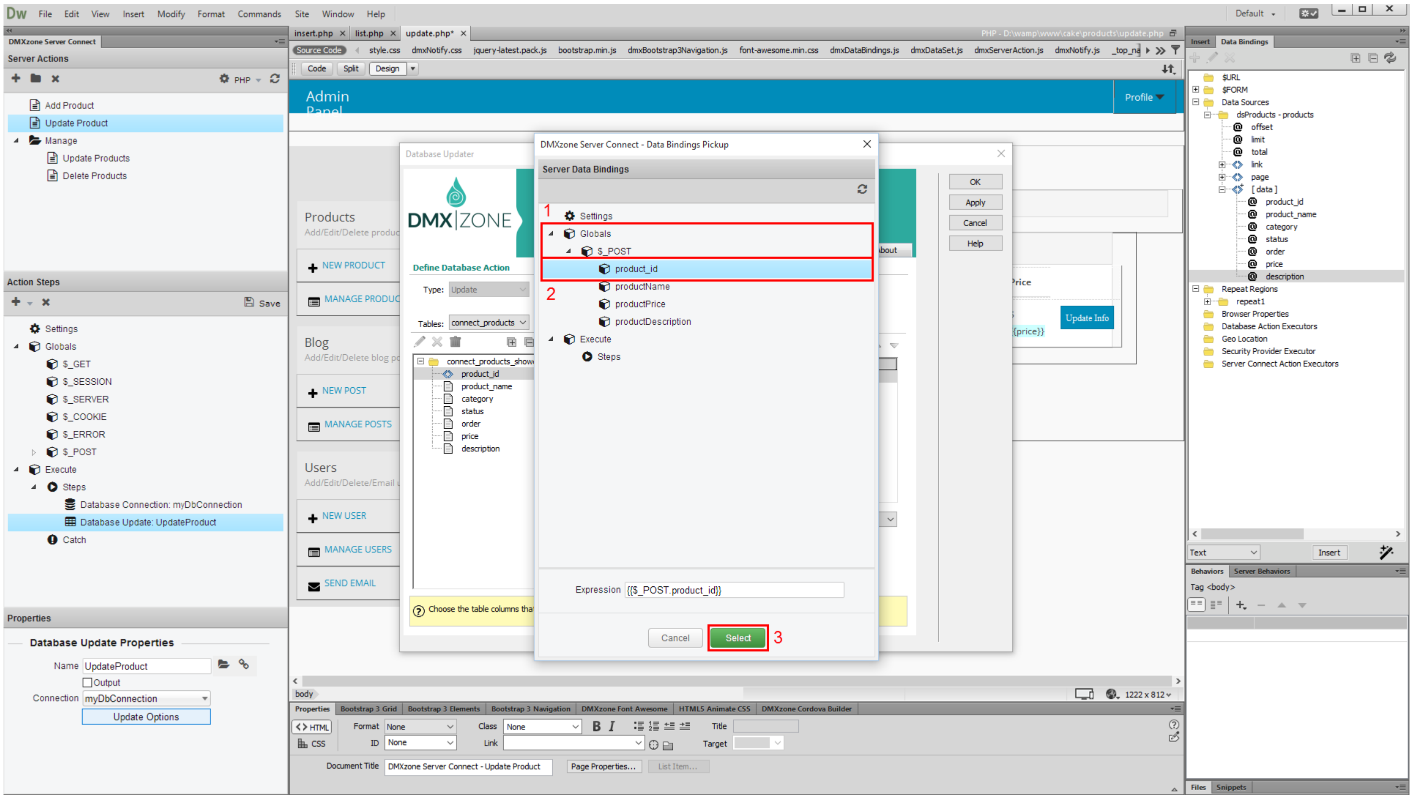
Database Update Properties

Name: UpdateProduct

Connection: myDbConnection

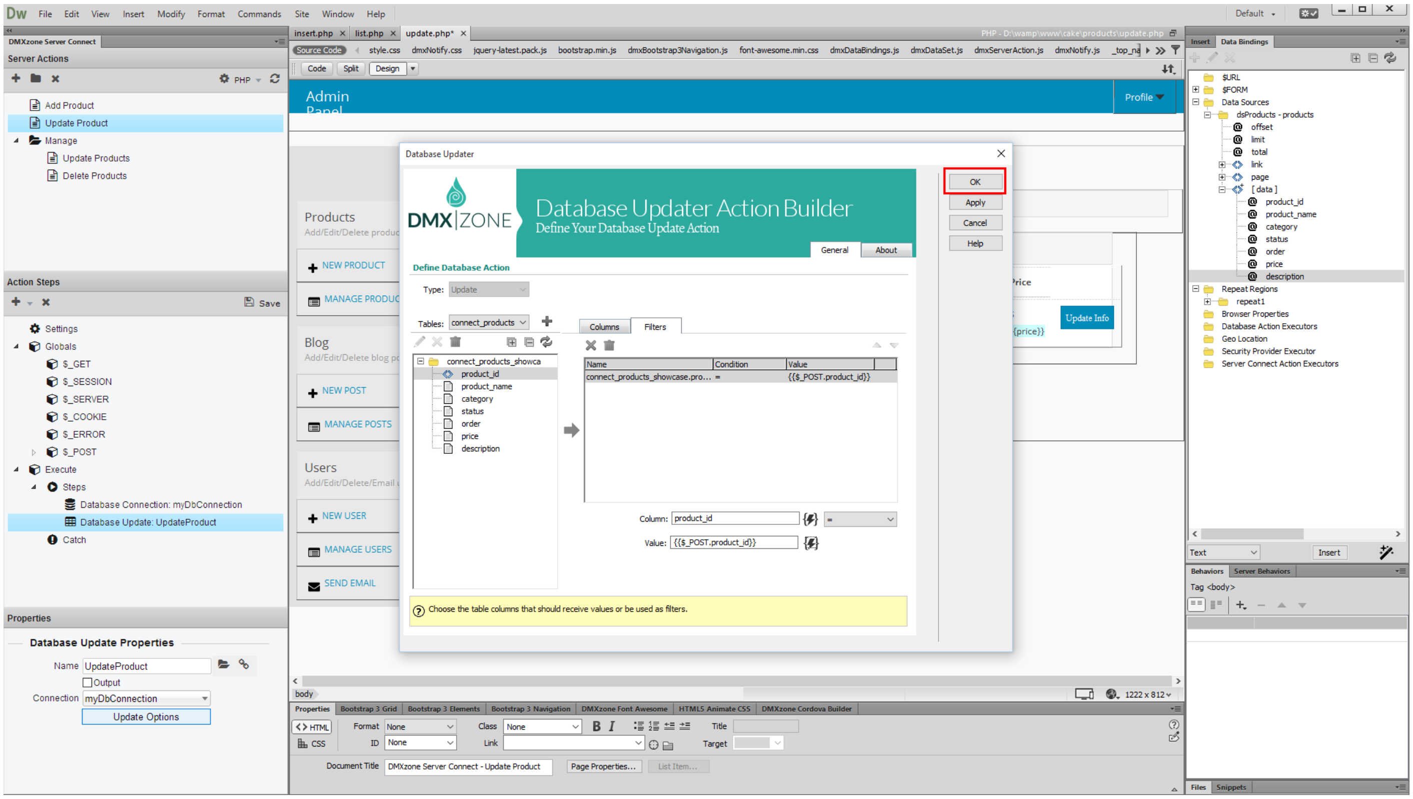
Update Options

26. Expand the global input variables and **under post (1)**, select the **hidden field name (2)**. When you're done, click the **select button (3)**.



Expression {{$_POST.product_id}}

27. Now **click OK** in order to apply the changes.



Database Updater

Database Updater Action Builder
Define Your Database Update Action

Define Database Action

Type: Update

Tables: connect_products

Name	Condition	Value
connect_products_showcase.pro...	=	{{$_POST.product_id}}

Column: product_id =

Value: {{$_POST.product_id}}

Choose the table columns that should receive values or be used as filters.

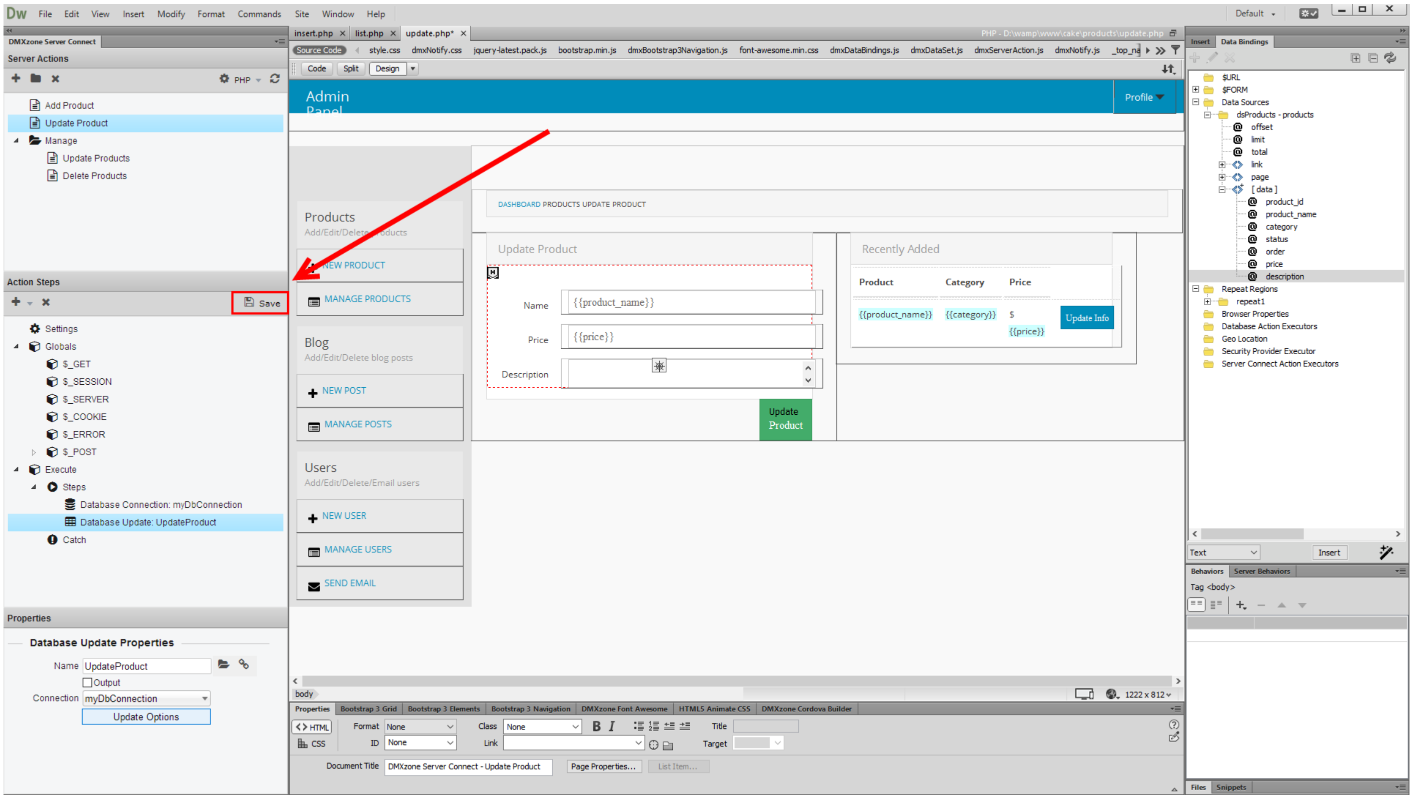
Database Update Properties

Name: UpdateProduct

☐ Output

Connection: myDbConnection

28. Also, make sure to **save your action steps**.



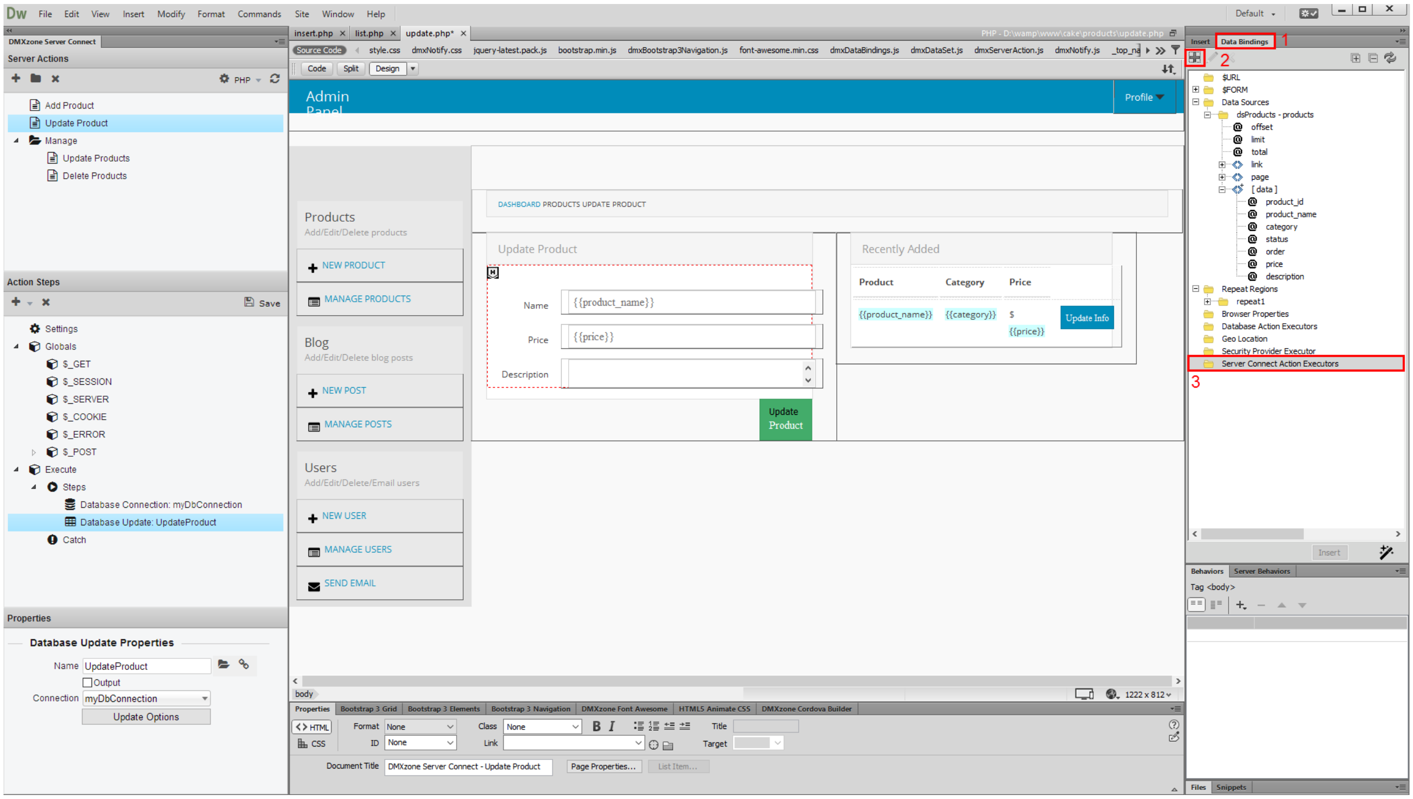
Database Update Properties

Name UpdateProduct

□ Output

Connection myDbConnection

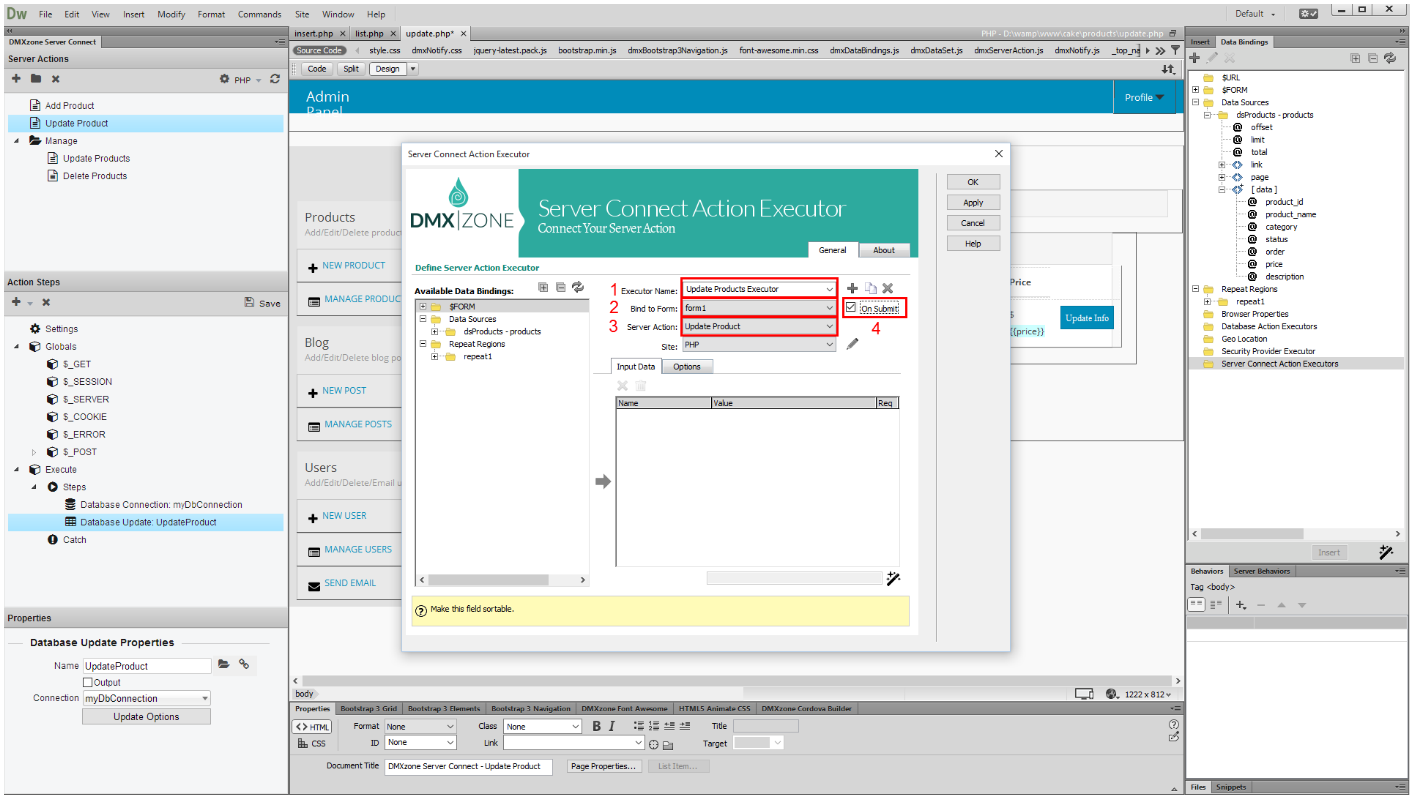
29. Then, in the **HTML5 Data Bindings panel (1)**, select the **server connect action executors (2)** and click **add new button (3)**.



Database Update Properties
Name UpdateProduct
□ Output
Connection myDbConnection
Update Options

Document Title DMXzone Server Connect - Update Product

30. Select your **server action** from the dropdown **(1)** and **bind it to the update form** on your page (2). Don't forget to **add a name for the executor (3)** and to **check the on submit checkbox (4)**.



Executor Name: Update Products Executor

Bind to Form: form1

Server Action: Update Product

Site: PHP

☑ On Submit

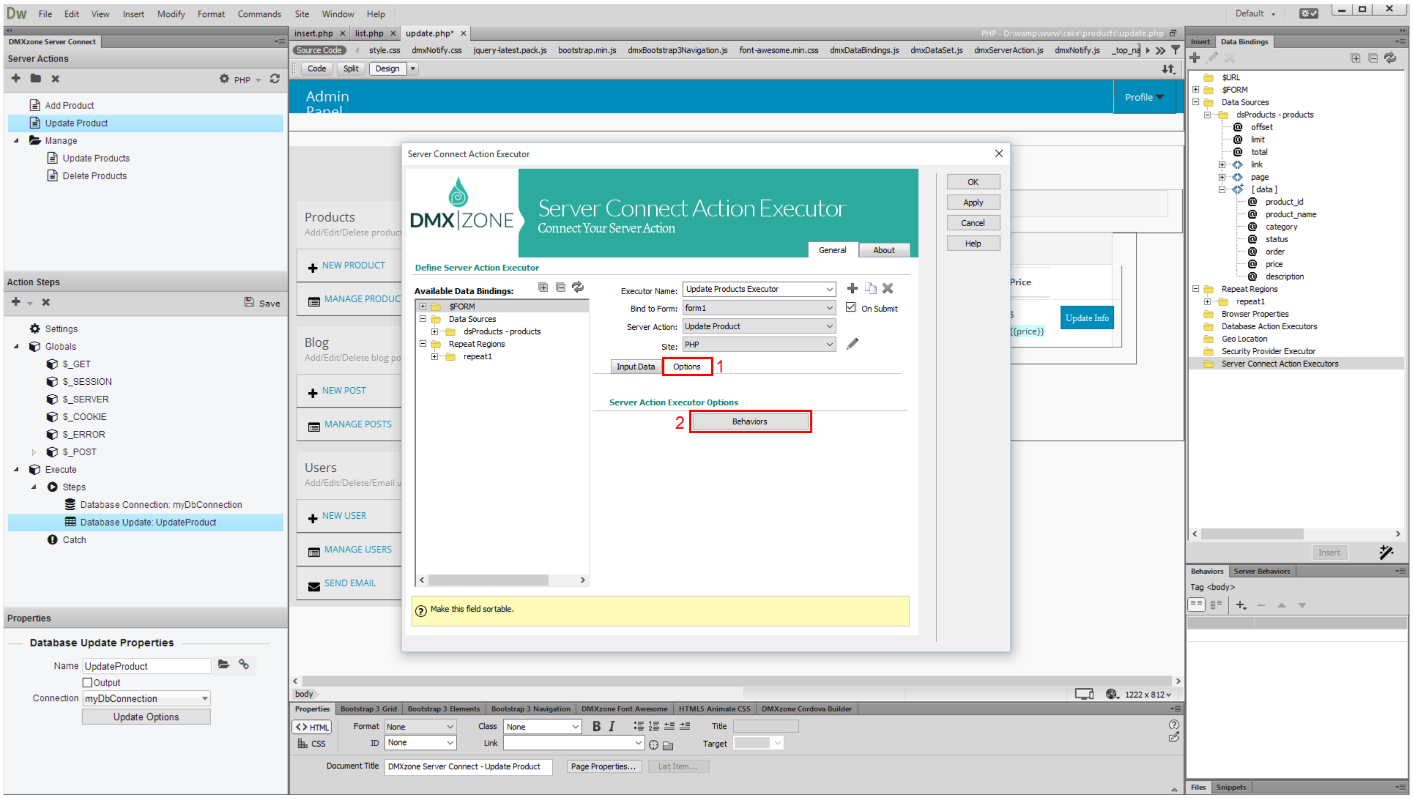
Name: UpdateProduct

□ Output

Connection: myDbConnection

Document Title: DMXzone Server Connect - Update Product

31. We'd like to see the changes in the products list immediately, so we select **options (1)** and **click the behaviors (2)**.



Server Connect Action Executor

Define Server Action Executor

Executor Name: Update Products Executor

Bind to Form: form1

☑ On Submit

Server Action: Update Product

Site: PHP

Server Action Executor Options

Make this field sortable.

Database Update Properties

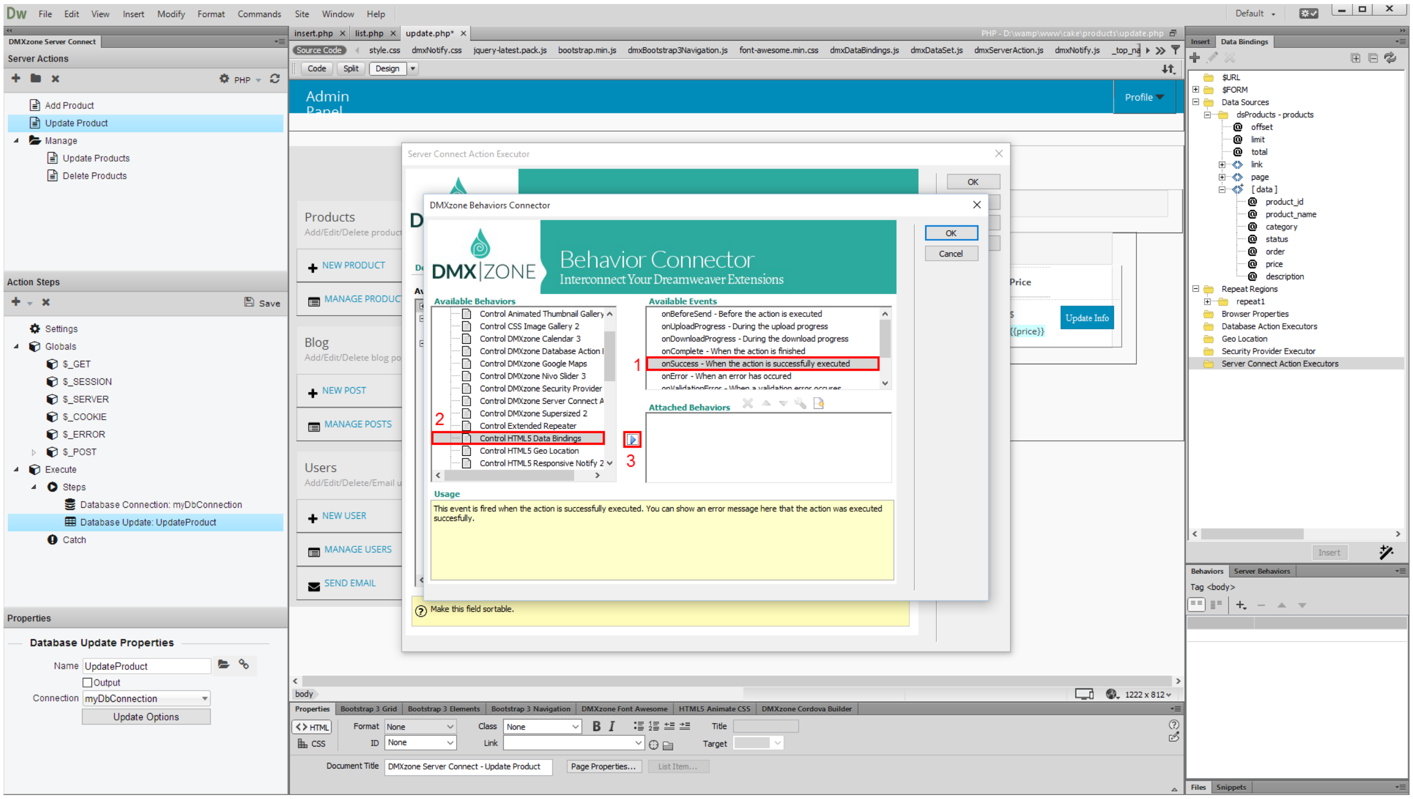
Name: UpdateProduct

□ Output

Connection: myDbConnection

Document Title: DMXzone Server Connect - Update Product

32. In the behavior connector, we select the **on success event (1)**. And then, under DMXzone, click **control HTML5 Data Bindings Behavior (2)** and **add it (3)**.



Server Connect Action Executor
DMXzone Behaviors Connector
Behavior Connector
Interconnect Your Dreamweaver Extensions
Available Behaviors
Available Events
Attached Behaviors
1
2
3
Control HTML5 Data Bindings
onSuccess - When the action is successfully executed
OK
Cancel

Usage

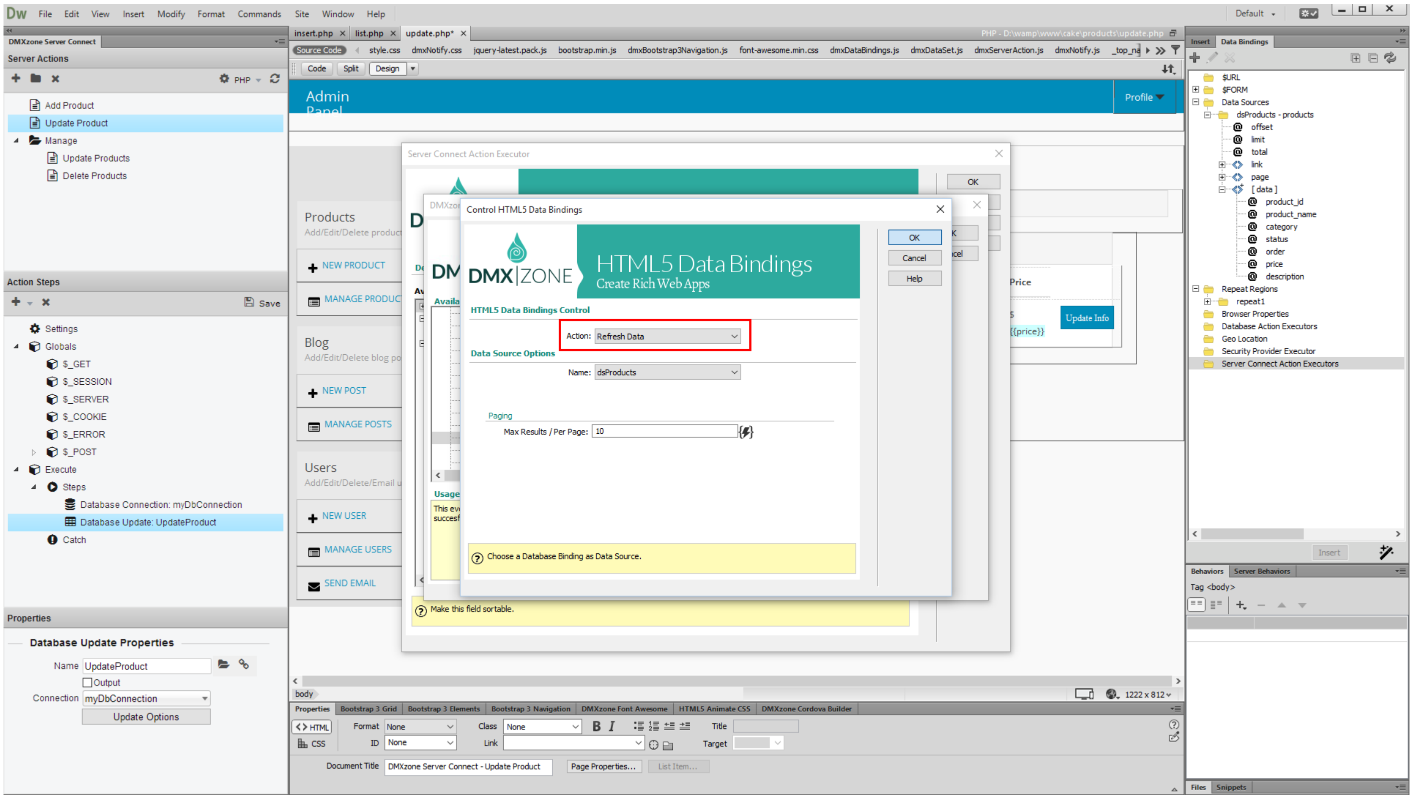
This event is fired when the action is successfully executed. You can show an error message here that the action was executed succesfully.

Database Update Properties

Name: UpdateProduct

Connection: myDbConnection

33. Now, set the action to – **refresh data** and click OK.



Control HTML5 Data Bindings

HTML5 Data Bindings Control

Action: Refresh Data

Data Source Options

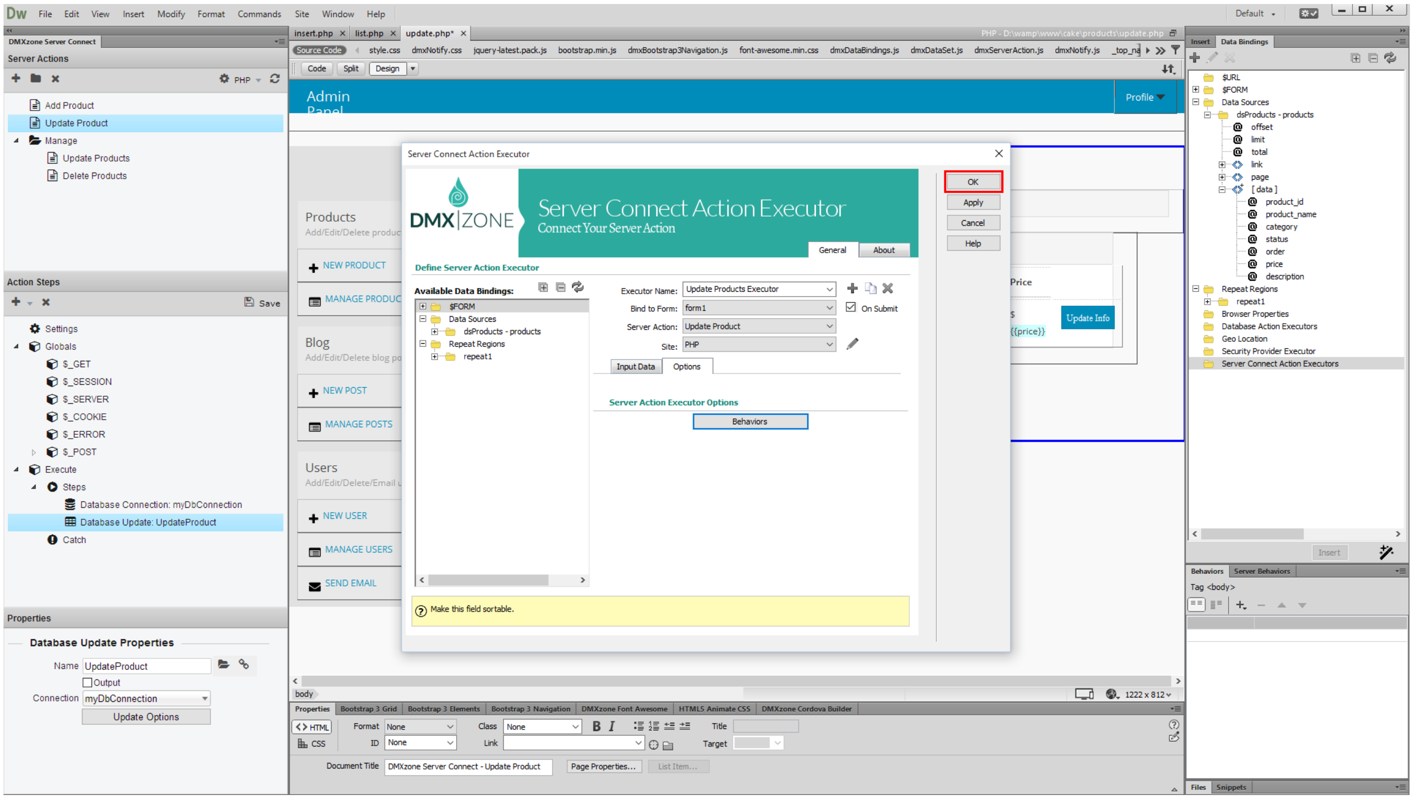
Name: dsProducts

Paging

Max Results / Per Page: 10

Choose a Database Binding as Data Source.

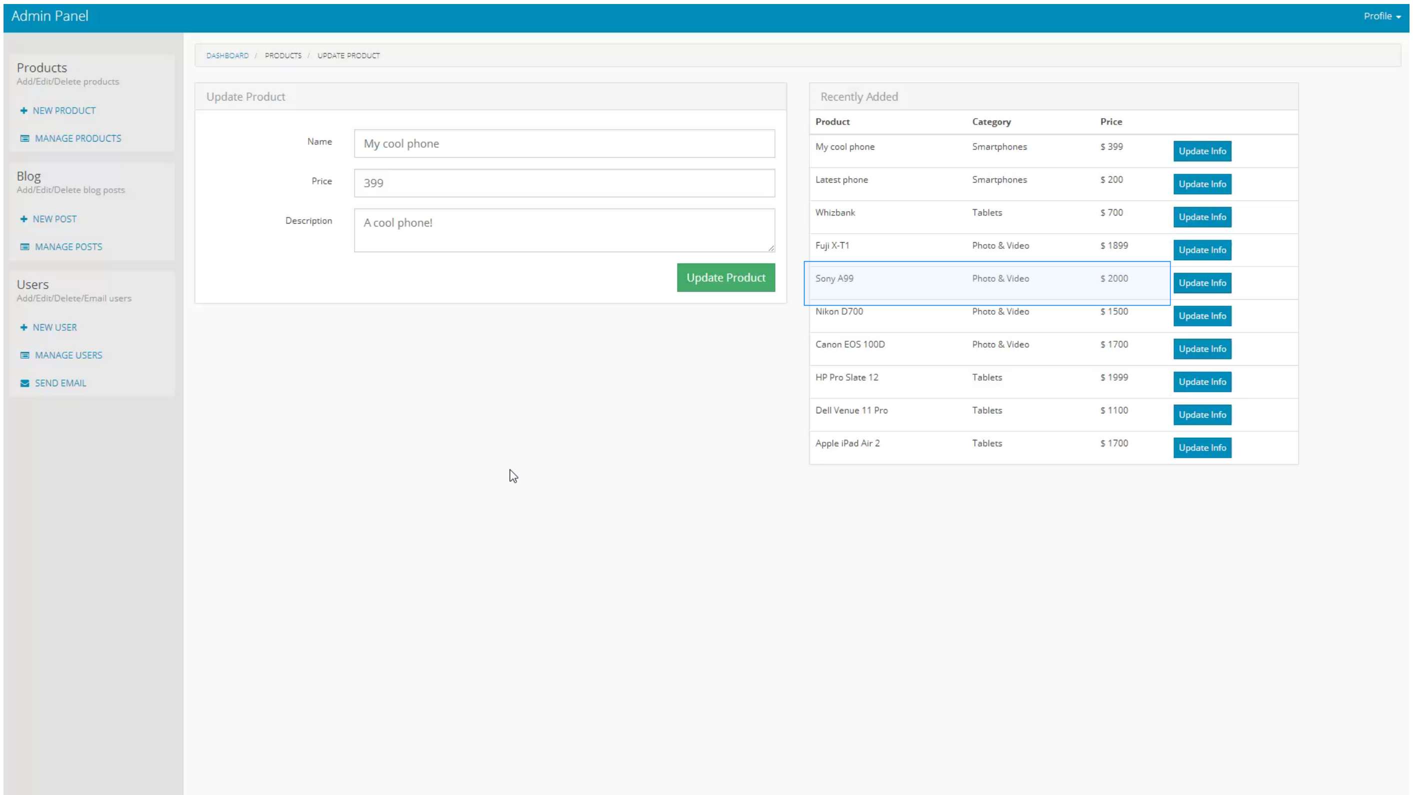
34. Click OK again.


Server Connect Action Executor
Server Connect Action Executor
Connect Your Server Action
General
About
OK
Apply
Cancel
Help
Define Server Action Executor
Available Data Bindings:
$FORM
Data Sources
dsProducts - products
Repeat Regions
repeat1
Input Data
Options
Server Action Executor Options
Behaviors
Make this field sortable.

Executor Name: Update Products Executor
Bind to Form: form1
☑ On Submit
Server Action: Update Product
Site: PHP

Database Update Properties
Name UpdateProduct
□ Output
Connection myDbConnection
Update Options

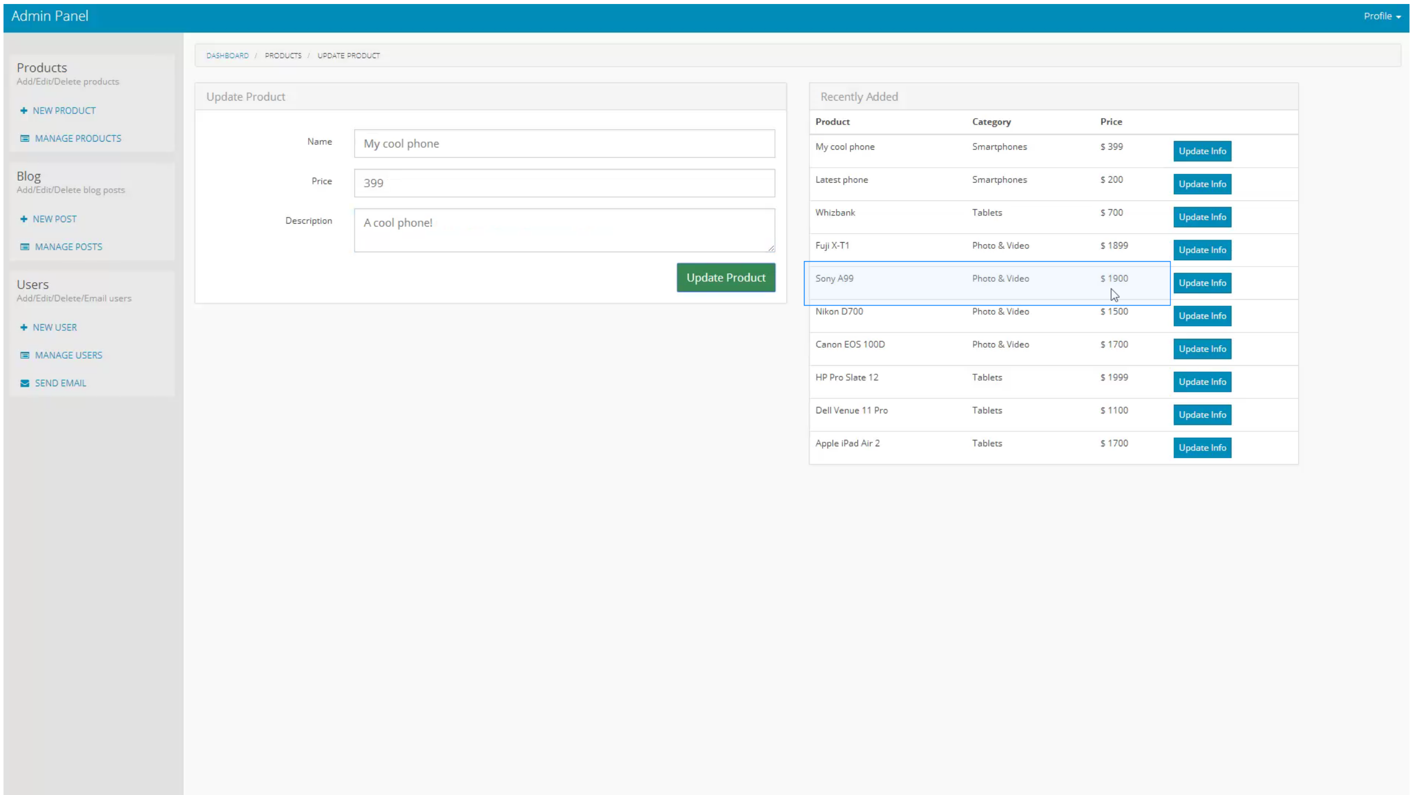
35. Let’s preview the results. You can see that the selected product info appears in the update form.



Name	My cool phone
Price	399
Description	A cool phone!

Product	Category	Price	
My cool phone	Smartphones	$ 399	Update Info
Latest phone	Smartphones	$ 200	Update Info
Whizbank	Tablets	$ 700	Update Info
Fuji X-T1	Photo & Video	$ 1899	Update Info
Sony A99	Photo & Video	$ 2000	Update Info
Nikon D700	Photo & Video	$ 1500	Update Info
Canon EOS 100D	Photo & Video	$ 1700	Update Info
HP Pro Slate 12	Tablets	$ 1999	Update Info
Dell Venue 11 Pro	Tablets	$ 1100	Update Info
Apple iPad Air 2	Tablets	$ 1700	Update Info

36. As soon as you make your changes and click the update product button, the products list refreshes and you can see the changes there.



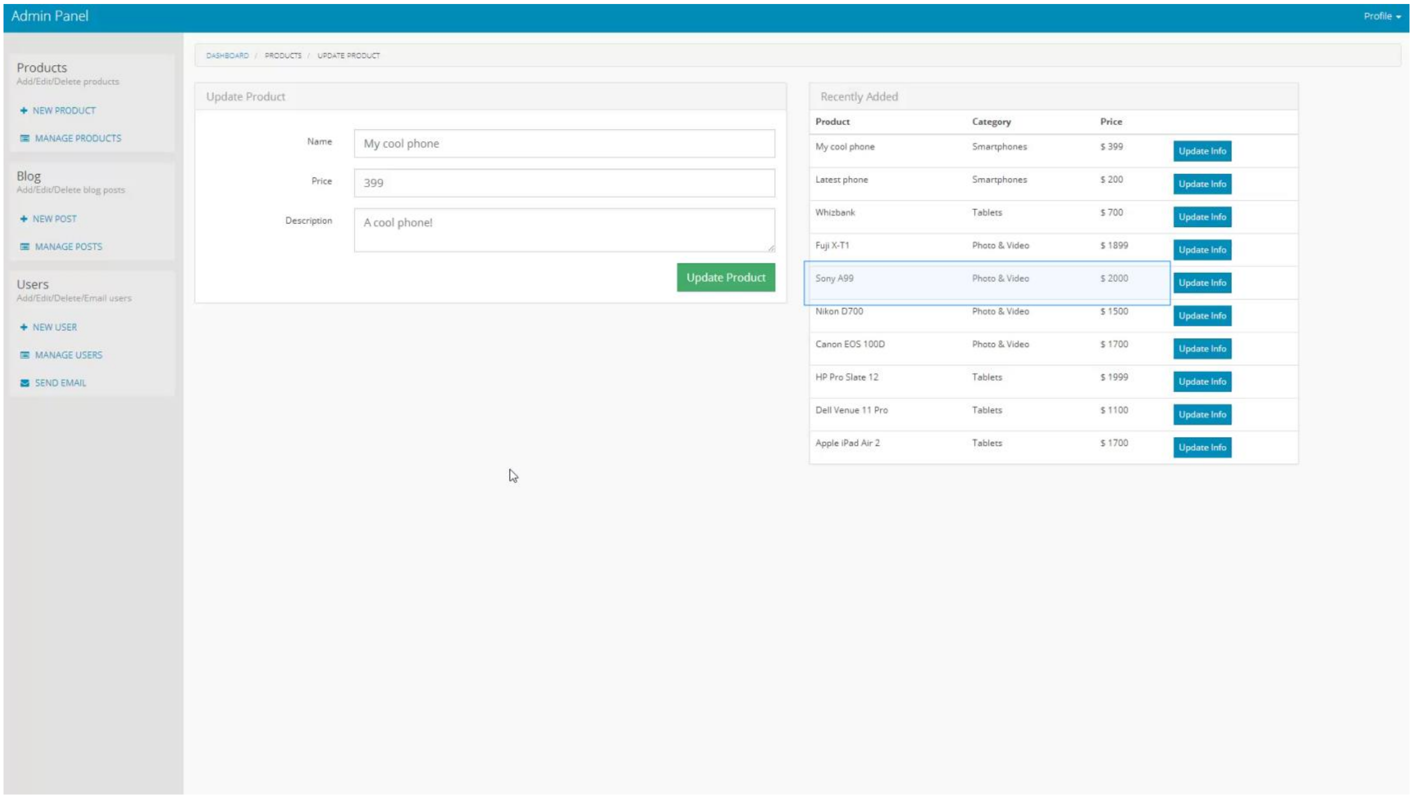
Name	My cool phone
Price	399
Description	A cool phone!

Product	Category	Price	
My cool phone	Smartphones	$ 399	Update Info
Latest phone	Smartphones	$ 200	Update Info
Whizbank	Tablets	$ 700	Update Info
Fuji X-T1	Photo & Video	$ 1899	Update Info
Sony A99	Photo & Video	$ 1900	Update Info
Nikon D700	Photo & Video	$ 1500	Update Info
Canon EOS 100D	Photo & Video	$ 1700	Update Info
HP Pro Slate 12	Tablets	$ 1999	Update Info
Dell Venue 11 Pro	Tablets	$ 1100	Update Info
Apple iPad Air 2	Tablets	$ 1700	Update Info

Advanced: Update Multiple Database Records

In this tutorial we will show you how to how to update multiple database records, using DMXzone Server Connect. In order to do that you'll also need HTML5 Data Bindings and DMXzone Database Connector extensions.

*You can also check out the video.

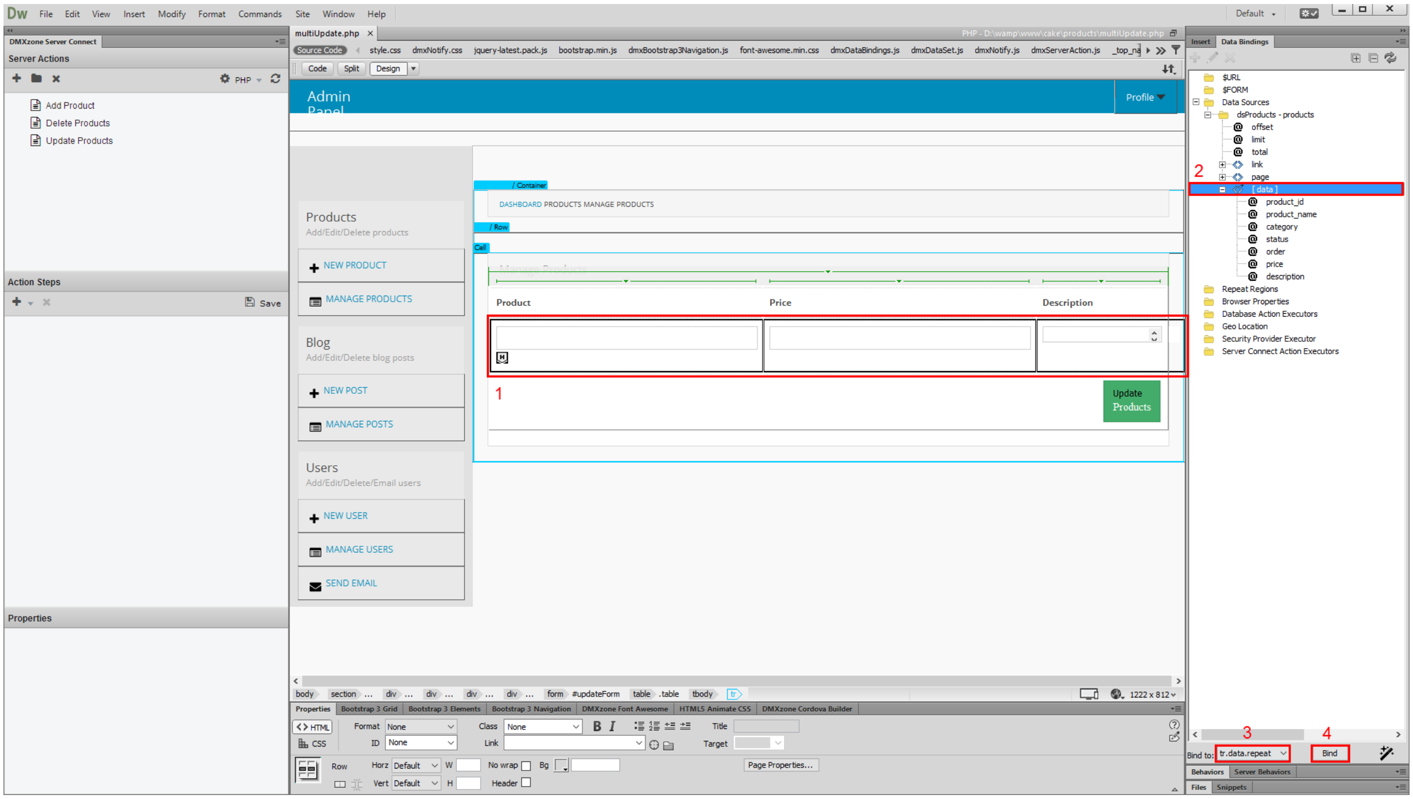


Name: My cool phone
Price: 399
Description: A cool phone!

Product	Category	Price	
My cool phone	Smartphones	$ 399	Update Info
Latest phone	Smartphones	$ 200	Update Info
Whizbank	Tablets	$ 700	Update Info
Fuji X-T1	Photo & Video	$ 1899	Update Info
Sony A99	Photo & Video	$ 2000	Update Info
Nikon D700	Photo & Video	$ 1500	Update Info
Canon EOS 100D	Photo & Video	$ 1700	Update Info
HP Pro Slate 12	Tablets	$ 1999	Update Info
Dell Venue 11 Pro	Tablets	$ 1100	Update Info
Apple iPad Air 2	Tablets	$ 1700	Update Info

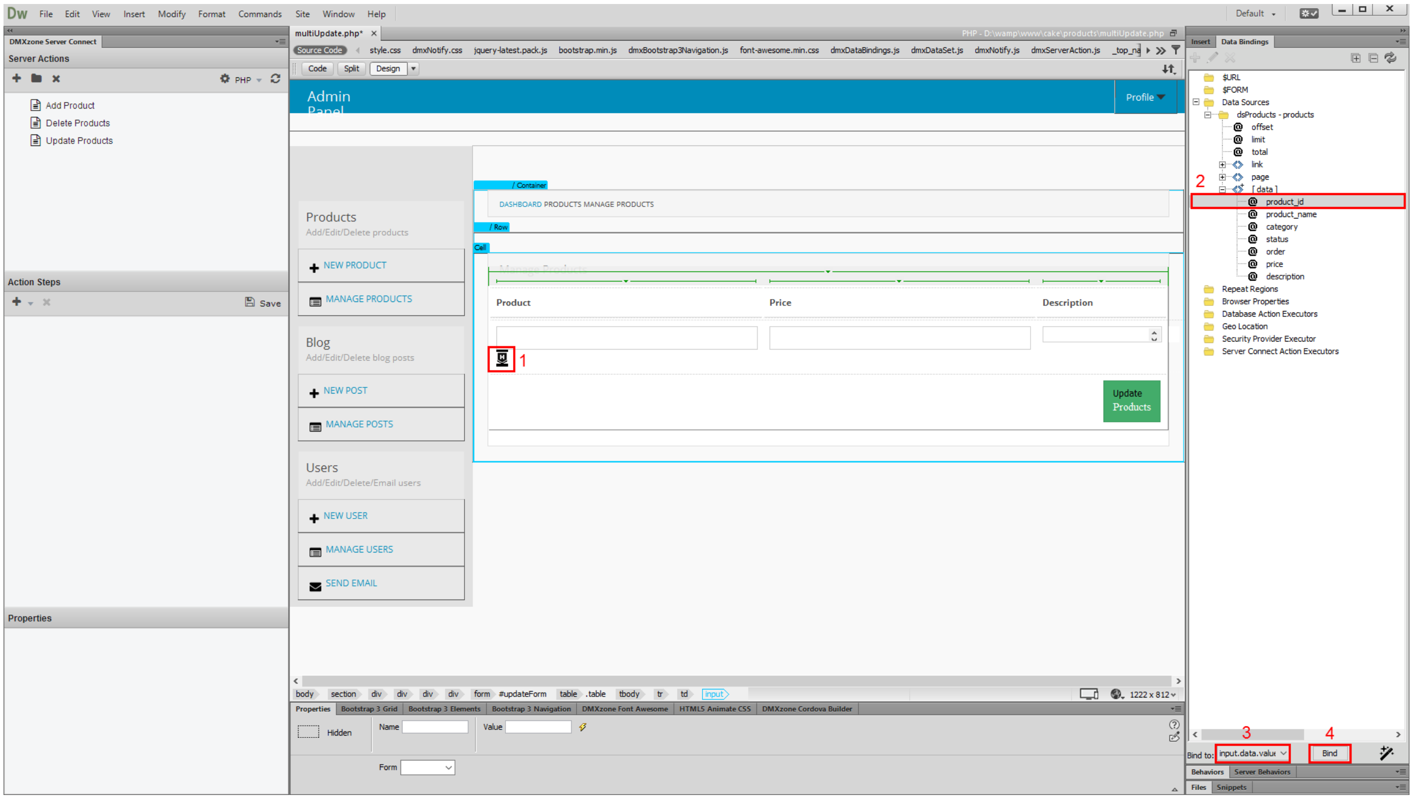
How to do it

1. We created a page, which contains a form with a hidden field and several text fields inside. We will use the hidden field, in order to filter the records that we want to update and we will use the text fields, in order to update the records info. We also created a database source, using the DMXzone Database Connector. It lists our products from the database. First, **select the container (1)** that wraps your form inputs. If there is no such container - then just add one. In this tutorial, our form fields are wrapped inside a table row. Then **click the repeating element (2)** of your data source and **bind (4)** to the **data.repeat.option (3)** in the bind to menu.



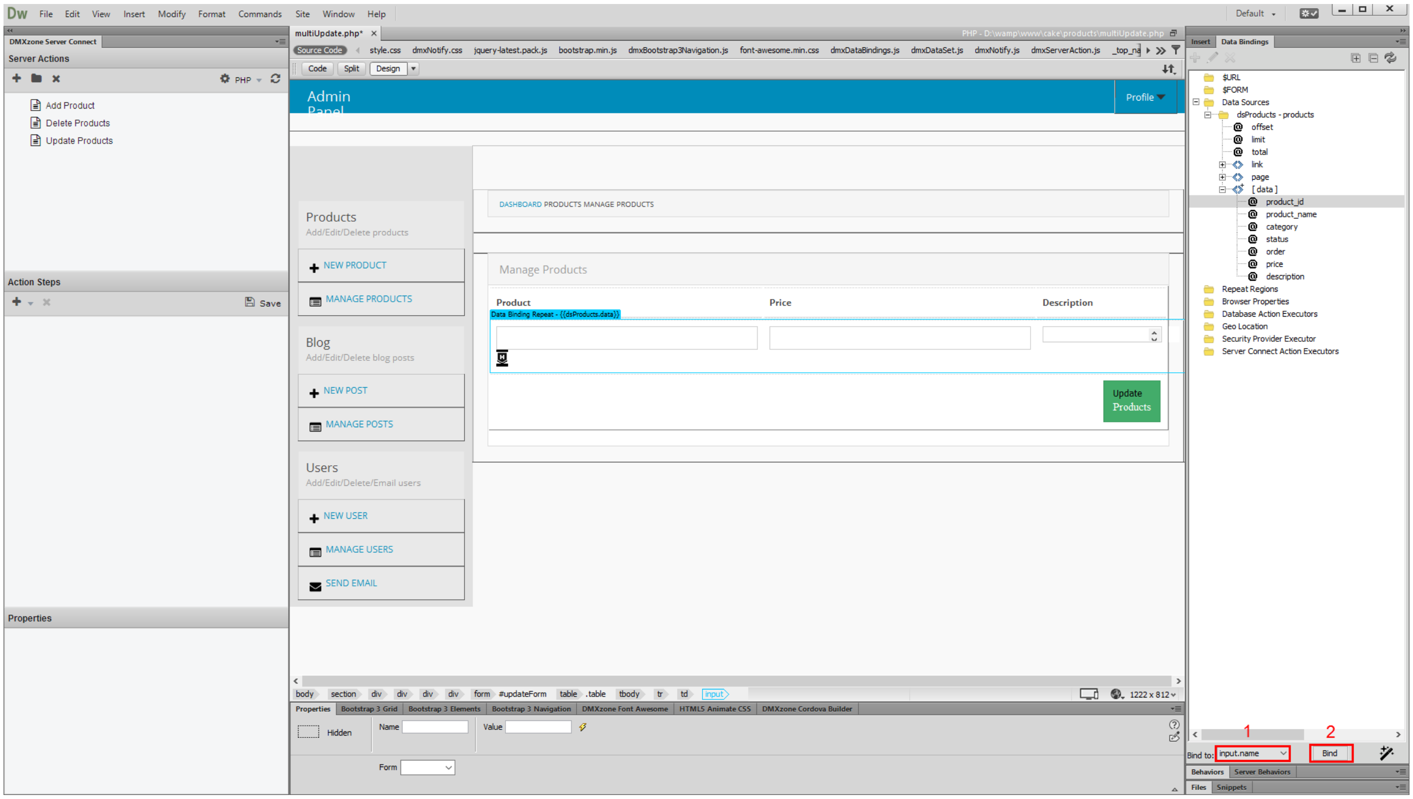
Bind to: tr.data.repeat

2. Then, select the **hidden field (1)**, inside the repeat region that we’ve just created, and select the **ID data binding (2)** from your data source. **Bind (4)** it to the **input.data.value (3)**.

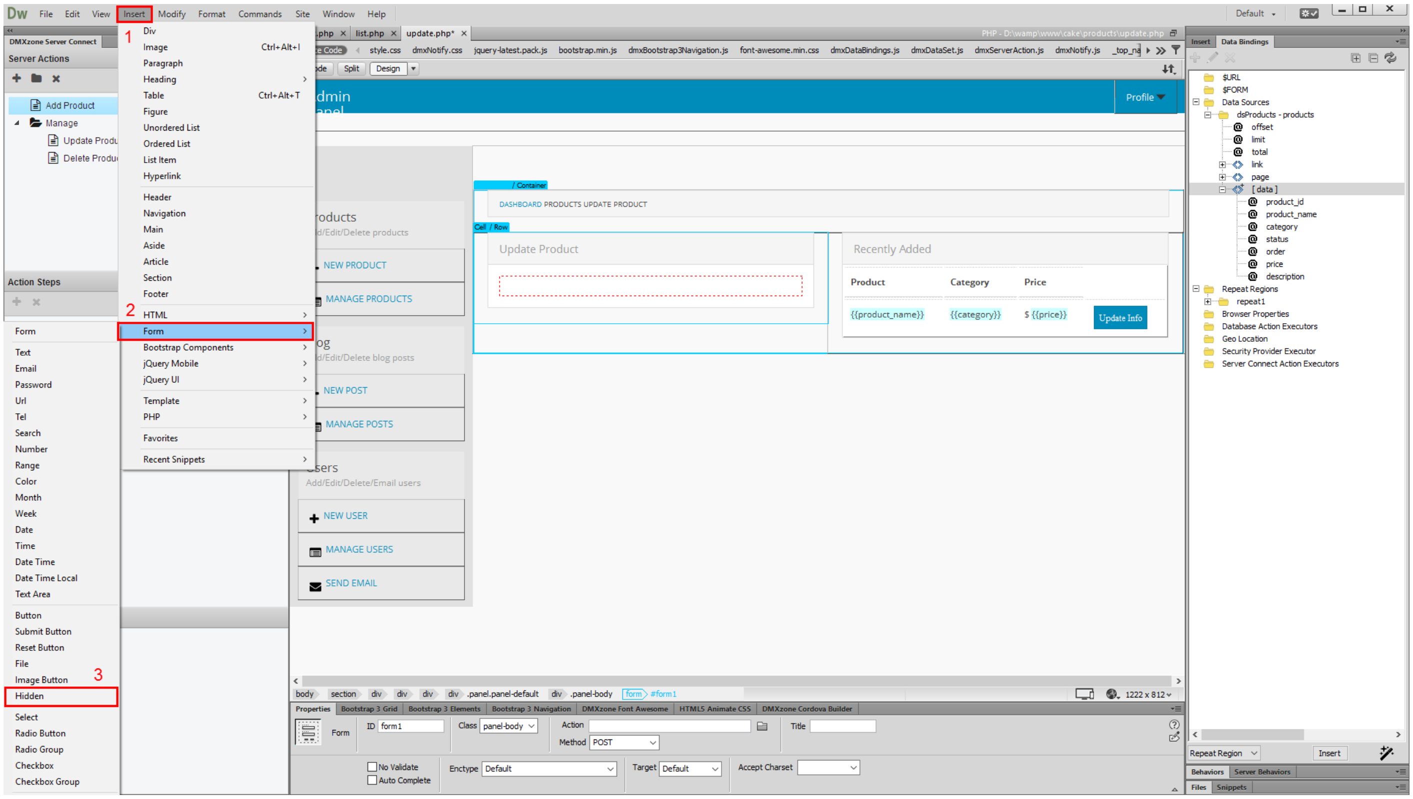


Bind to: input.data.value

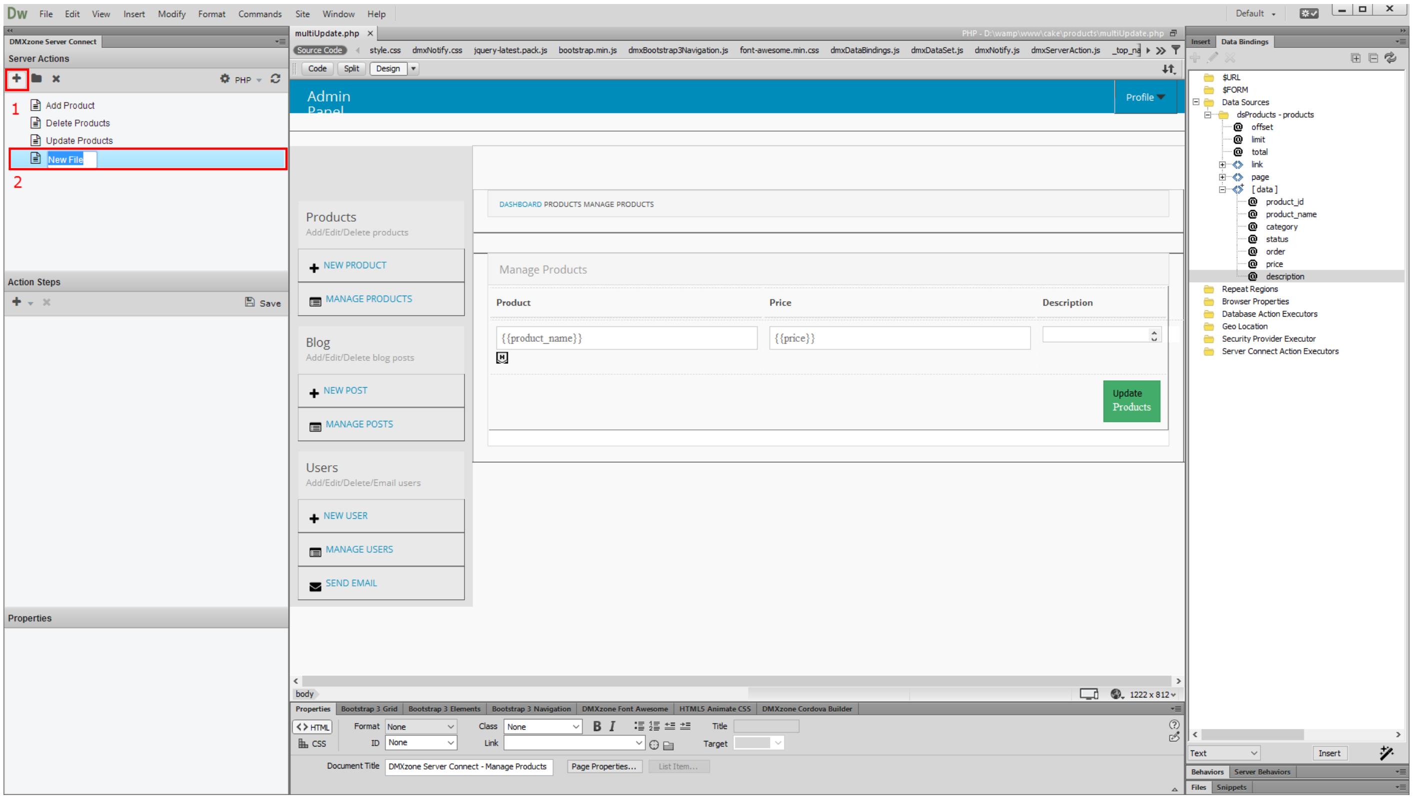
3. And also **bind (2)** it to the **input.name (1)**.



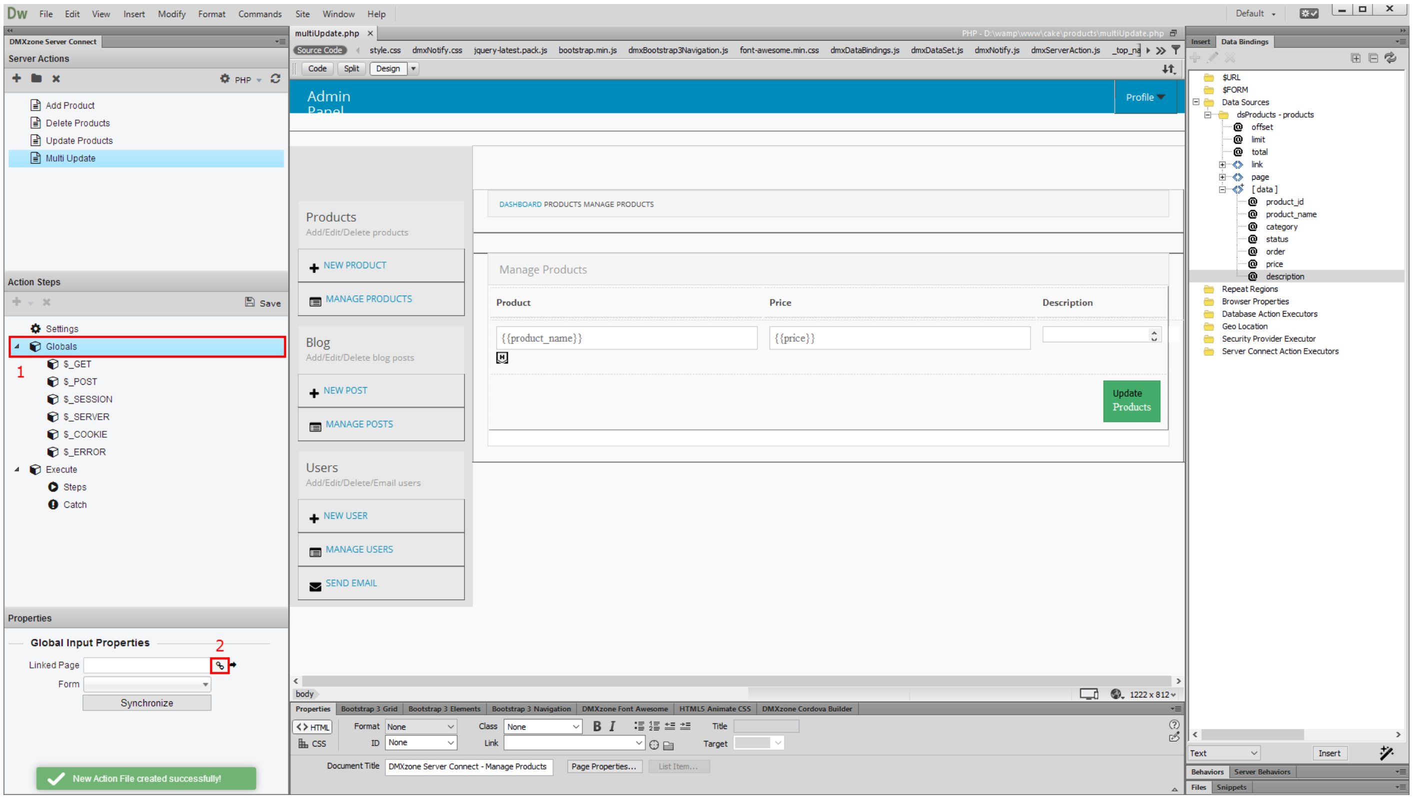
4. Do the same for the text fields, binding the proper bindings to the input.data.value and input.name.



5. Click **add new server action button (1)**, when you are done binding the fields names and values. And **add a name** for this server action **(2)**.

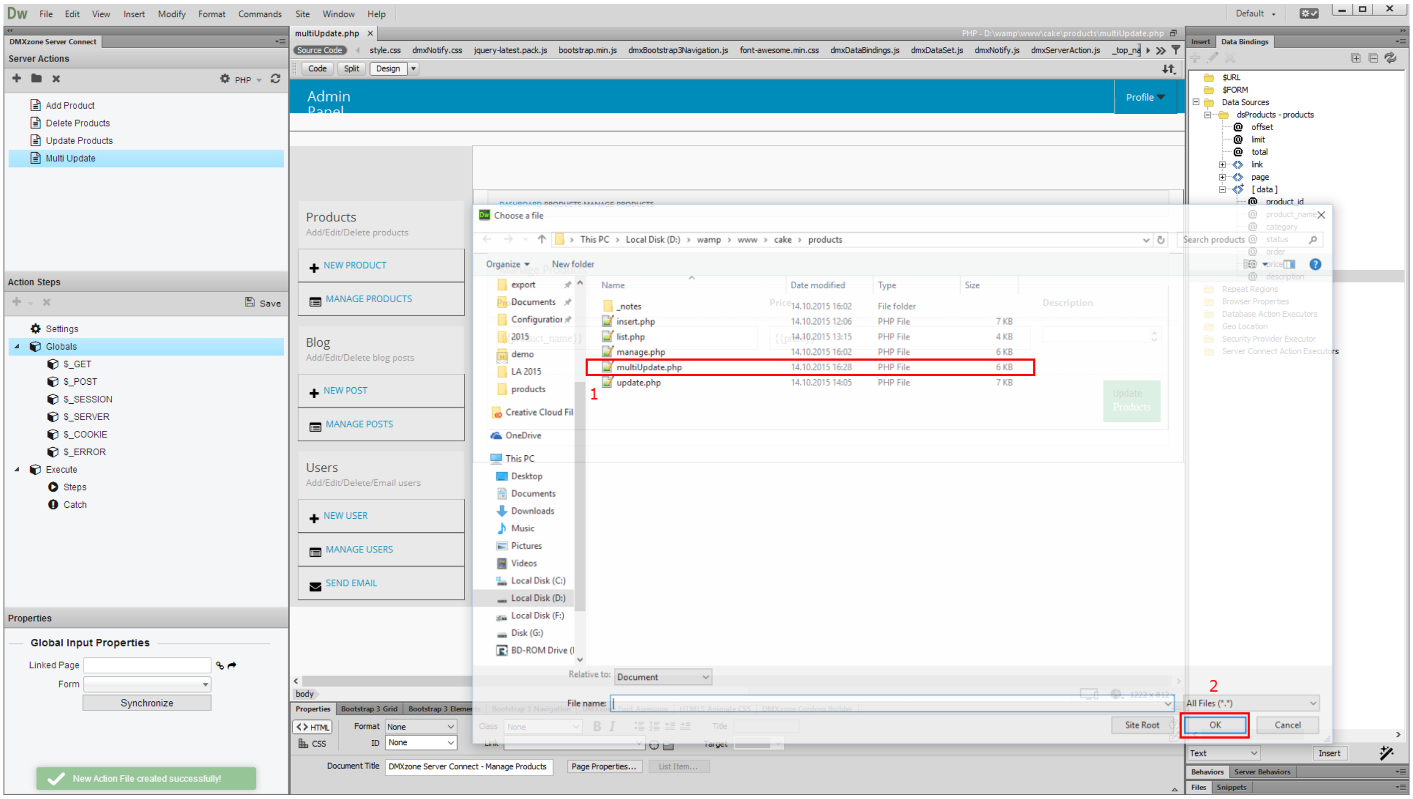


6. Expand the **global variables** of this action **(1)** and **link your page**, containing the form inputs. **(2)**.

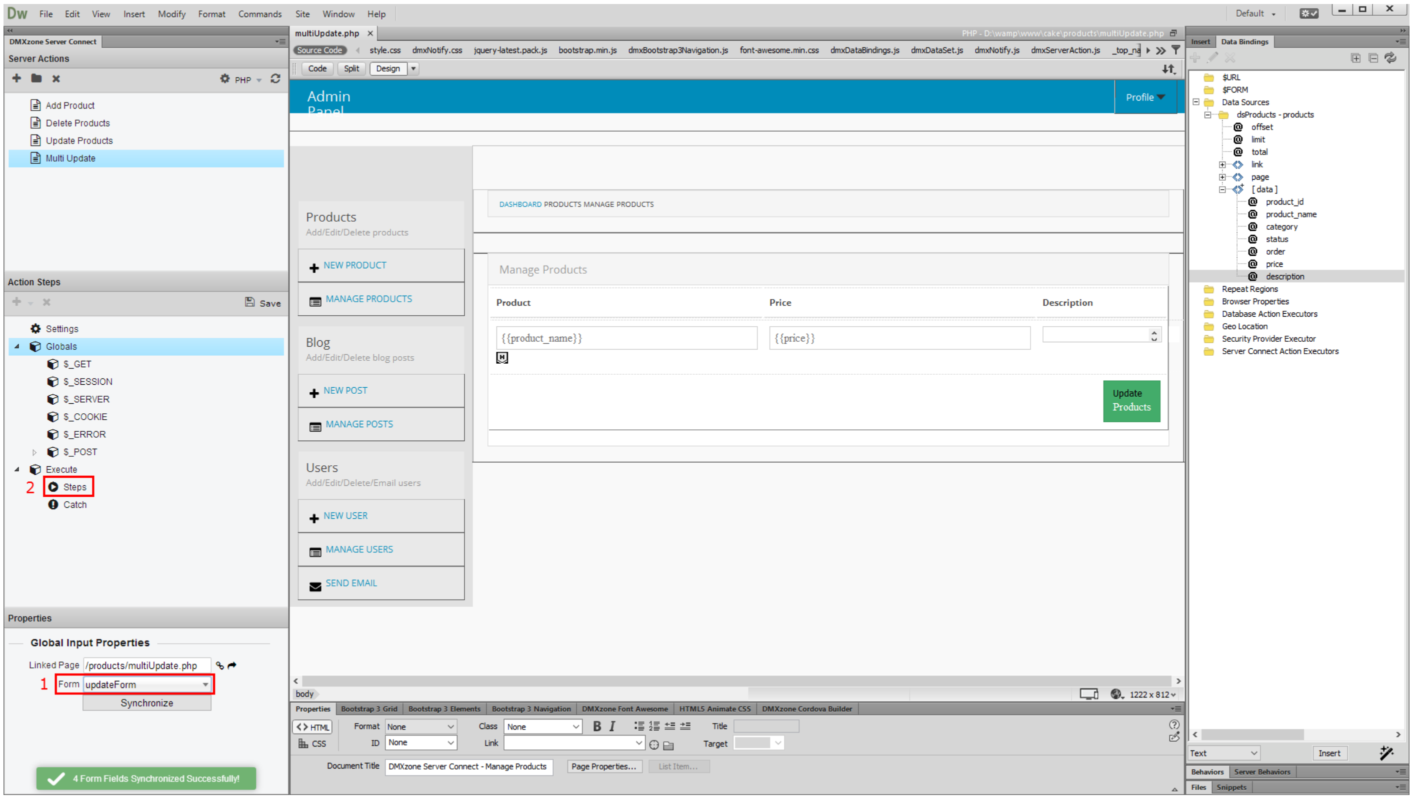


New Action File created successfully!

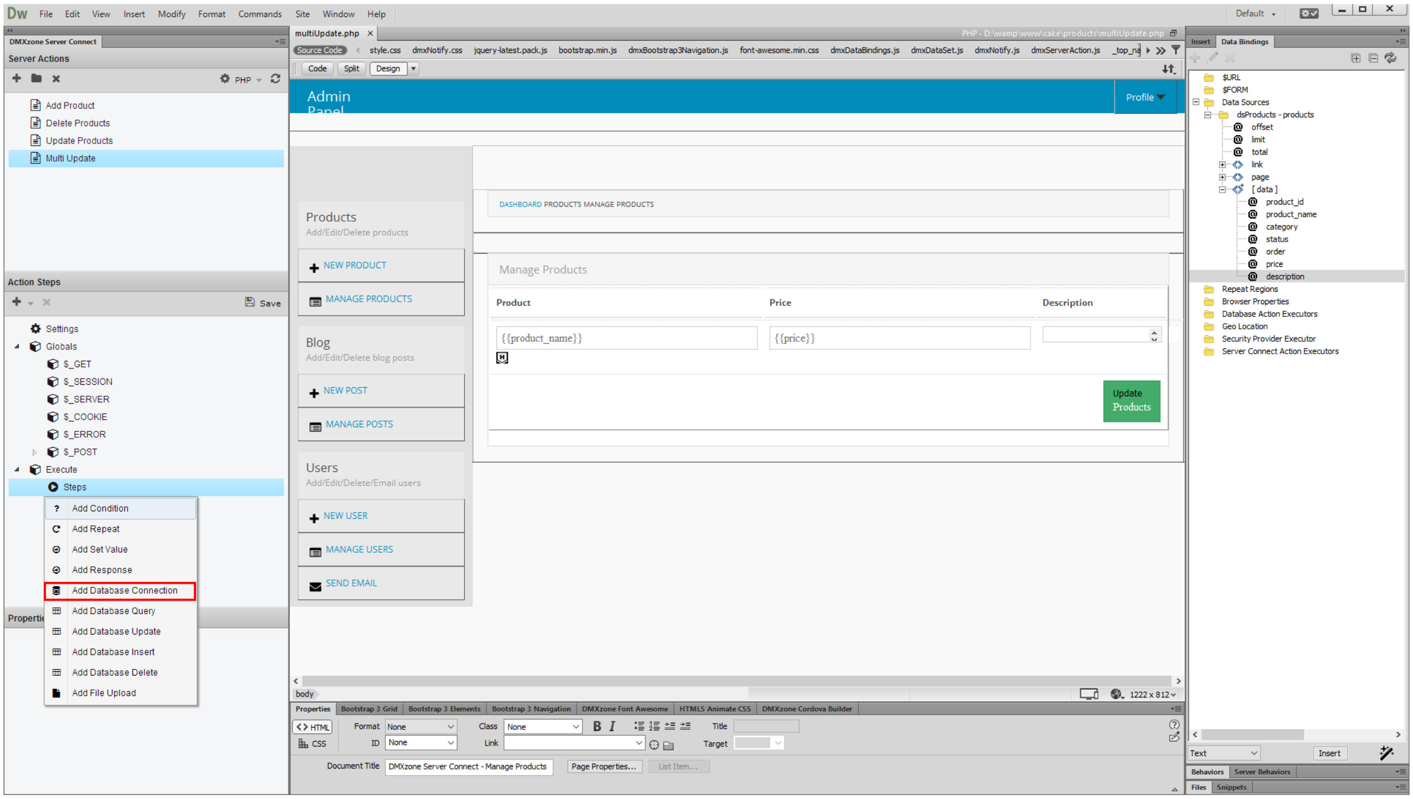
7. Choose **your page (1)** and **click OK (2)**.



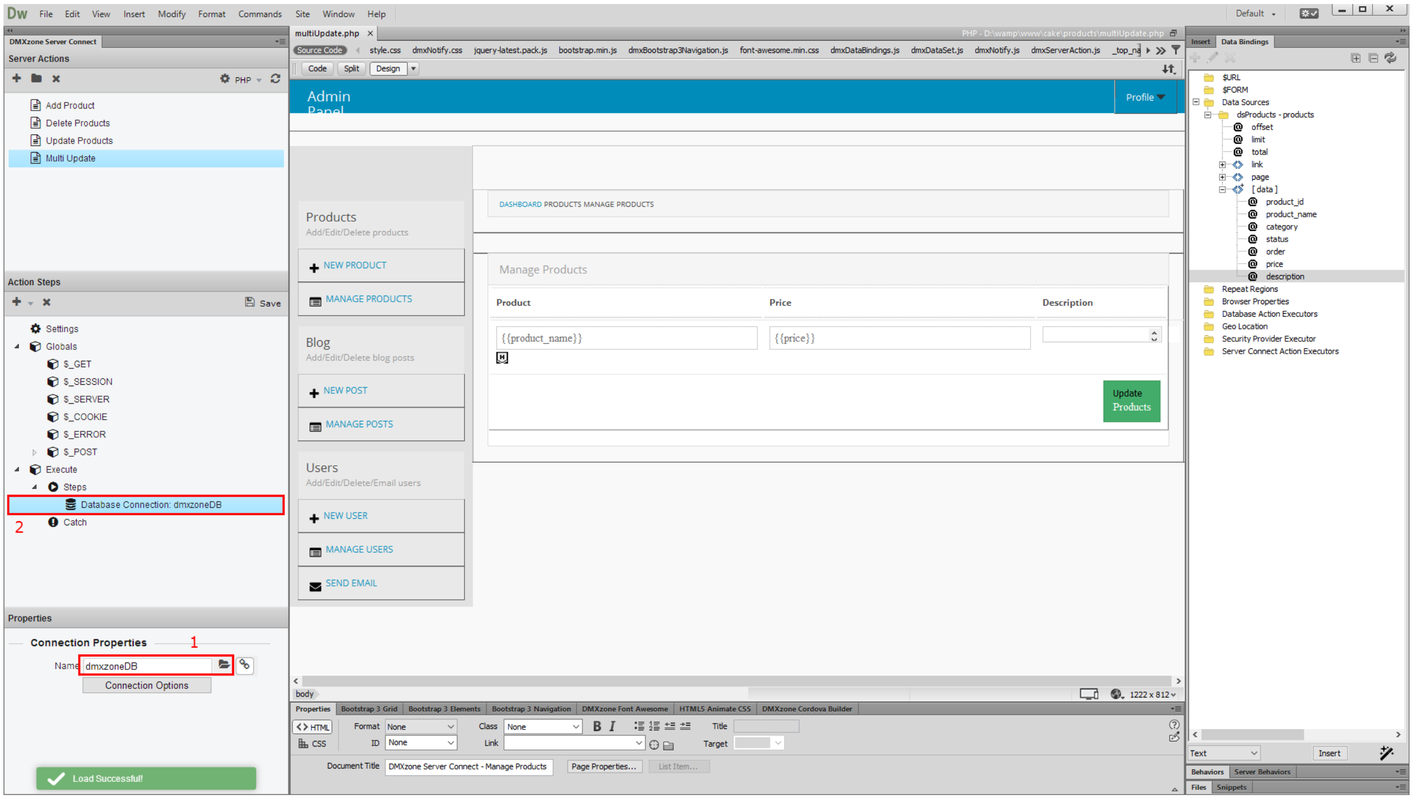
8. Select your **form**, from the dropdown **(1)** and right **click steps**, under the execute list **(2)**.



9. Now, add a **database connection**.



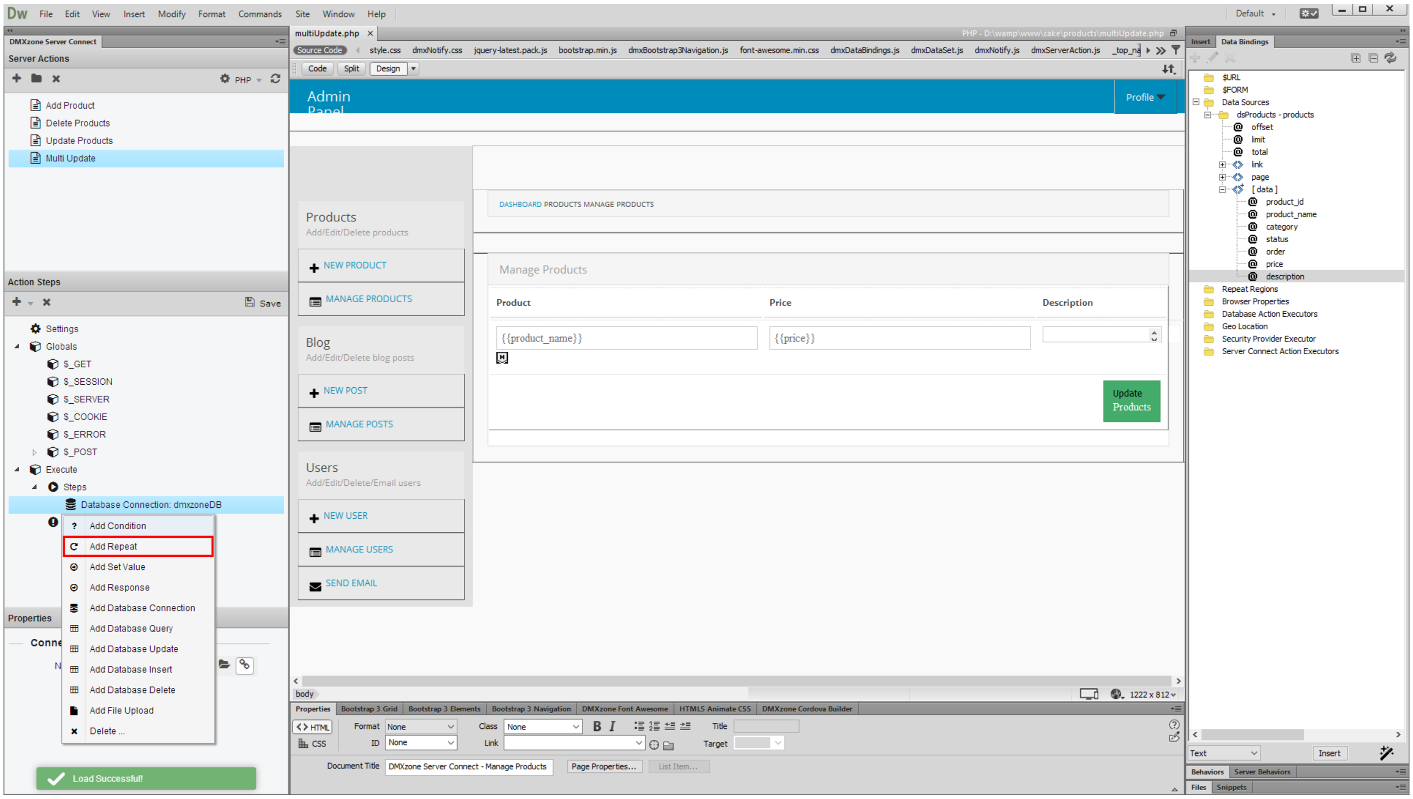
10. We already defined our database connection, when we created our data source, so we just **select it from the dropdown (1)**. Right click the **database connection step (2)**.



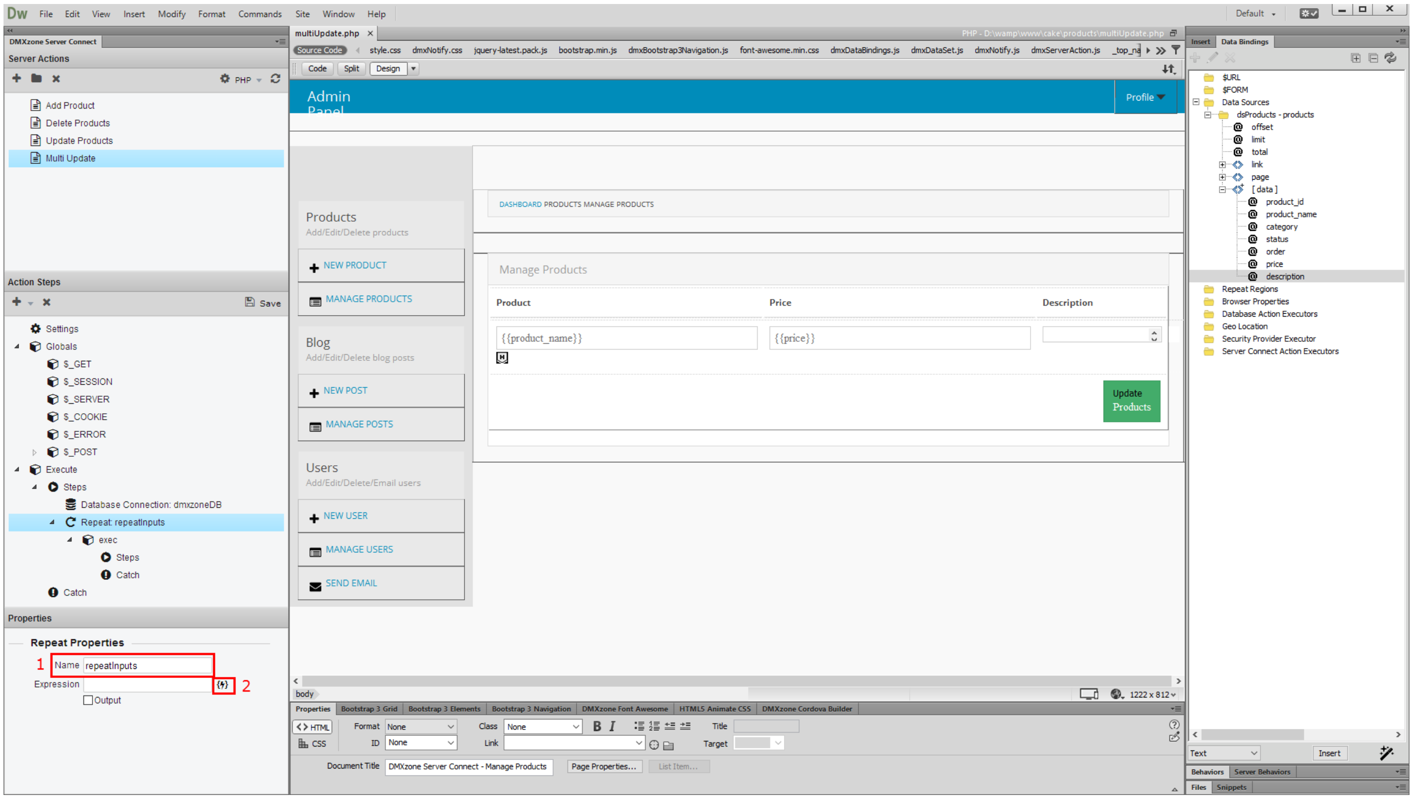
Connection Properties

Name dmxzoneDB

11. Now, add a **repeater**.



12. And **a name** for the repeater **(1)**. Then click the **server data bindings picker (2)**, in order to the repeat expression.



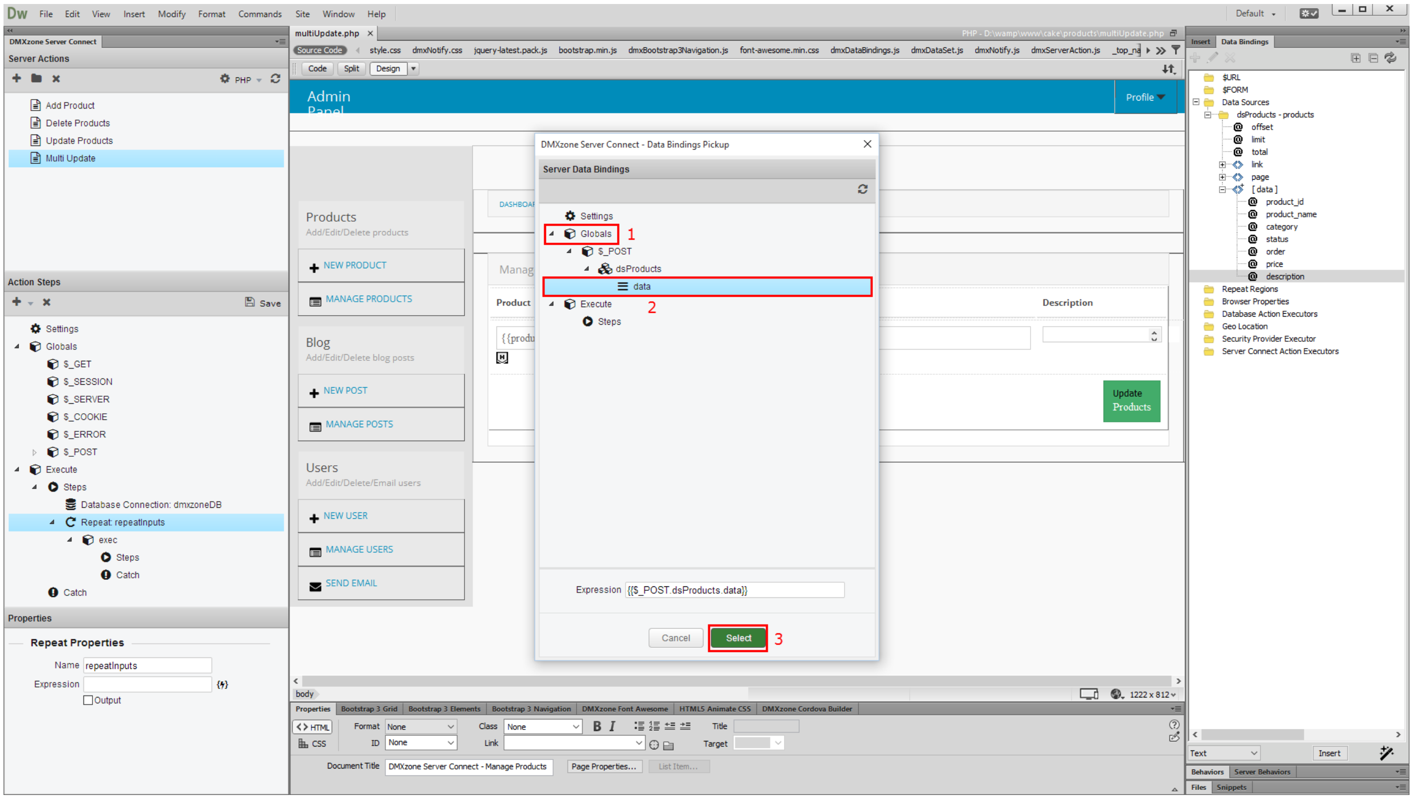
Repeat Properties

1 Name repeatInputs

Expression 2

□ Output

13. Expand the **globals (1)** and under post you'll find an object, which has the same name as your data source. Select the **data array**, located inside it **(2)**. When you're done, click **select (3)**.

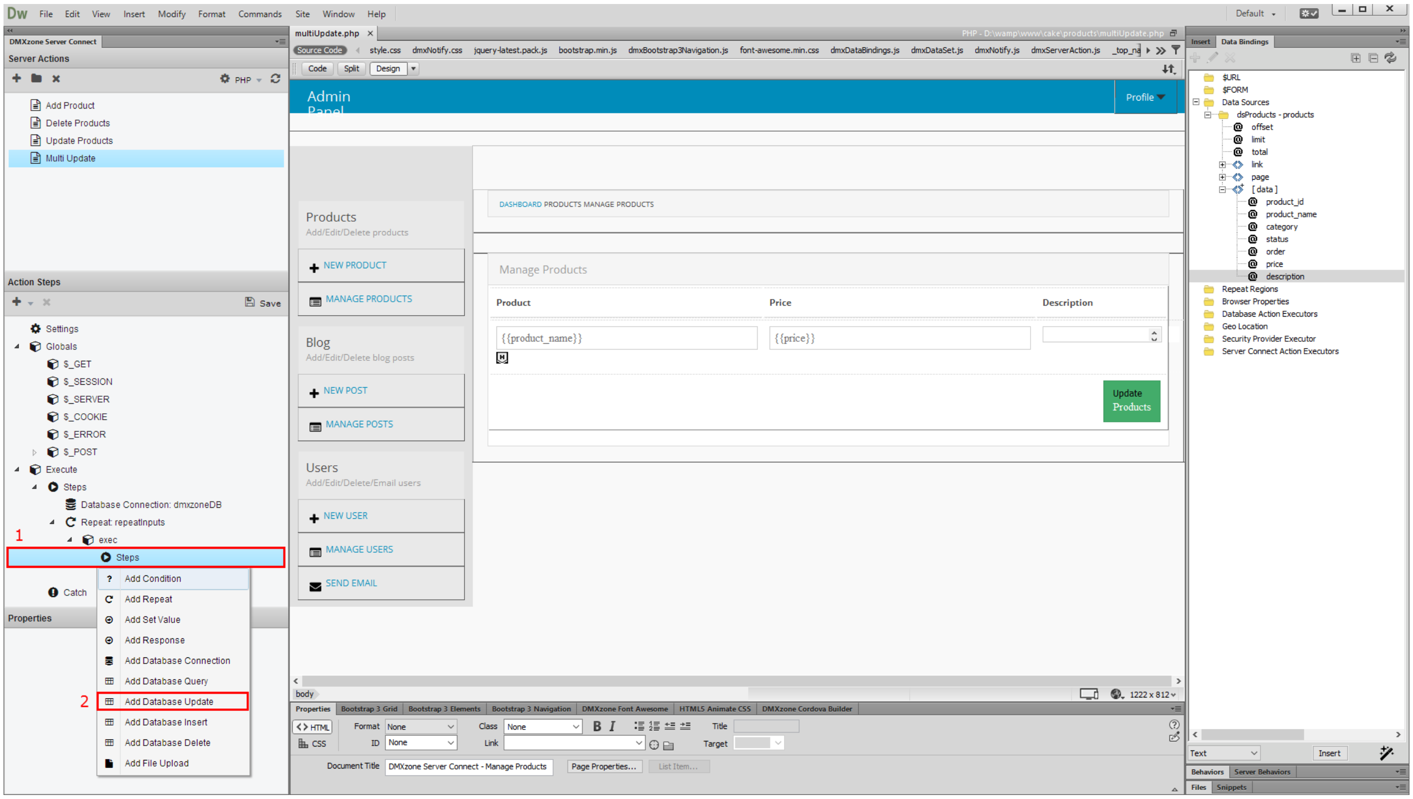


Expression {{$_POST.dsProducts.data}}

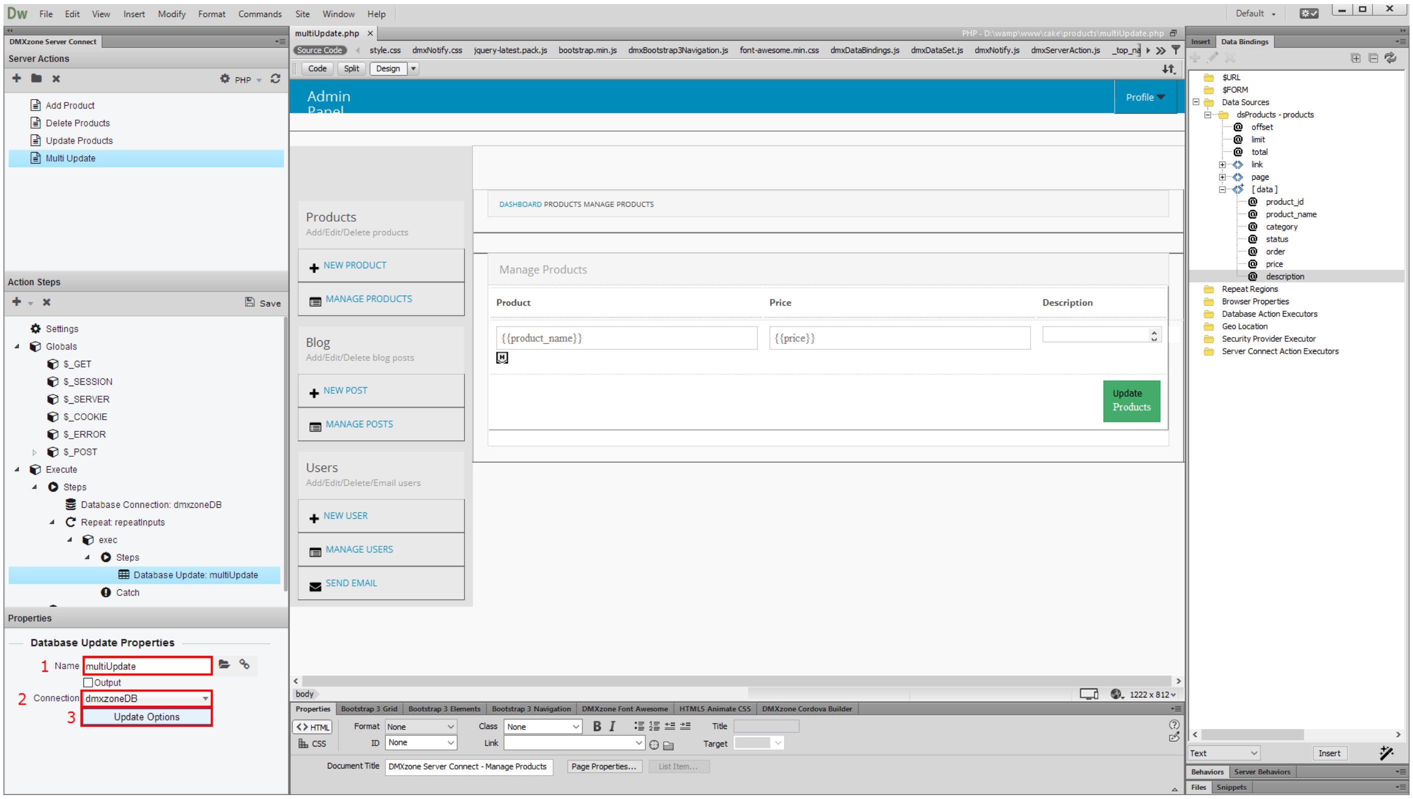
Name repeatInputs

Document Title DMXzone Server Connect - Manage Products

14. **Right click steps**, located in the repeat that we've just added **(1)** and **add database update (2)**.



15. Add **a name for this update (1)** and **select your connection** from the dropdown **(2)**. When you're done, click **update options button (3)**.



Database Update Properties

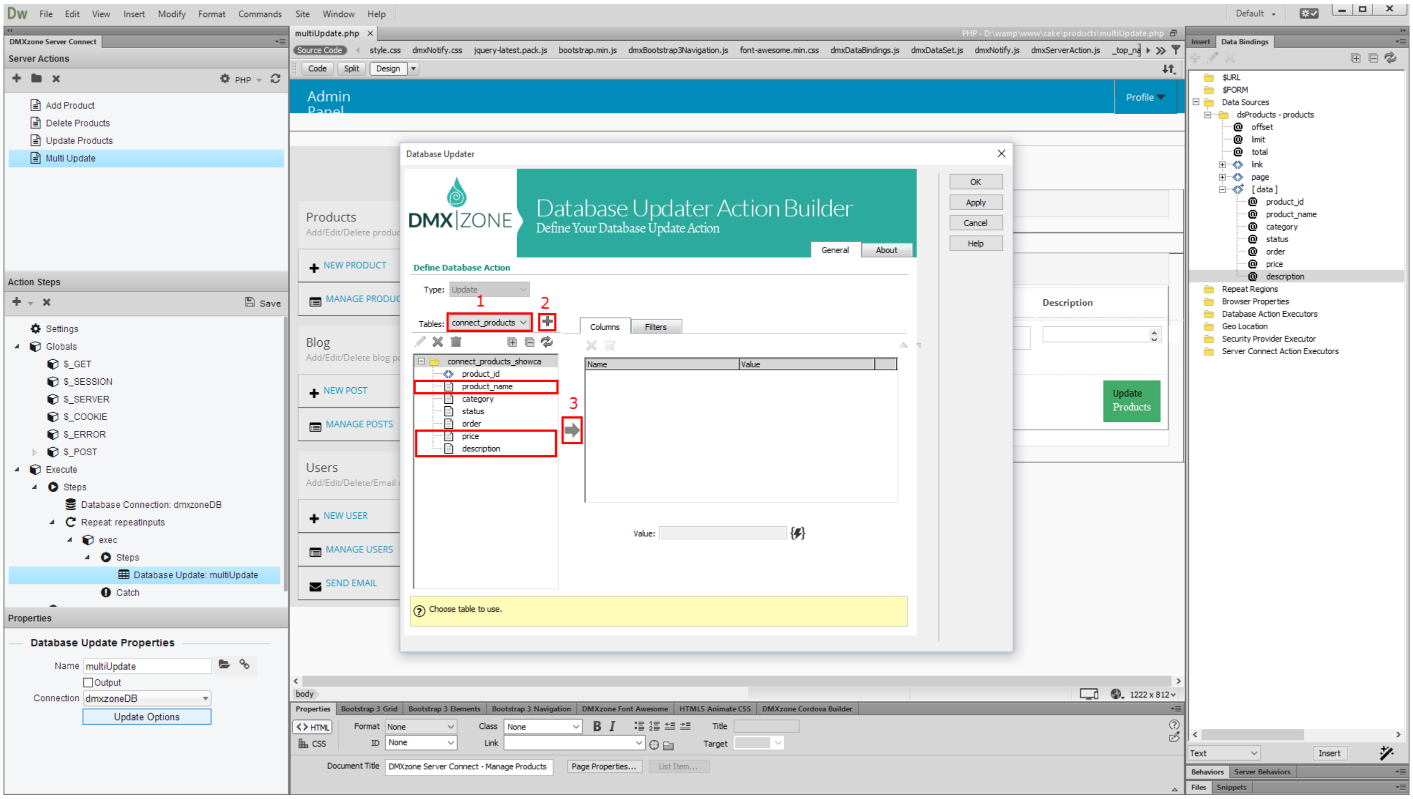
1 Name: multiUpdate

☐ Output

2 Connection: dmxzoneDB

3 Update Options

16. **Select (1)** and **add (2) the database table** that you want to update. Add the **database columns** that need to be updated **(3)**.



Database Updater

Database Updater Action Builder
Define Your Database Update Action

Define Database Action

Type: Update

Tables: connect_products

Choose table to use.

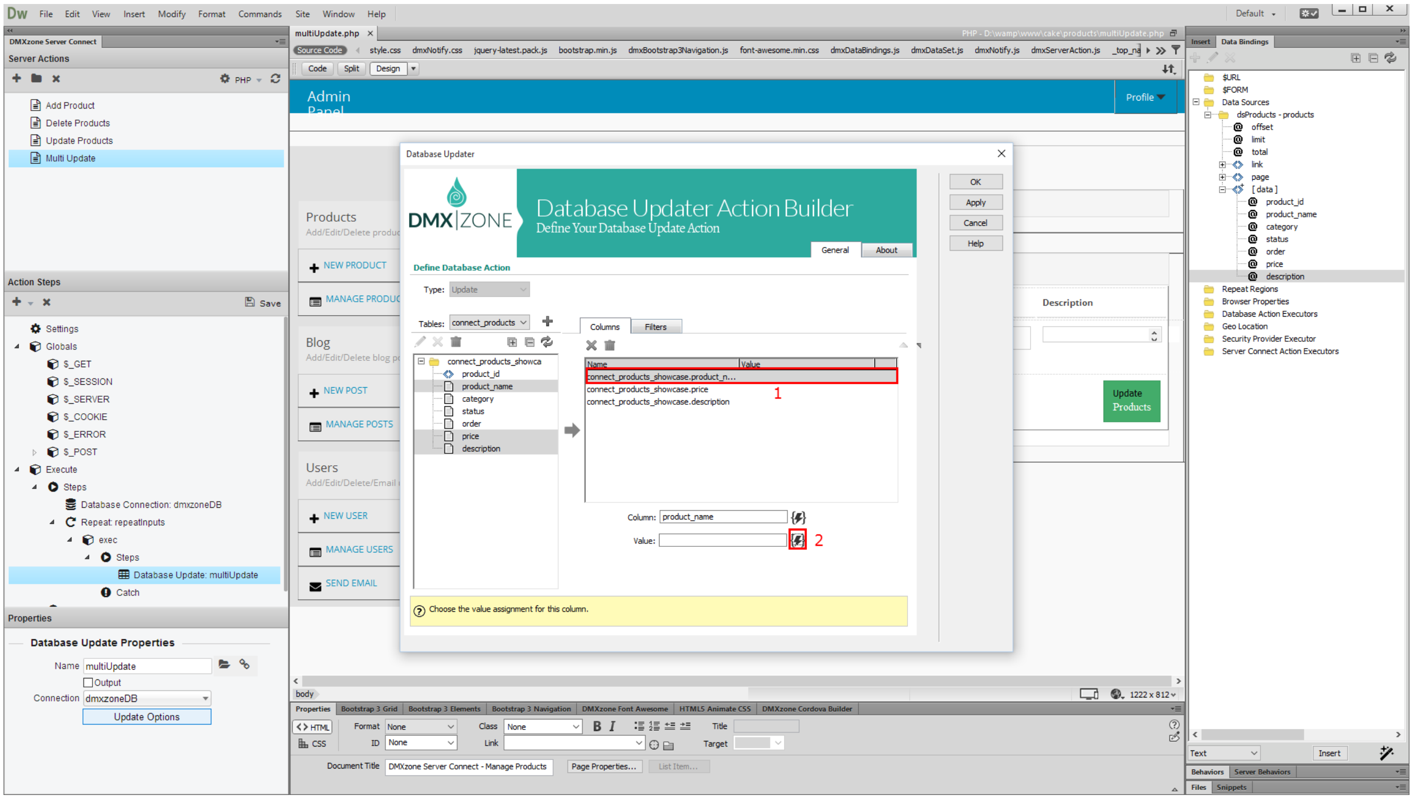
Database Update Properties

Name multiUpdate

□ Output

Connection dmxzoneDB

17. Select the **first column (1)** and click the **server data bindings picker (2)**, in order to setup its value.



Database Updater

Database Updater Action Builder
Define Your Database Update Action

Define Database Action

Type: Update

Tables: connect_products

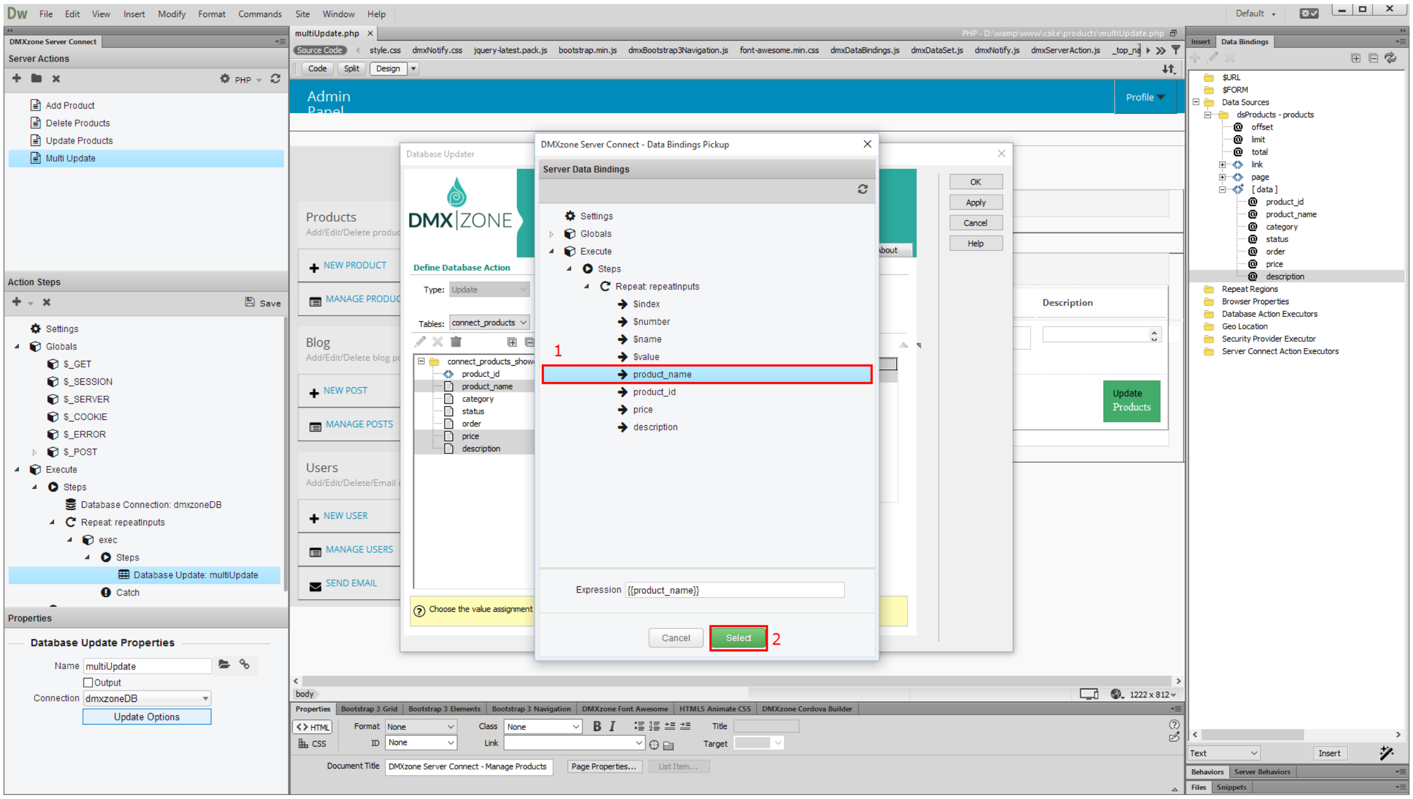
Name	Value
connect_products_showcase.product_n...	
connect_products_showcase.price	
connect_products_showcase.description	

Column: product_name

Value:

Choose the value assignment for this column.

18. Select the **input name (1)** that needs to be used, from the repeat results. In our case this is the product name. Then click the **select button (2)**.



Expression {{product_name}}

Name multiUpdate

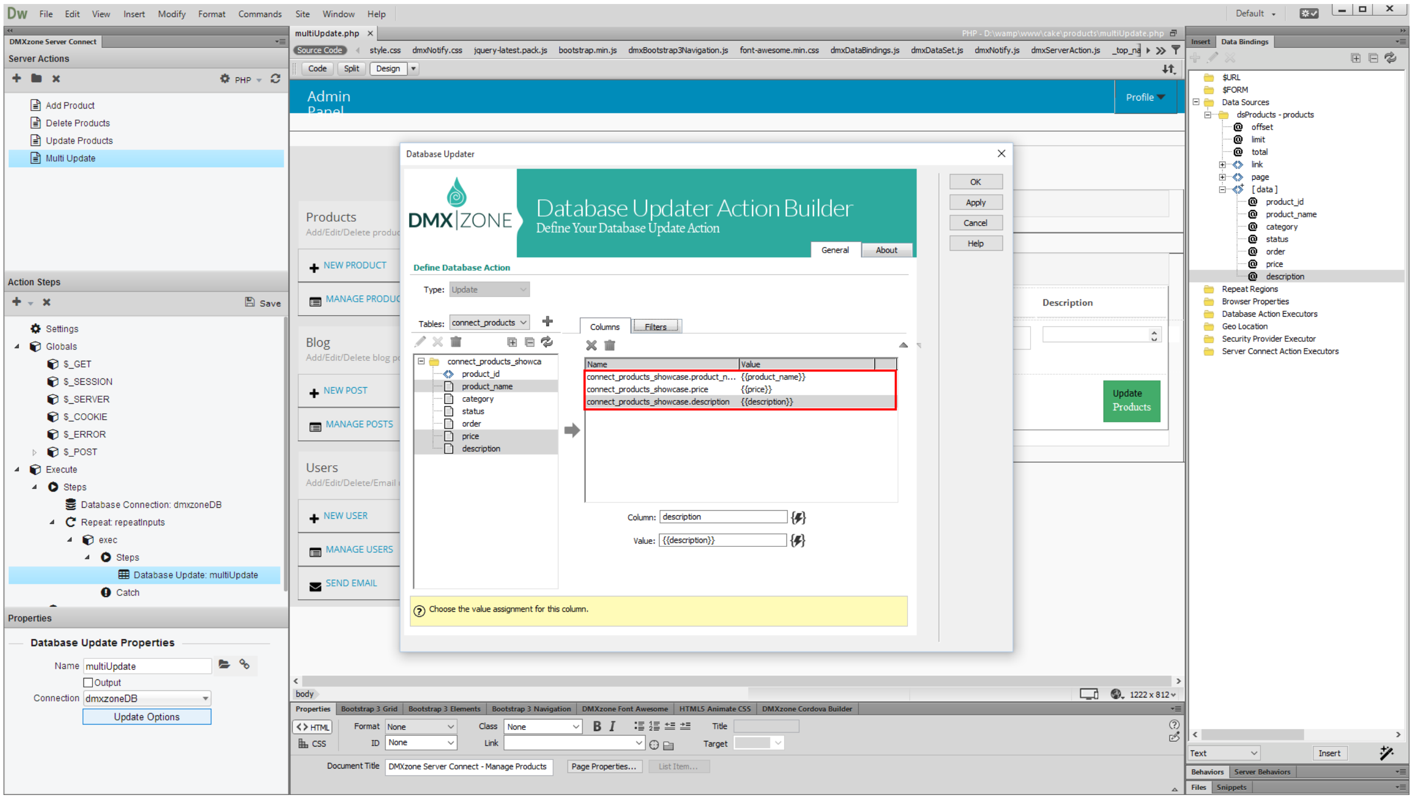
Connection dmxzoneDB

Type: Update

Tables: connect_products

Document Title DMXzone Server Connect - Manage Products

19. Do the same with the rest of the database columns, selecting the proper values.



Database Updater

Database Updater Action Builder
Define Your Database Update Action

Define Database Action

Type: Update

Tables: connect_products

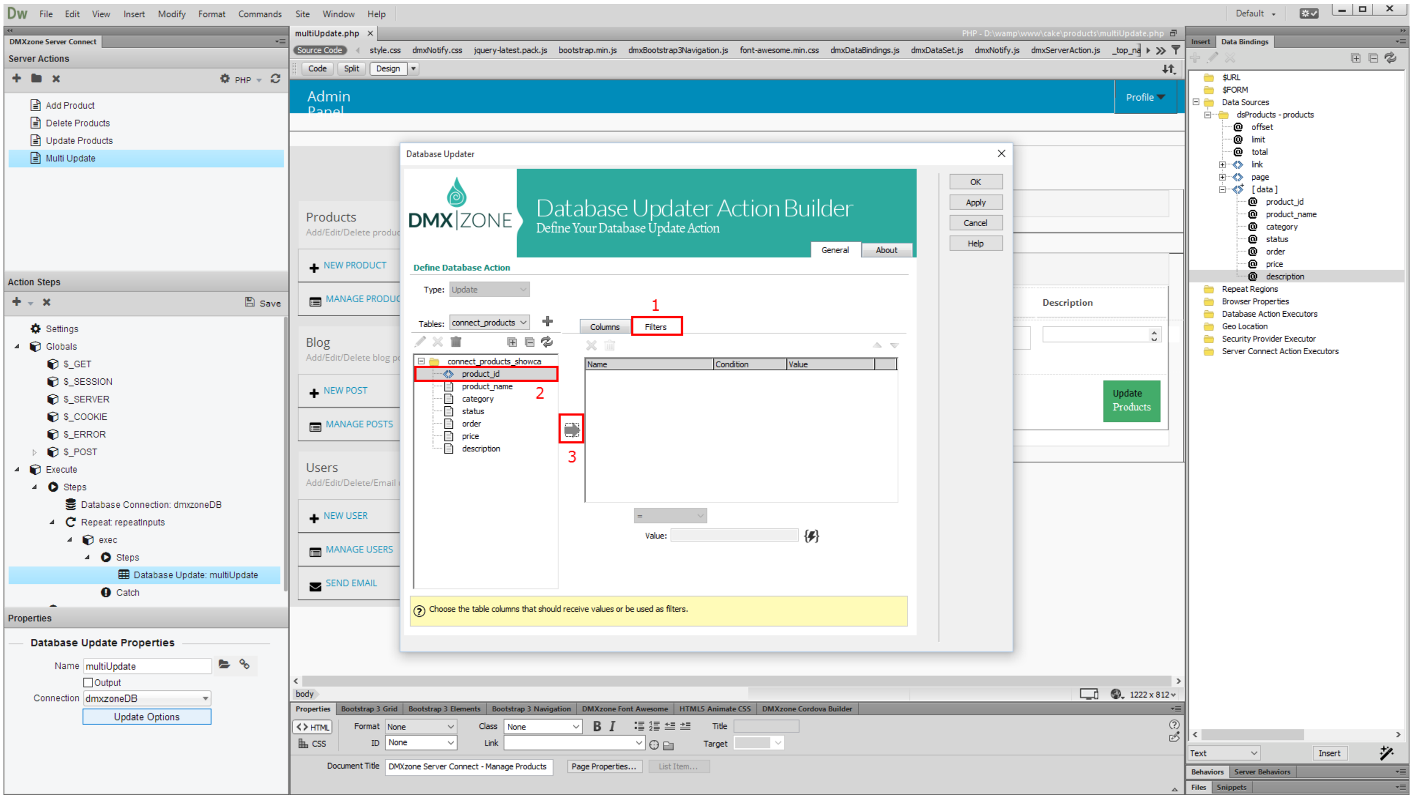
Name	Value
connect_products_showcase.product_n...	{{product_name}}
connect_products_showcase.price	{{price}}
connect_products_showcase.description	{{description}}

Column: description

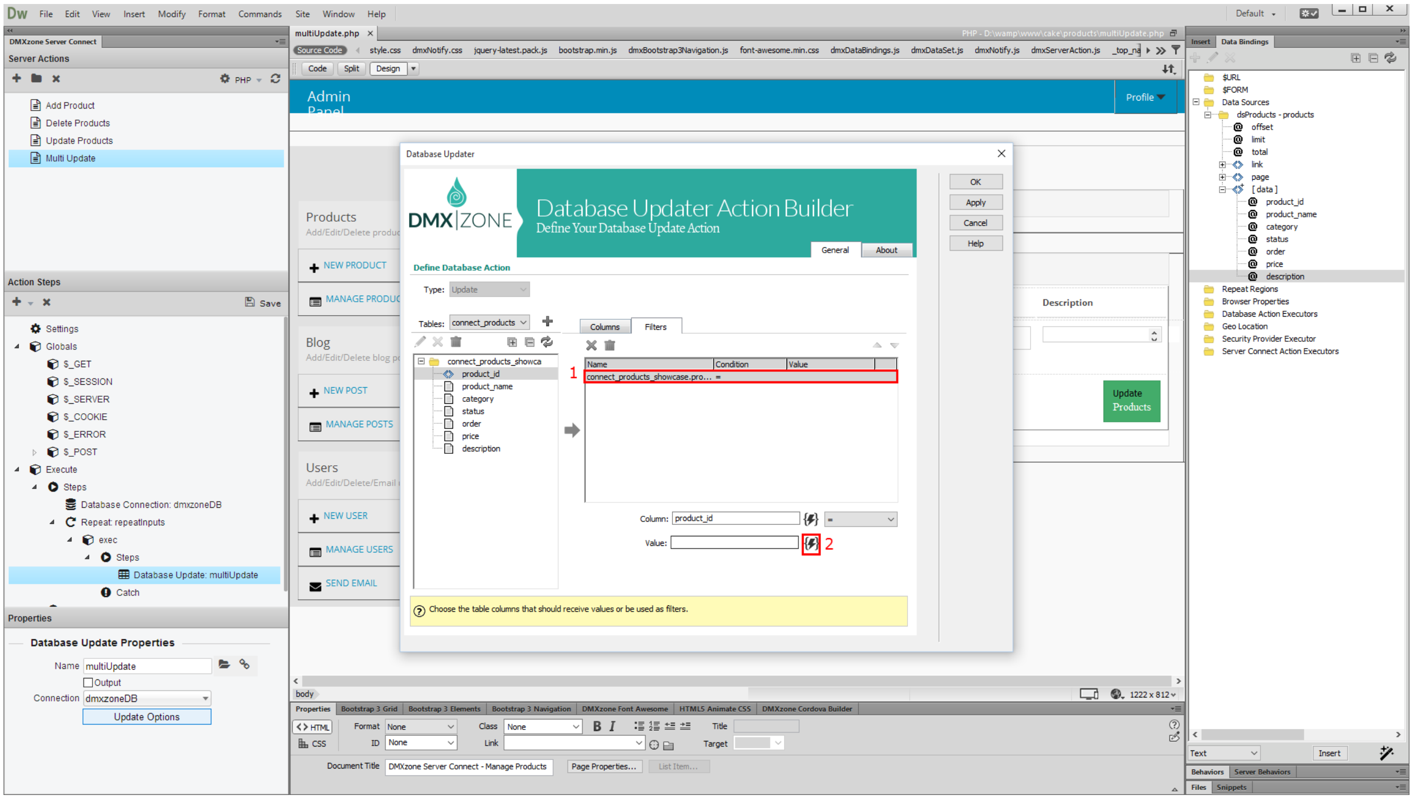
Value: {{description}}

Choose the value assignment for this column.

20. In the **filter tab (1)**, select the **column (2)** that you want to filter your products by, and **add it (3)** to filters list.



21. Select the **column (1)** and assign a value to it, using the **server data bindings picker (2)**.



Type: Update

Tables: connect_products

Name	Condition	Value
connect_products_showcase.pro...	=	

Column: product_id

Value:

Choose the table columns that should receive values or be used as filters.

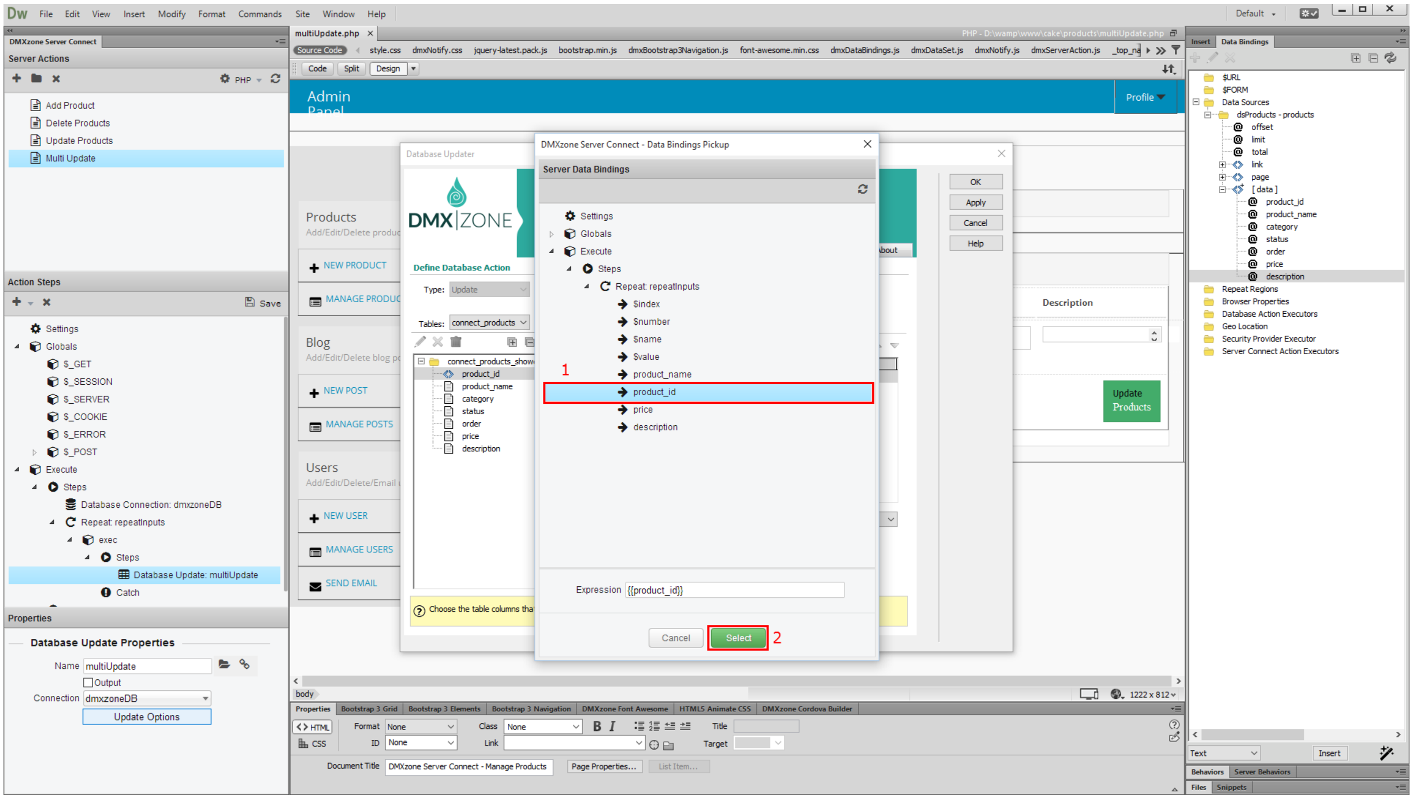
Database Update Properties

Name: multiUpdate

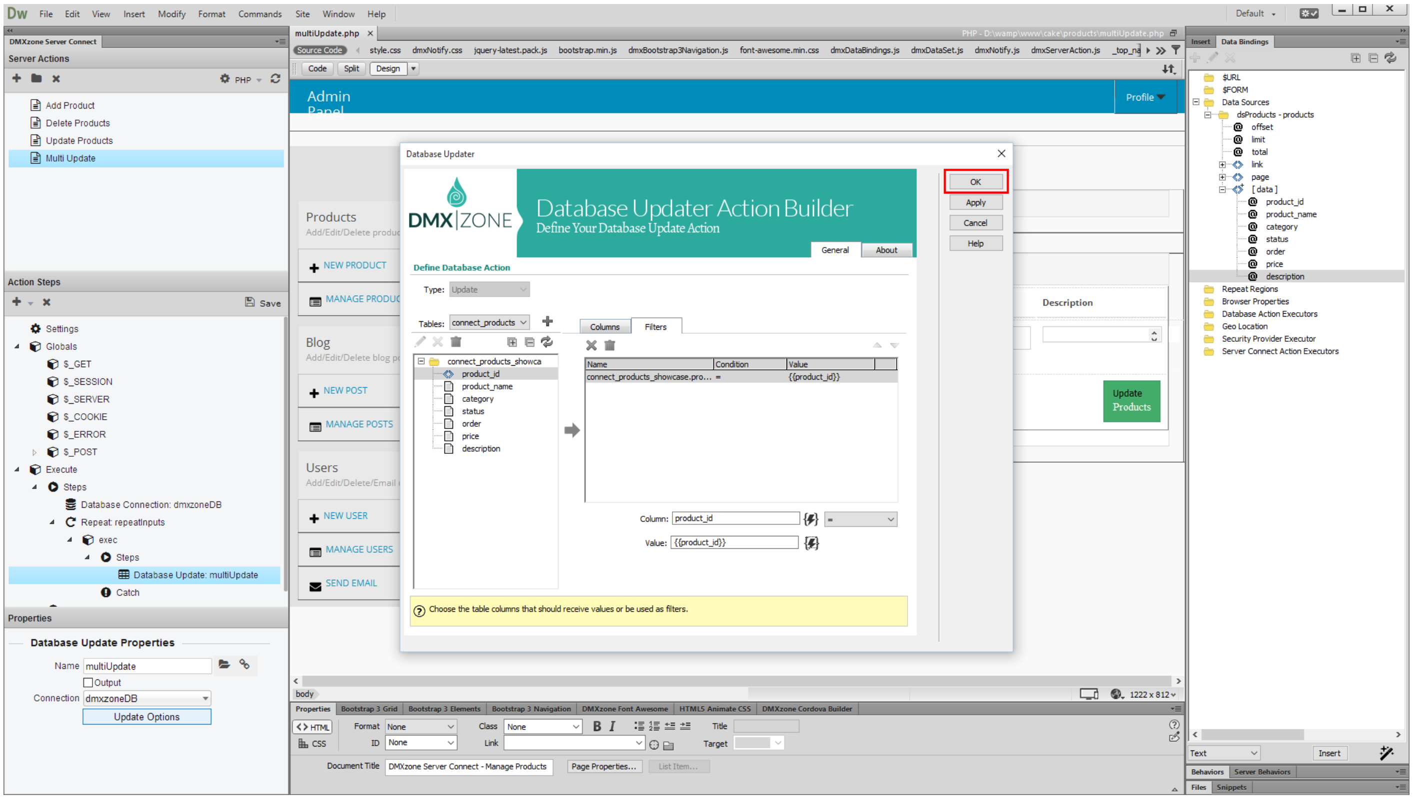
☐ Output

Connection: dmxzoneDB

22. Select the **hidden field name** from the repeat results **(1)** and click the **select button (2)**.



23. Click OK when you're done.



Database Updater

Database Updater Action Builder
Define Your Database Update Action

Define Database Action

Type: Update

Tables: connect_products

Name	Condition	Value
connect_products_showcase.pro...	=	{{product_id}}

Column: product_id =

Value: {{product_id}}

Choose the table columns that should receive values or be used as filters.

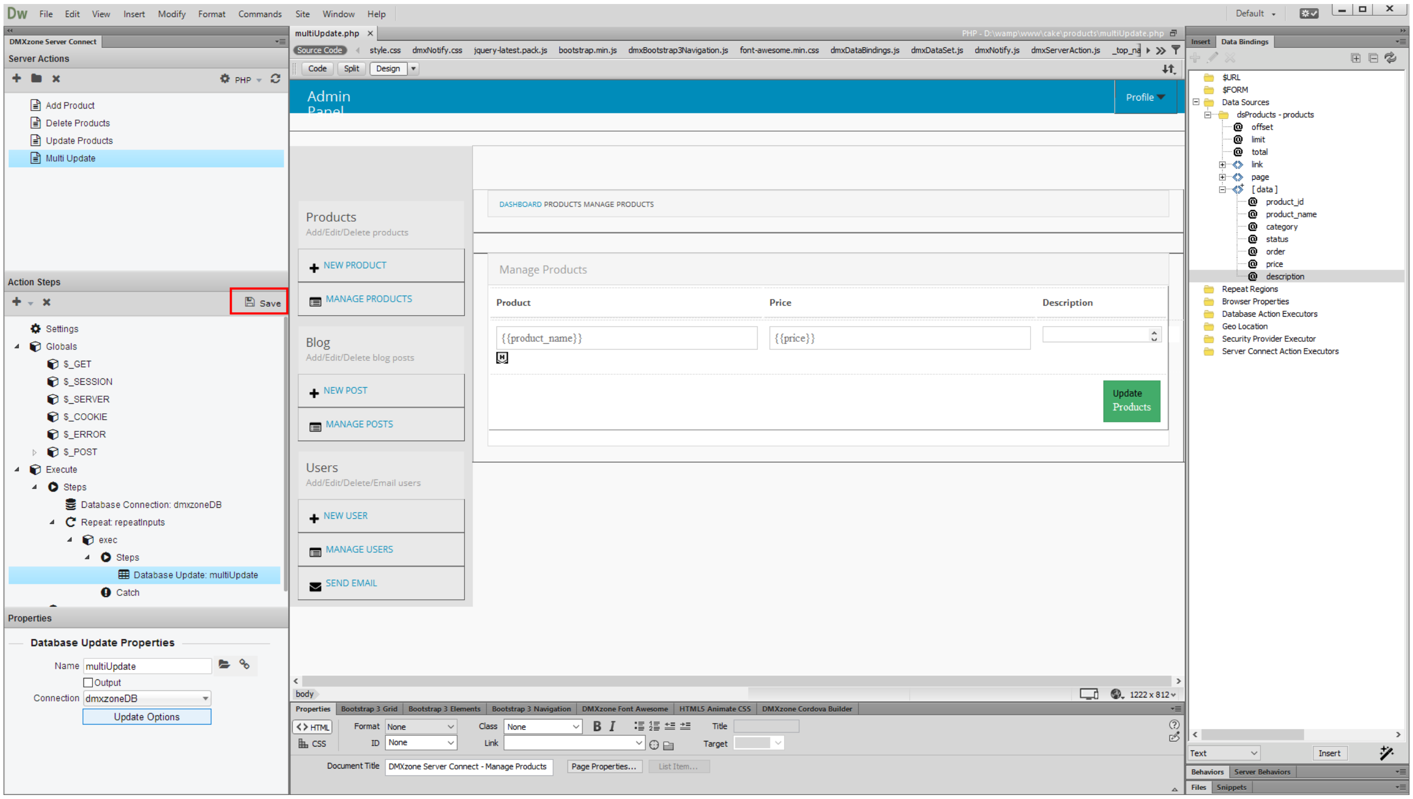
Database Update Properties

Name: multiUpdate

□ Output

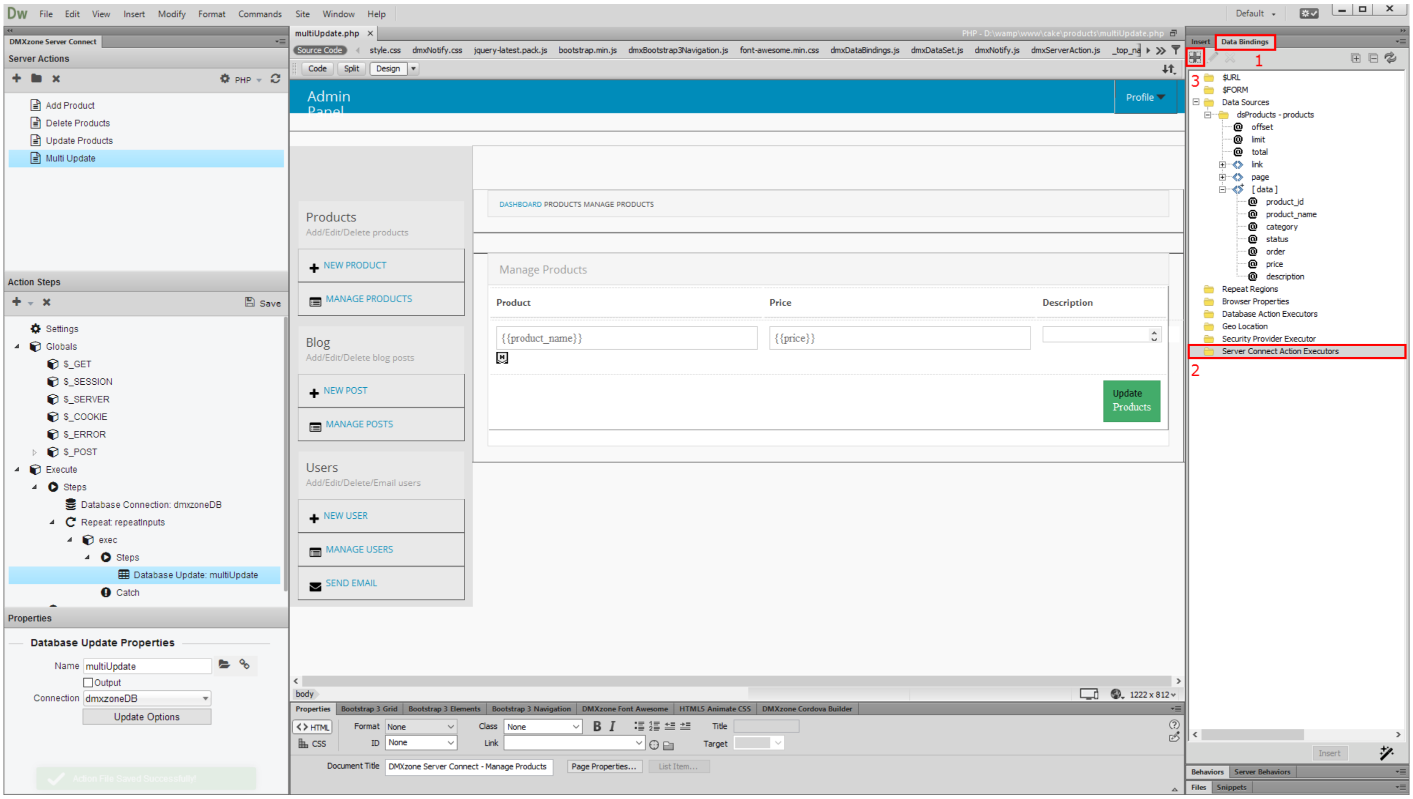
Connection: dmxzoneDB

24. Do not forget to **save your action steps**.

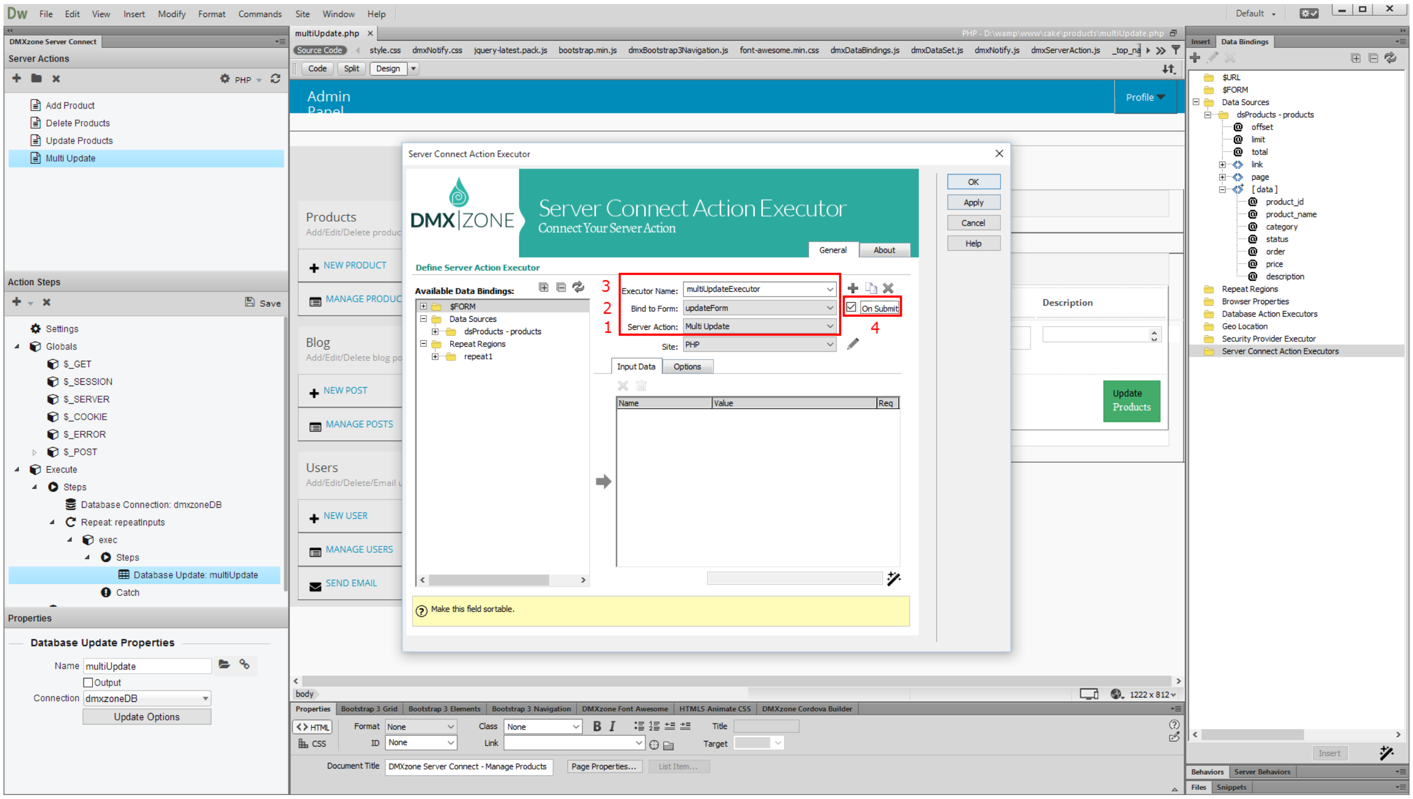


Name multiUpdate
☐ Output
Connection dmxzoneDB
Update Options

25. Then, in the **HTML5 Data Bindings panel (1)**, select the **server connect action executors (2)** and click **add new button (3)**.



26. Select your **server action** from the dropdown **(1)** and **bind it to the update form** on your page **(2).** Don't forget to **add a name for the executor (3)** and to **check the on submit checkbox (4)**. When you're done, click OK.



Server Connect Action Executor

Define Server Action Executor

3 Executor Name: multiUpdateExecutor

2 Bind to Form: updateForm

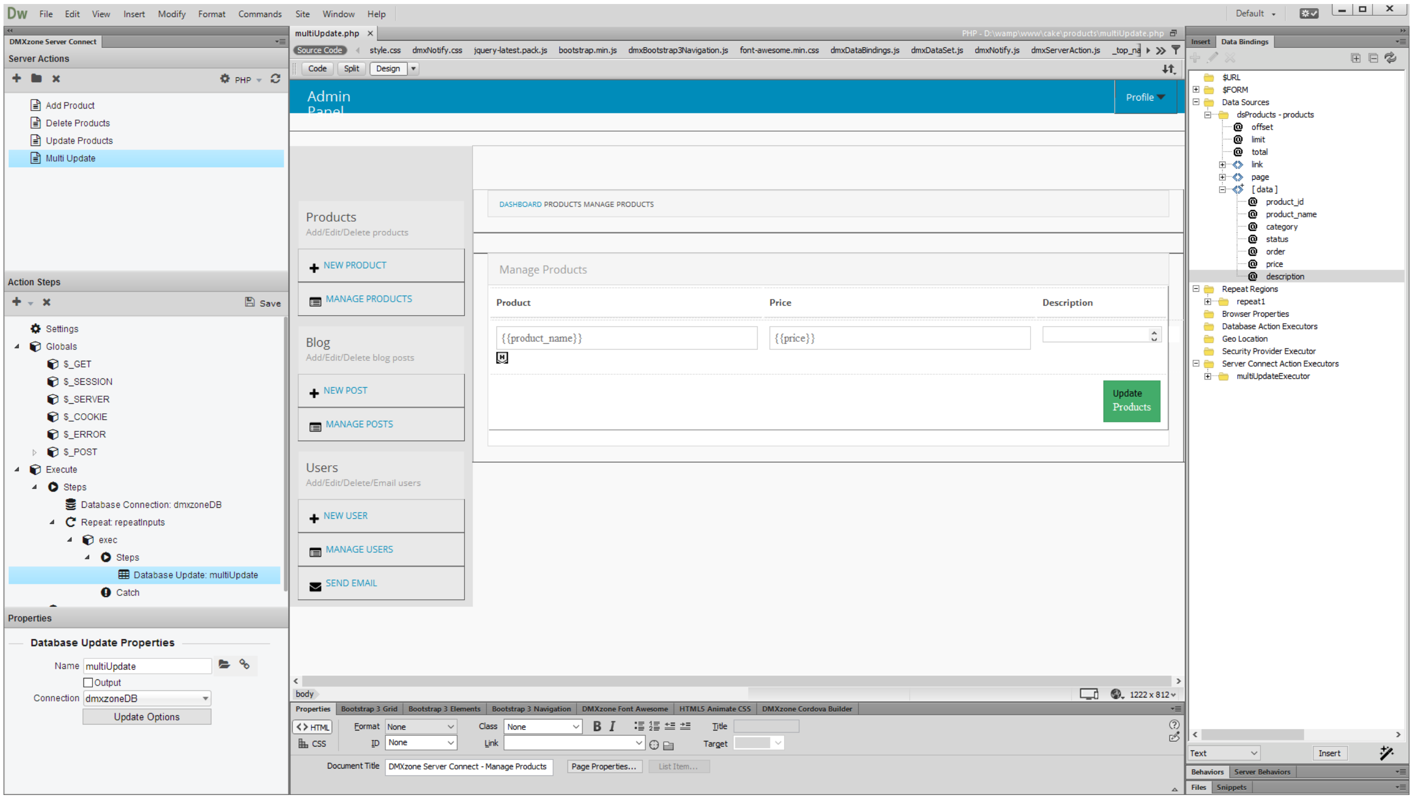
1 Server Action: Multi Update

Site: PHP

4 ☑ On Submit

Make this field sortable.

27. Let’s preview the results.



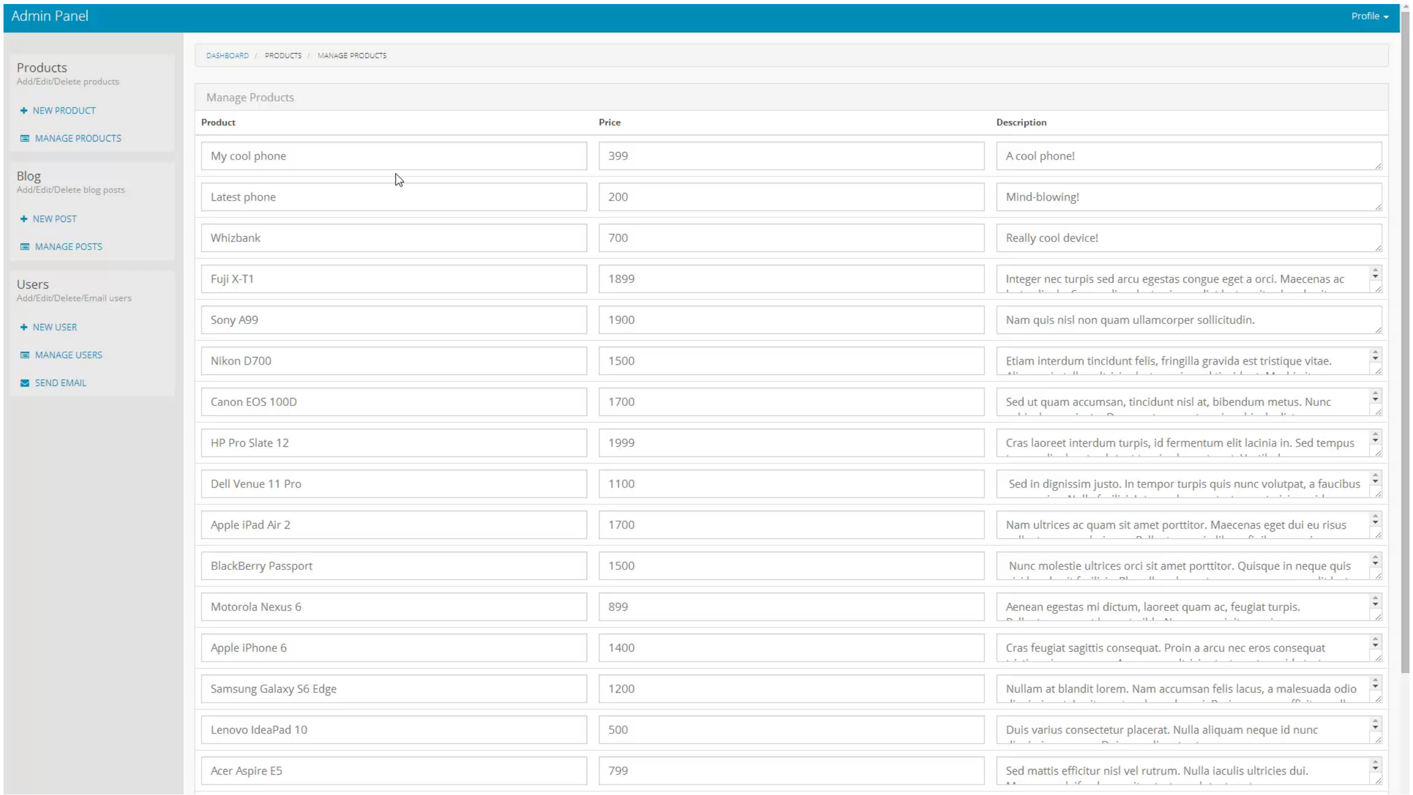
Name: multiUpdate

☐ Output

Connection: dmxzoneDB

Document Title: DMXzone Server Connect - Manage Products

28. You can edit different properties, for different records, and then when you click the update button the changes are immediately saved! That’s how easy it is to update multiple records, using DMXzone Server Connect.



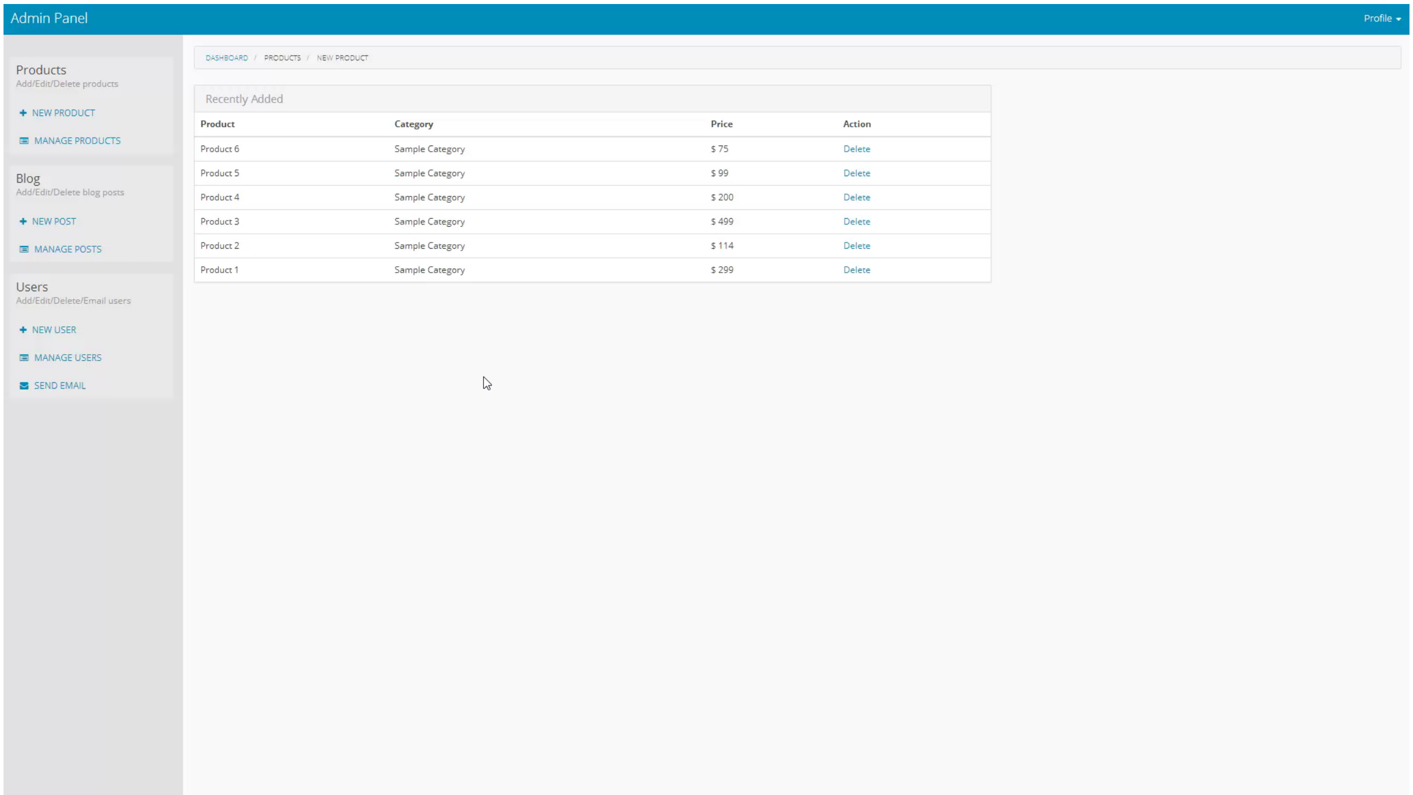
Manage Products

Product	Price	Description
My cool phone	399	A cool phone!
Latest phone	200	Mind-blowing!
Whizbank	700	Really cool device!
Fuji X-T1	1899	Integer nec turpis sed arcu egestas congue eget a orci. Maecenas ac
Sony A99	1900	Nam quis nisl non quam ullamcorper sollicitudin.
Nikon D700	1500	Etiam interdum tincidunt felis, fringilla gravida est tristique vitae.
Canon EOS 100D	1700	Sed ut quam accumsan, tincidunt nisl at, bibendum metus. Nunc
HP Pro Slate 12	1999	Cras laoreet interdum turpis, id fermentum elit lacinia in. Sed tempus
Dell Venue 11 Pro	1100	Sed in dignissim justo. In tempor turpis quis nunc volutpat, a faucibus
Apple iPad Air 2	1700	Nam ultrices ac quam sit amet porttitor. Maecenas eget dui eu risus
BlackBerry Passport	1500	Nunc molestie ultrices orci sit amet porttitor. Quisque in neque quis
Motorola Nexus 6	899	Aenean egestas mi dictum, laoreet quam ac, feugiat turpis.
Apple iPhone 6	1400	Cras feugiat sagittis consequat. Proin a arcu nec eros consequat
Samsung Galaxy S6 Edge	1200	Nullam at blandit lorem. Nam accumsan felis lacus, a malesuada odio
Lenovo IdeaPad 10	500	Duis varius consectetur placerat. Nulla aliquam neque id nunc
Acer Aspire E5	799	Sed mattis efficitur nisl vel rutrum. Nulla iaculis ultricies dui.

Advanced: Delete Database Records

In this tutorial we will show you how delete database records, using DMXzone Server Connect. In order to do that you'll also need HTML5 Data Bindings, DMXzone Database Connector PHP/ASP and DMXzone Database Updater PHP/ASP extensions.

*You can also check out the video.

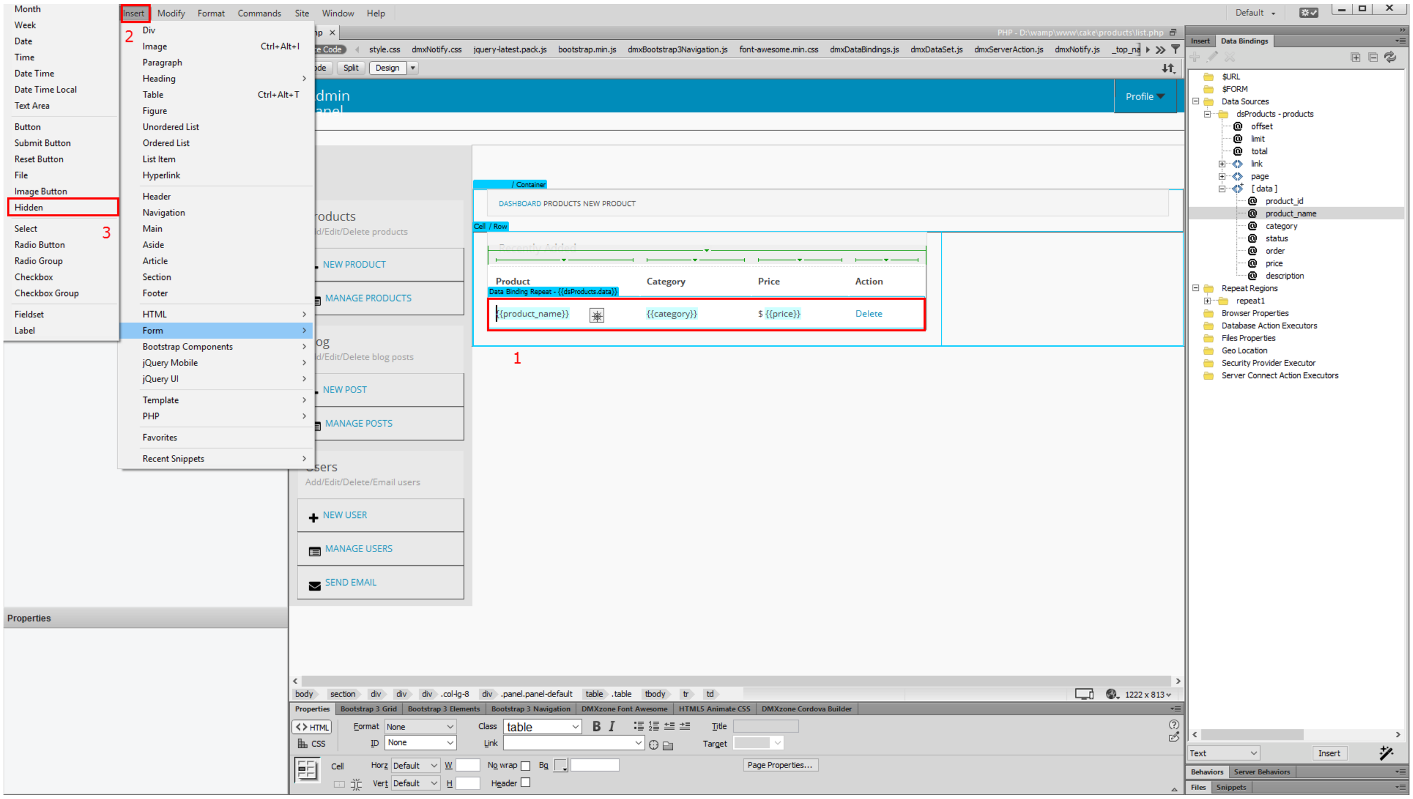


Recently Added

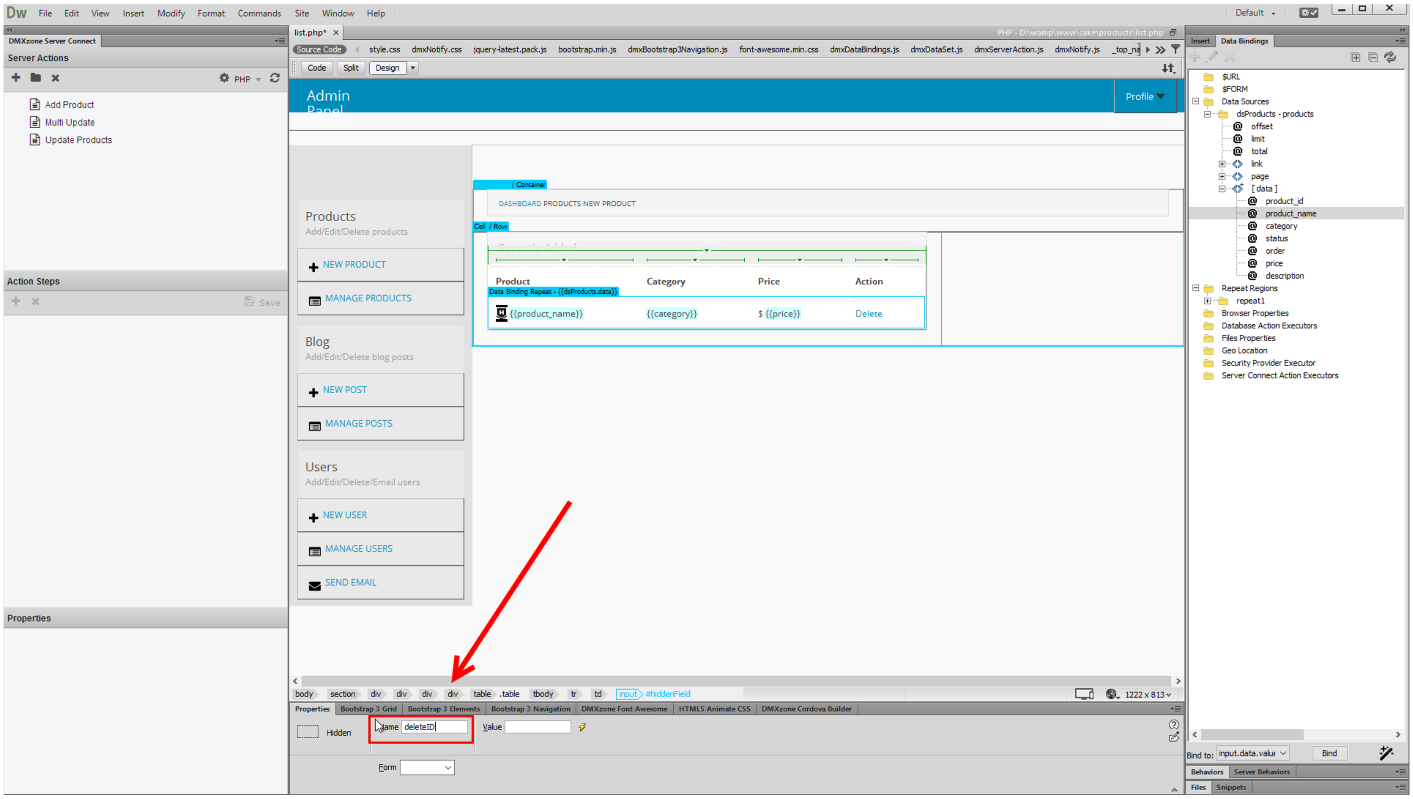
Product	Category	Price	Action
Product 6	Sample Category	$ 75	Delete
Product 5	Sample Category	$ 99	Delete
Product 4	Sample Category	$ 200	Delete
Product 3	Sample Category	$ 499	Delete
Product 2	Sample Category	$ 114	Delete
Product 1	Sample Category	$ 299	Delete

How to do it

1. We created a repeat region on our page, which lists the products from our database. We placed a simple dummy link, saying delete, for each of the records. We will use this link to delete records later. Click inside the **repeat region (1)**, open the **insert menu (2)** and under the form sub menu, select **hidden field (3)**.



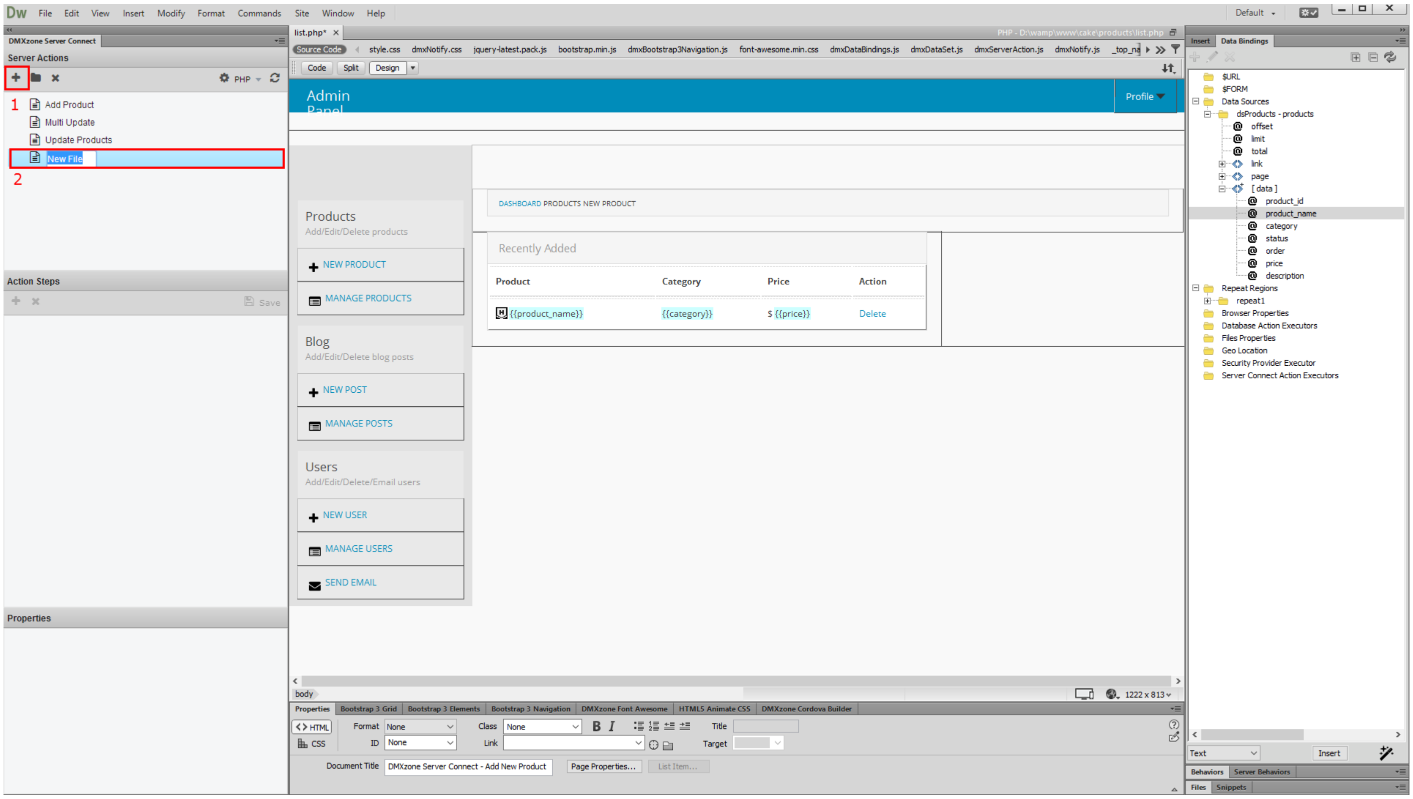
2. We will use this hidden field, in order to filter the records for the delete action. Let’s change its default name to **delete ID**.



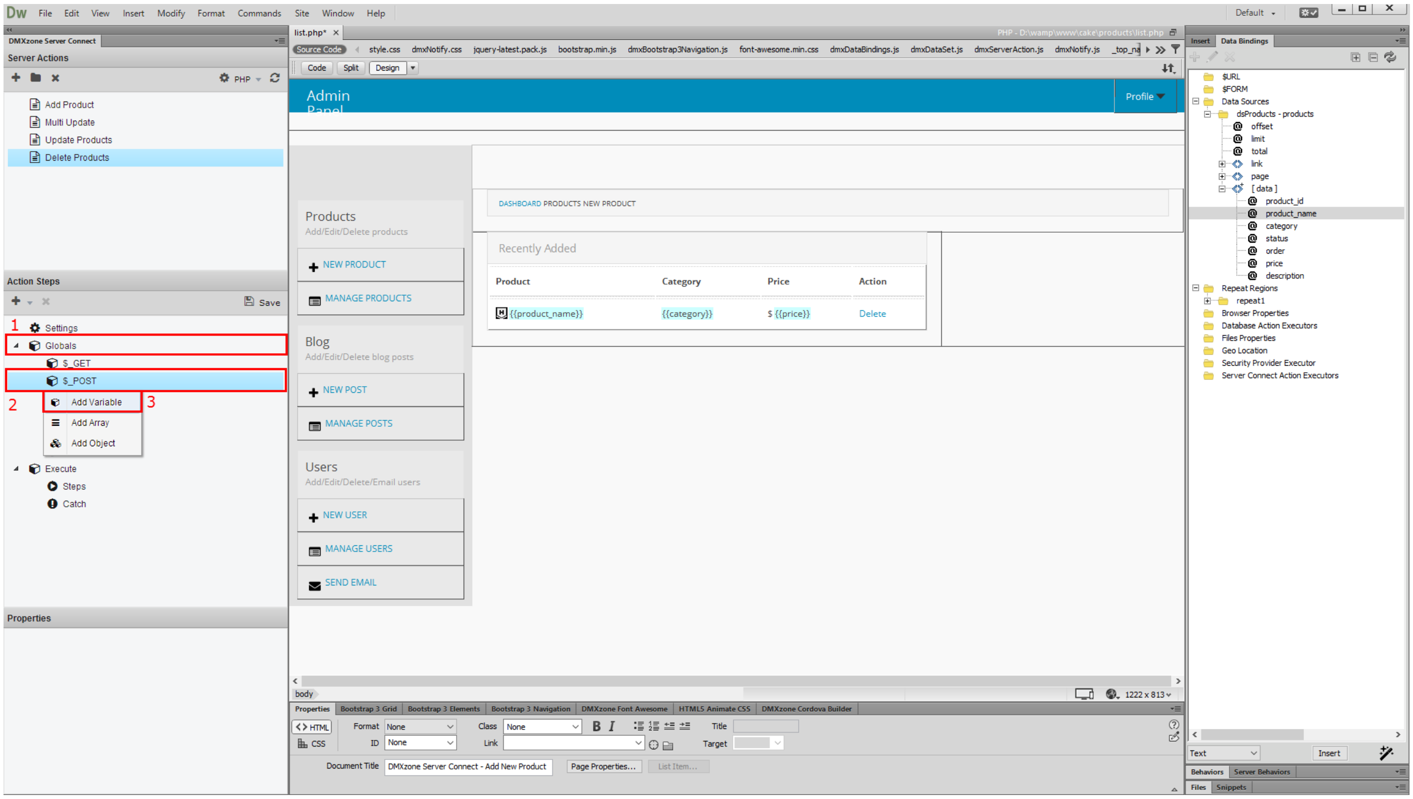
Hidden Name deleteID

Dw File Edit View Insert Modify Format Commands Site Window Help
DMXzone Server Connect
Server Actions
Add Product
Multi Update
Update Products
Action Steps
Properties
list.php*
Source Code style.css dmxNotify.css jquery-latest.pack.js bootstrap.min.js dmxBootstrap3Navigation.js font-awesome.min.css dmxDataBindings.js dmxDataSet.js dmxServerAction.js dmxNotify.js
Code Split Design
Admin Panel
Profile
Products
Add/Edit/Delete products
NEW PRODUCT
MANAGE PRODUCTS
Blog
Add/Edit/Delete blog posts
NEW POST
MANAGE POSTS
Users
Add/Edit/Delete/Email users
NEW USER
MANAGE USERS
SEND EMAIL
DASHBOARD PRODUCTS NEW PRODUCT
Product Category Price Action
Data Binding Repeat - {{dsProducts.data}}
{{product_name}} {{category}} $ {{price}} Delete
body section div div div div table .table tbody tr td input #hiddenField
1222 x 813
Properties Bootstrap 3 Grid Bootstrap 3 Elements Bootstrap 3 Navigation DMXzone Font Awesome HTML5 Animate CSS DMXzone Cordova Builder
Value
Form
Insert Data Bindings
$URL
$FORM
Data Sources
dsProducts - products
offset
limit
total
link
page
[data]
product_id
product_name
category
status
order
price
description
Repeat Regions
repeat1
Browser Properties
Database Action Executors
Files Properties
Geo Location
Security Provider Executor
Server Connect Action Executors
Bind to: input.data.value
Bind
Behaviors Server Behaviors
Files Snippets

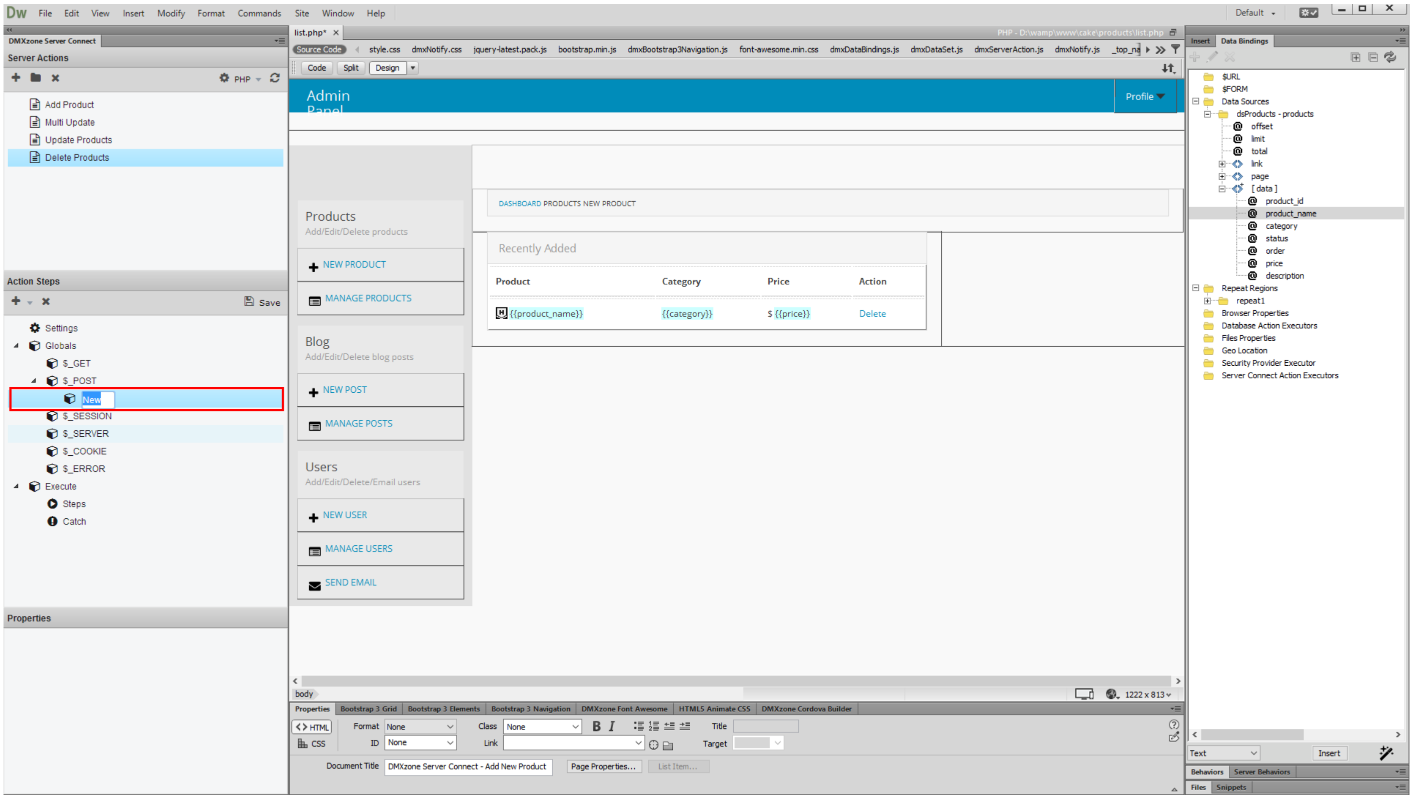
3. Then click **create new server action button (1)** and add a **name for it (2).**



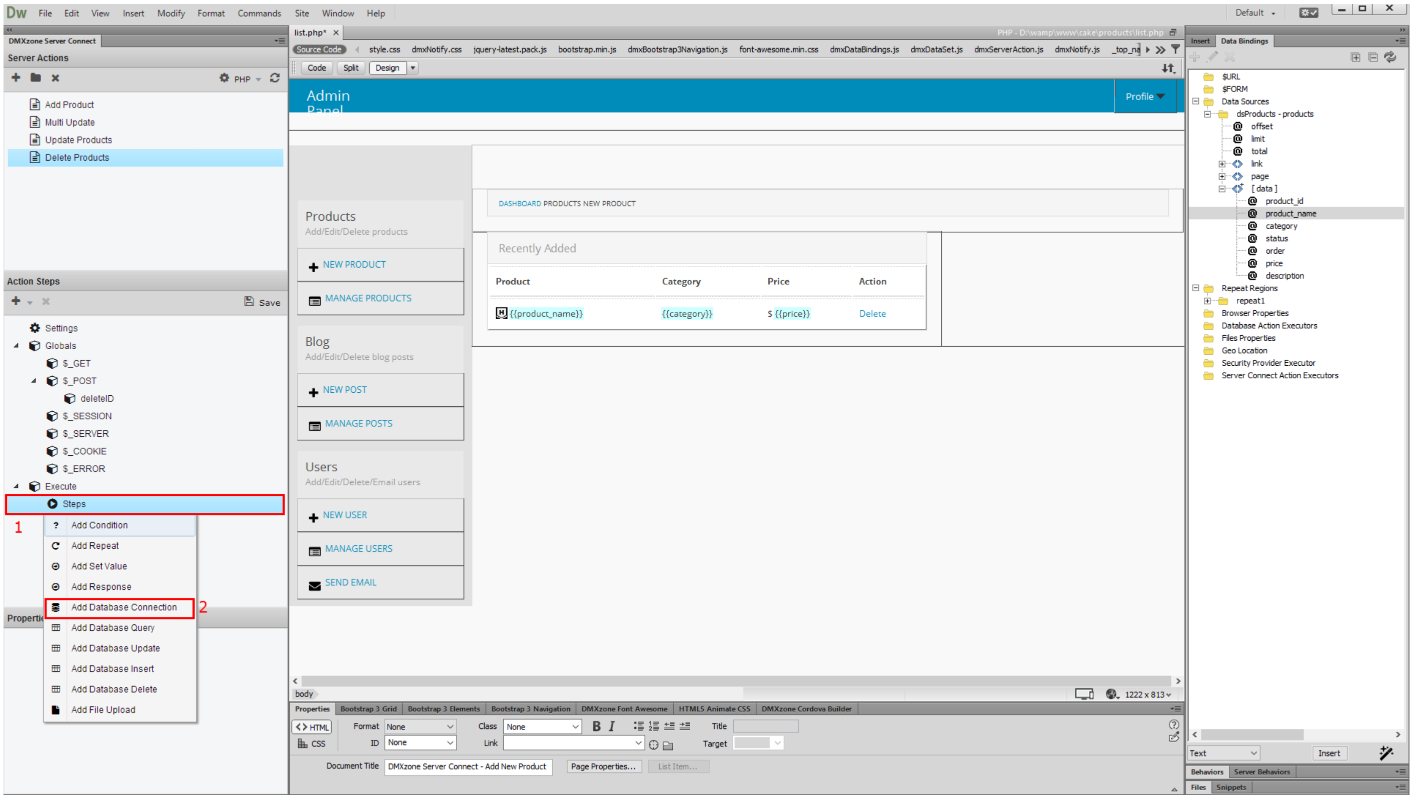
4. Expand the **global variables list (1)**, **right click post (2)** and click **add variable (3)**.



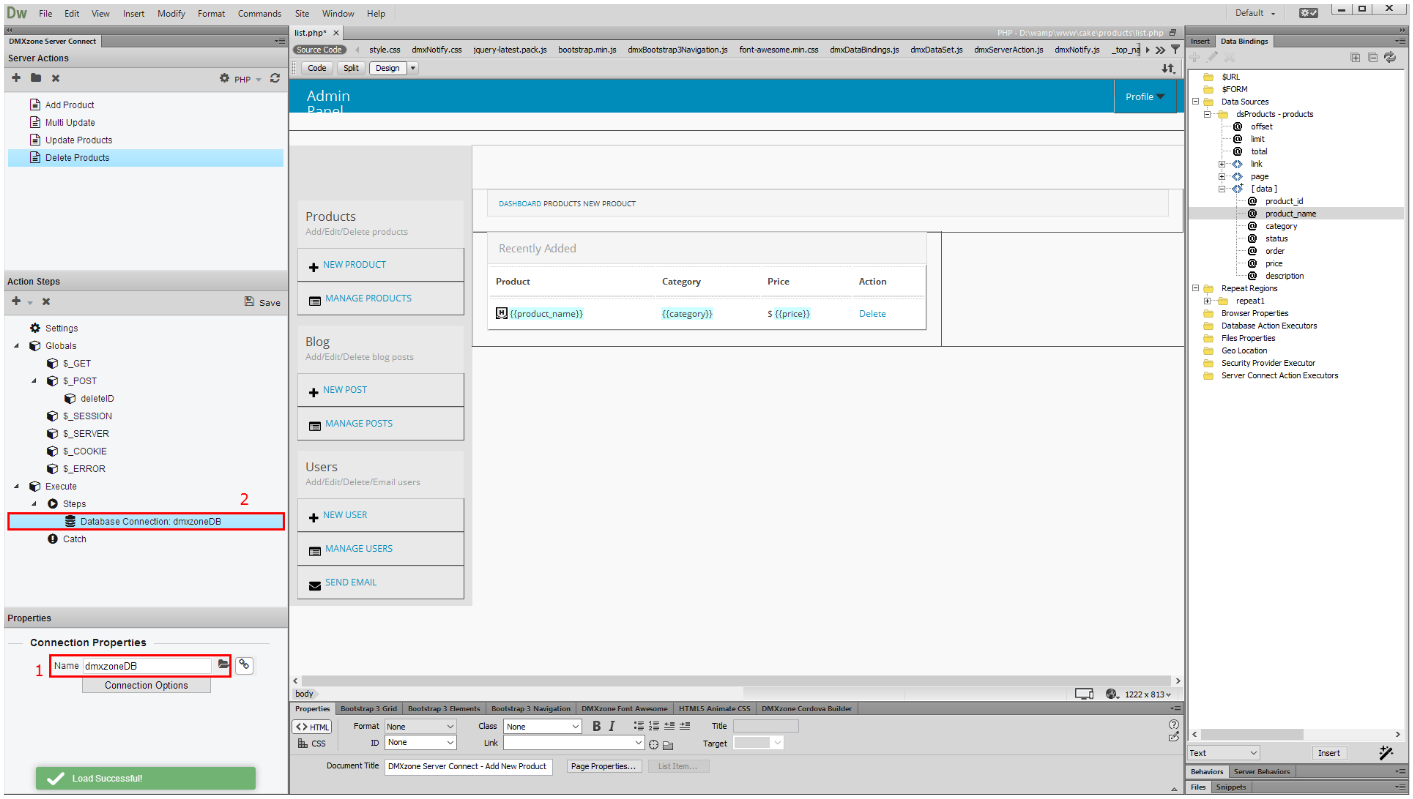
5. Add the **hidden input name** here.



6. Now, under execute, **right click steps (1)** and add a **database connection (2)**.



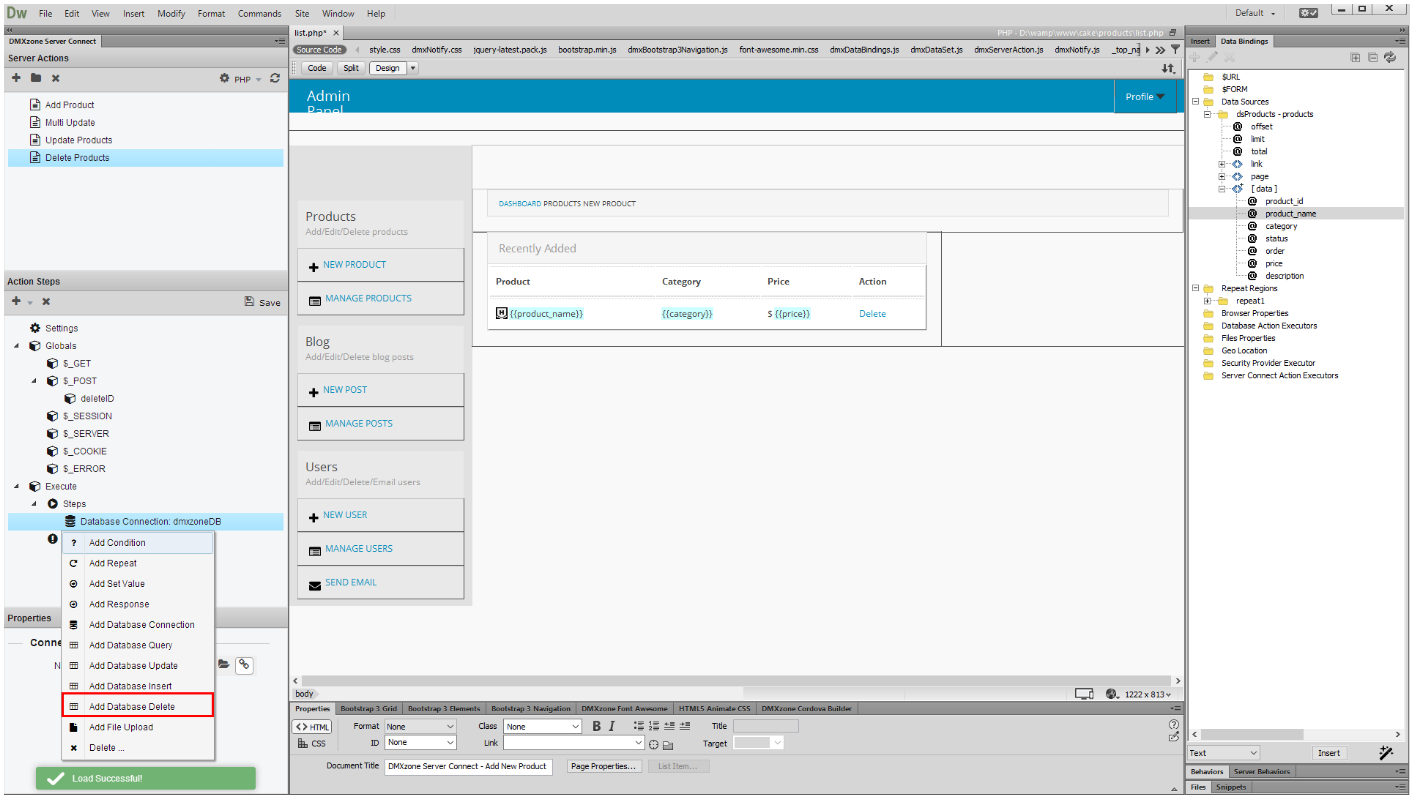
7. Select your **database connection** from the dropdown **(1)** and **right click the connection** in order to add the next step **(2)**.



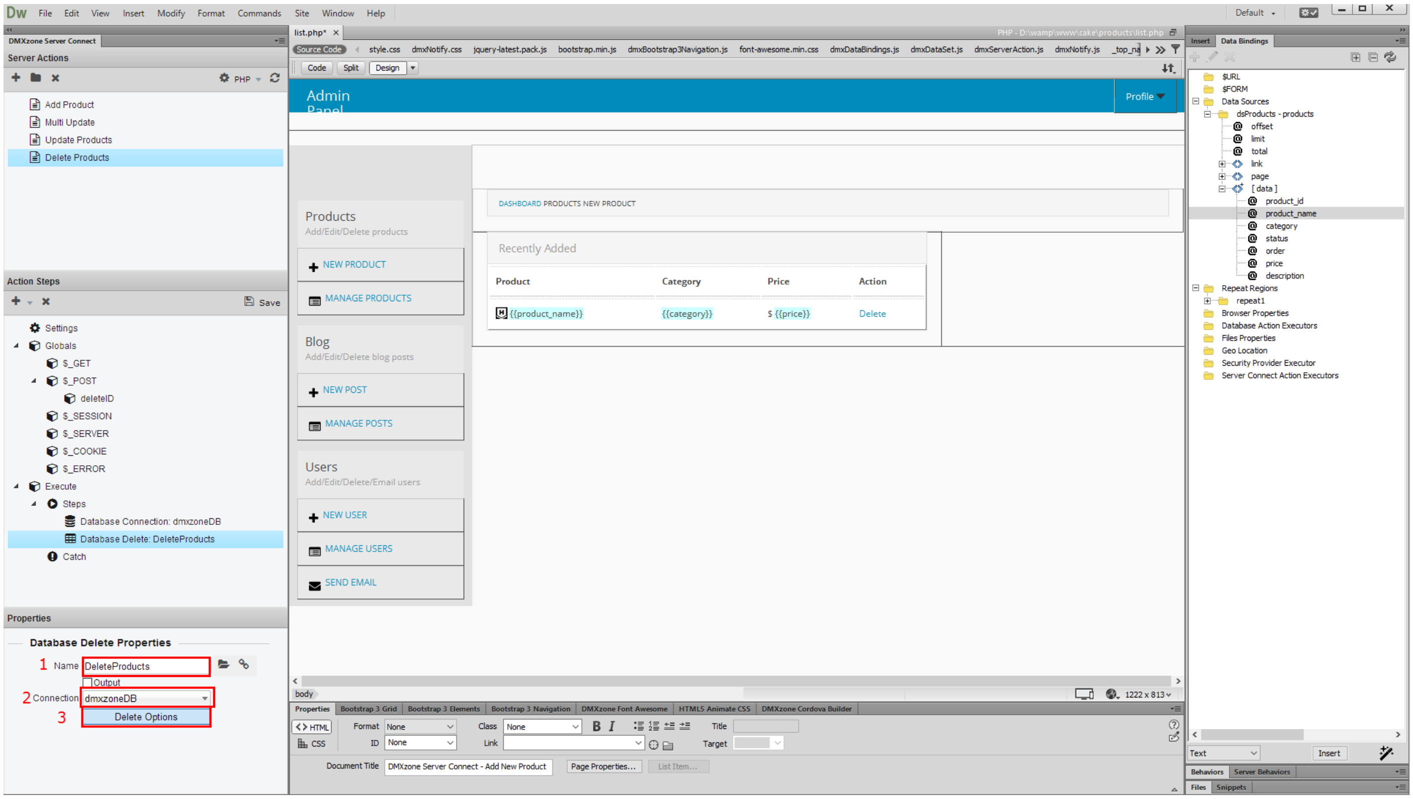
Name dmxzoneDB

Document Title DMXzone Server Connect - Add New Product

8. Our next step is **delete**, so we add it.



9. Add a name for the database delete **(1)** and select your **connection** from the dropdown **(2). ** Click the **delete options button** to set them up **(3)**.



Database Delete Properties

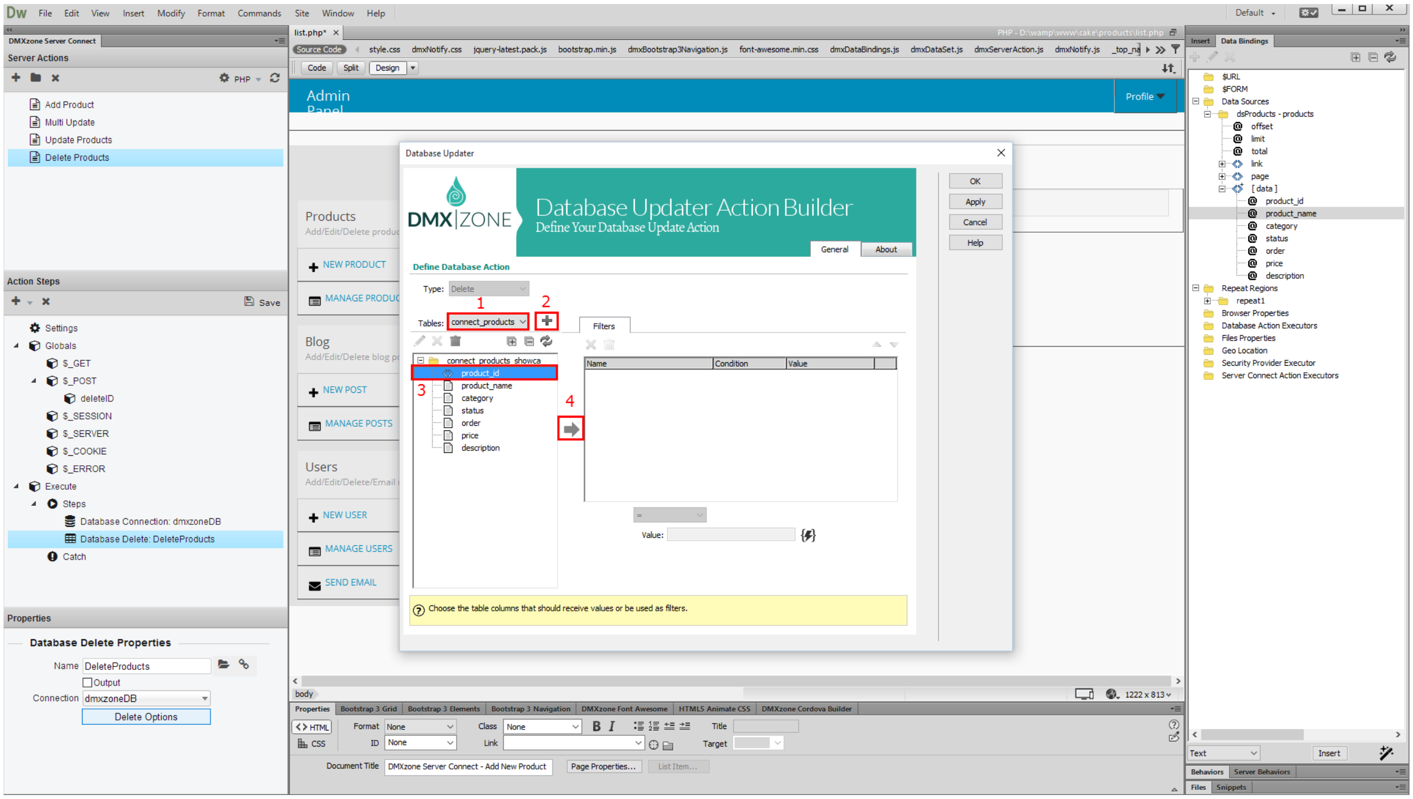
1 Name: DeleteProducts

□ Output

2 Connection: dmxzoneDB

3 Delete Options

10. Select your **database table (1)** and **add it** to the list **(2)**. Choose your **ID column (3)** and **add it** to the filters list **(4)**.



Type: Delete

Tables: connect_products

connect_products_showca
- product_id
- product_name
- category
- status
- order
- price
- description

Value:

Choose the table columns that should receive values or be used as filters.

Database Delete Properties

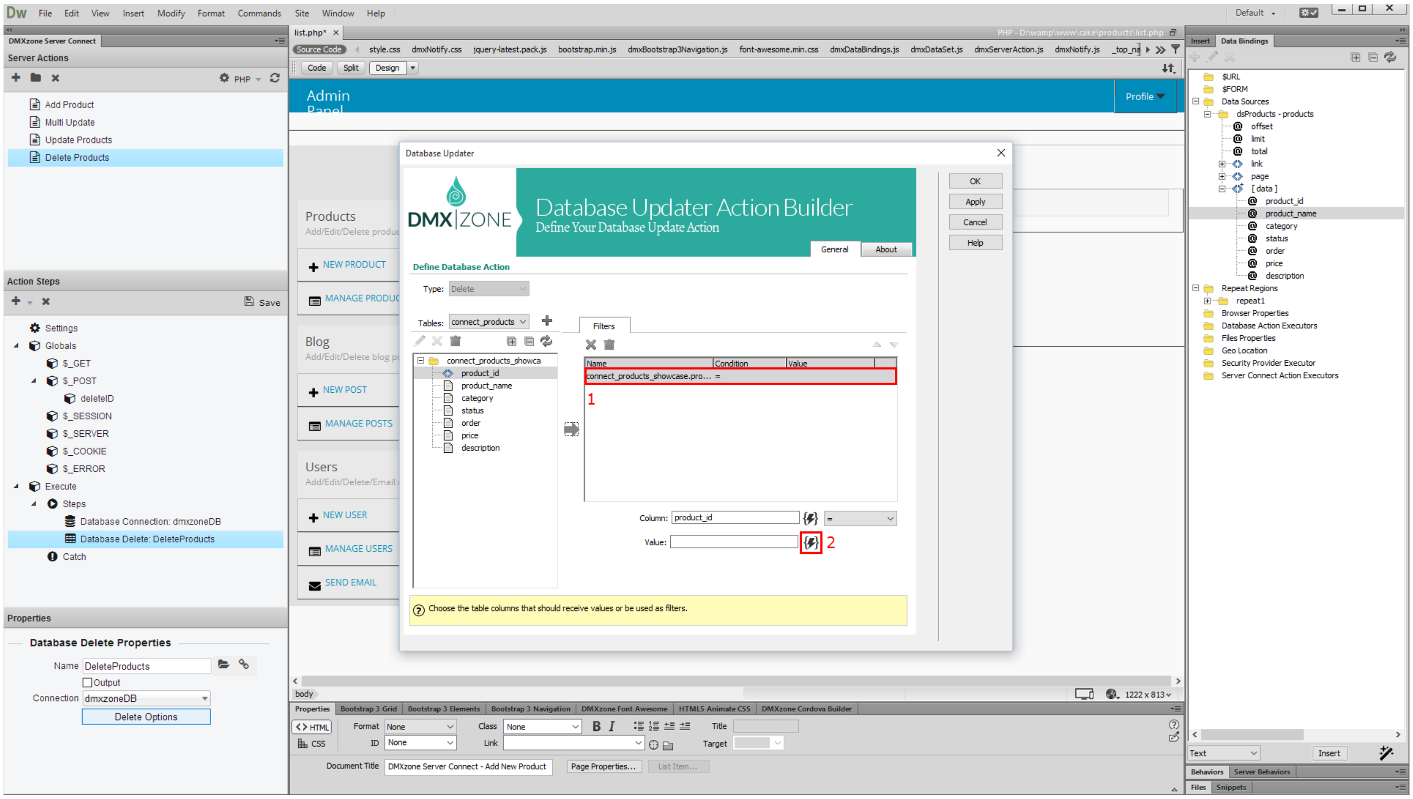
Name: DeleteProducts

□ Output

Connection: dmxzoneDB

Document Title: DMXzone Server Connect - Add New Product

11. Choose the **column (1)** and **click the server data bindings picker** to select a value **(2)**.



Database Updater Action Builder
Define Your Database Update Action

Define Database Action

Type: Delete

Tables: connect_products

Name	Condition	Value
connect_products_showcase.pro...	=	

Column: product_id =

Value:

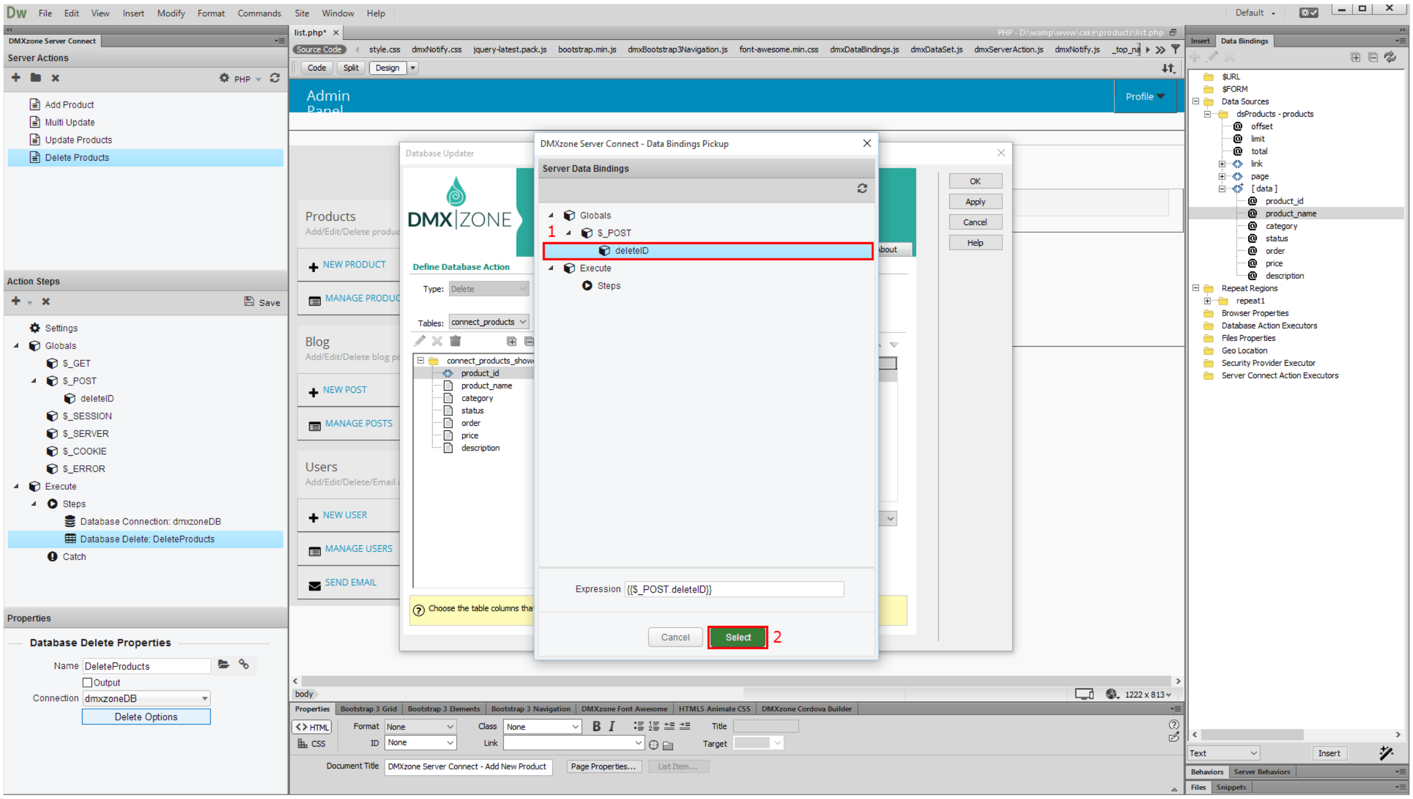
Choose the table columns that should receive values or be used as filters.

Database Delete Properties

Name: DeleteProducts

Connection: dmxzoneDB

12. Select the **hidden field name** that we added earlier as a post variable, under globals **(1)** and **click the select button (2)**.



Expression {{$_POST.deleteID}}

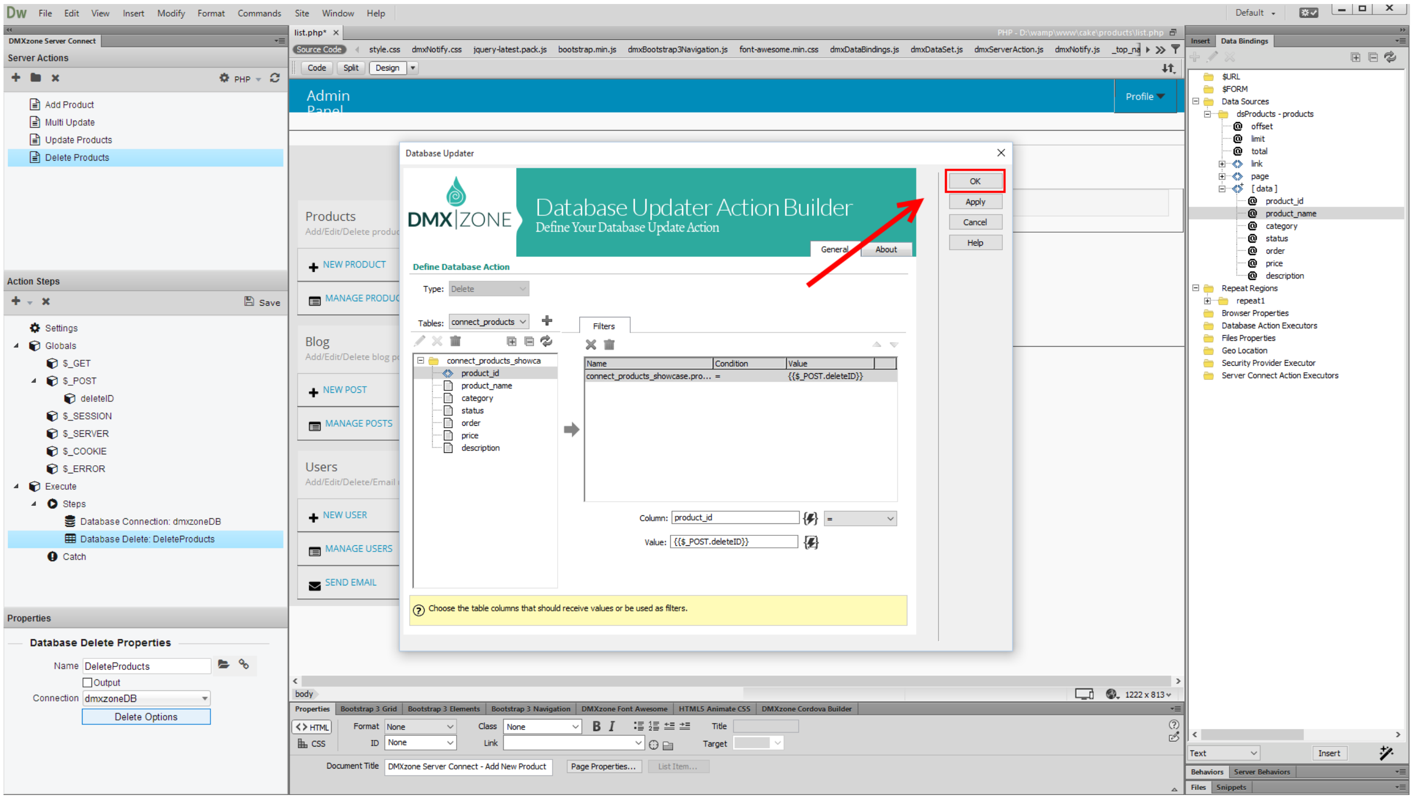
Name DeleteProducts

Connection dmxzoneDB

Type: Delete

Tables: connect_products

13. Click OK.



Database Updater

Database Updater Action Builder
Define Your Database Update Action

Define Database Action

Type: Delete

Tables: connect_products

Filters

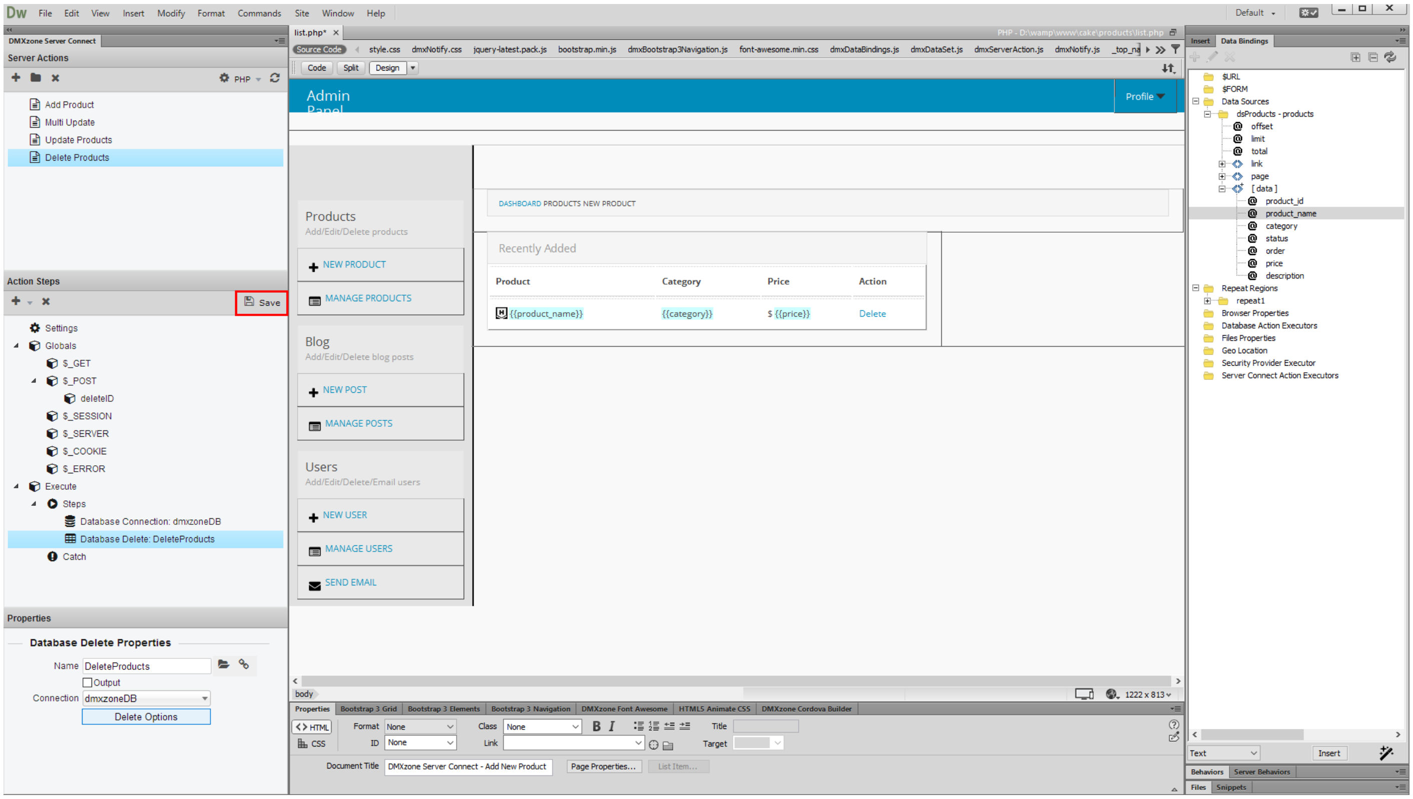
Name	Condition	Value
connect_products_showcase.pro...	=	{{$_POST.deleteID}}

Column: product_id =

Value: {{$_POST.deleteID}}

Choose the table columns that should receive values or be used as filters.

14. Do not forget to **save your action steps**.



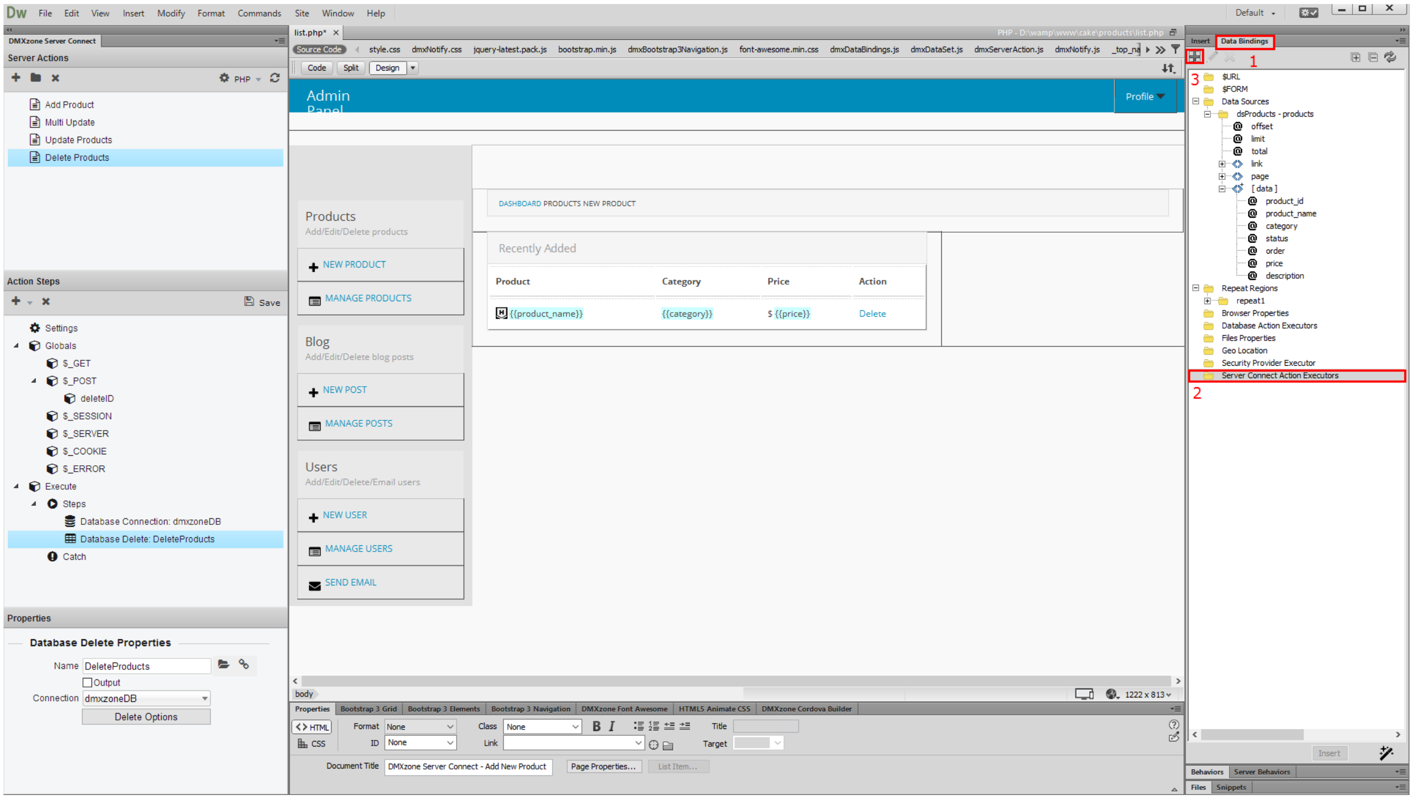
Name: DeleteProducts

☐ Output

Connection: dmxzoneDB

Document Title: DMXzone Server Connect - Add New Product

15. Now, go to **HTML5 Data Bindings panel (1)** and select **server connect action executors (2)**, then click **add new button (3)**.

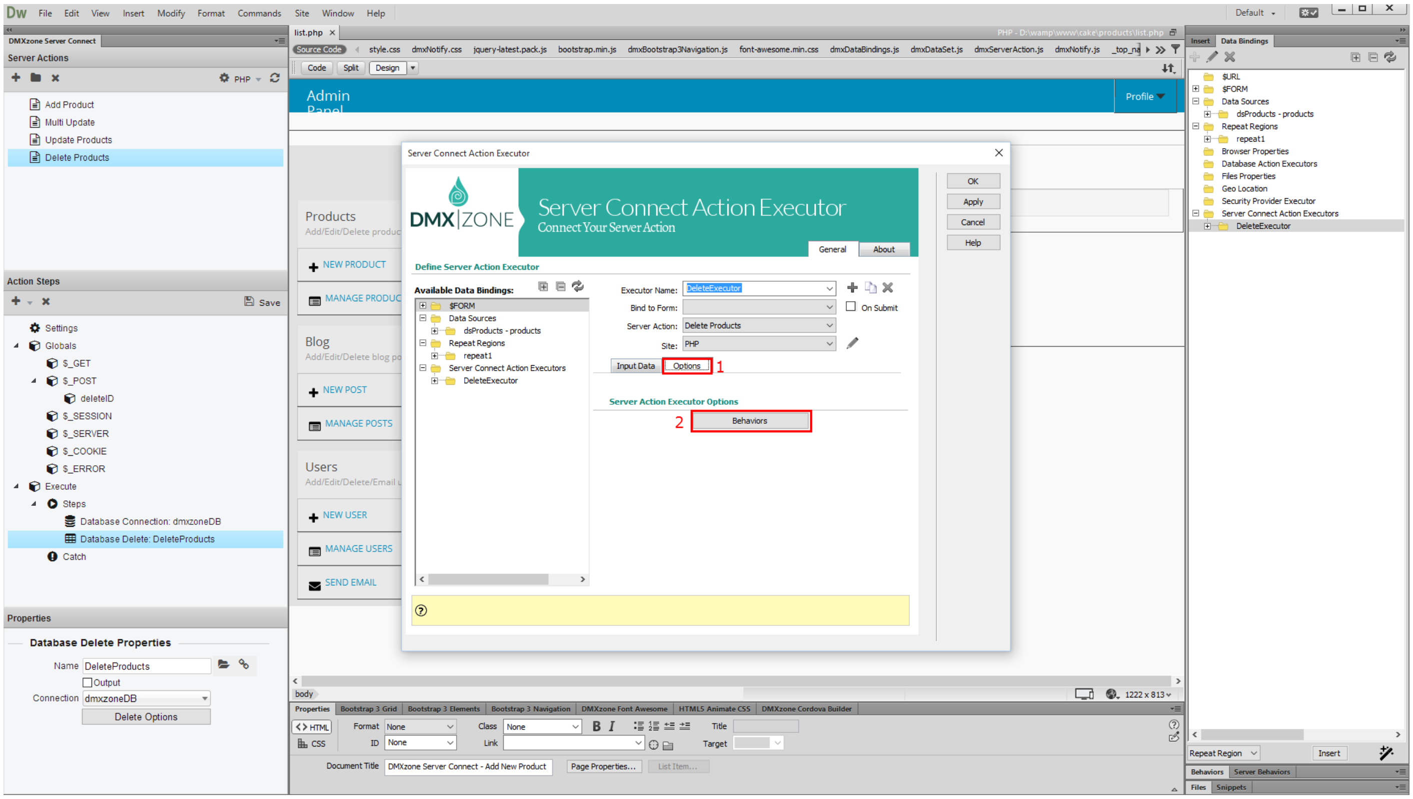


Name DeleteProducts
□Output
Connection dmxzoneDB

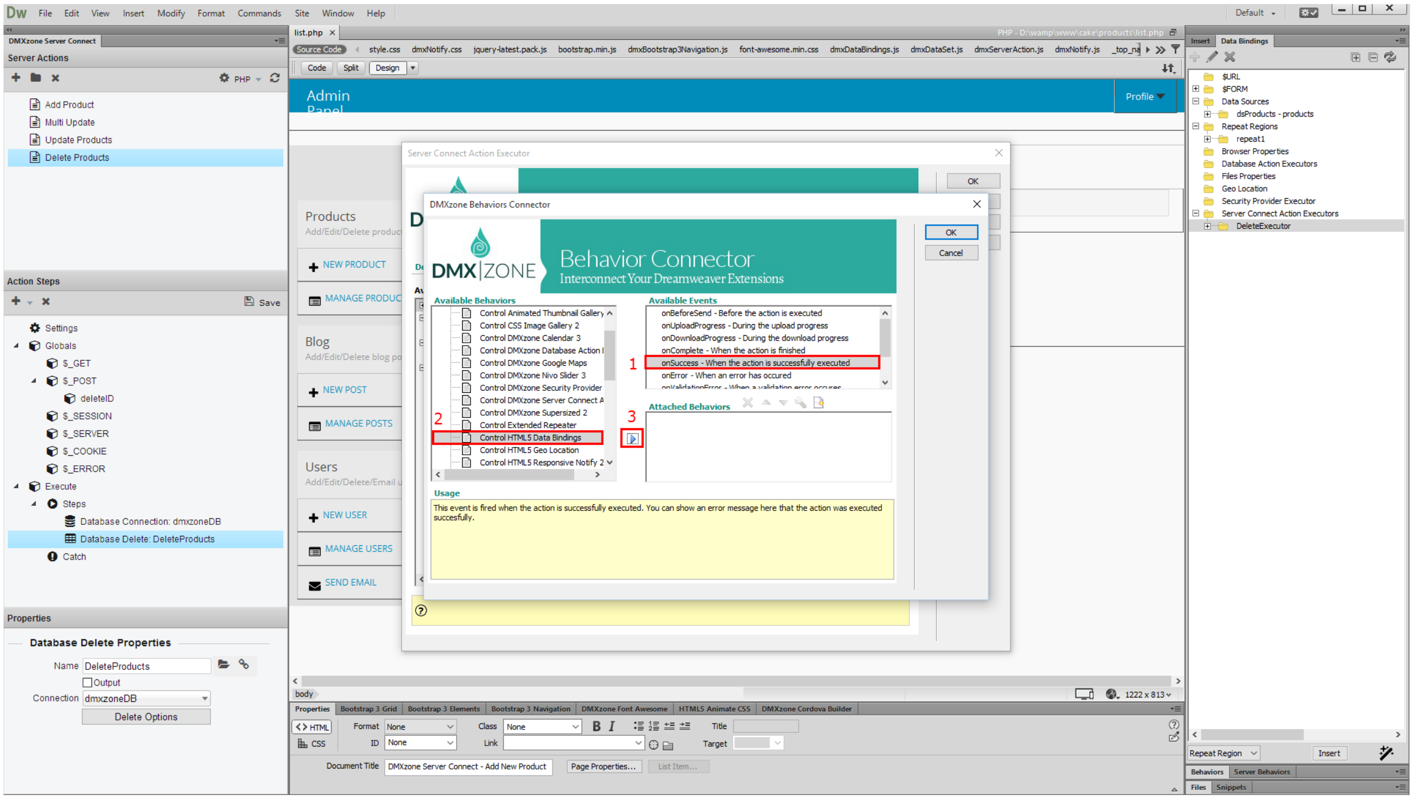
Document Title DMXzone Server Connect - Add New Product

16. Select your **server action (1)** from the dropdown and **add a name for the executor (2)**.

17. In the **options tab (1)**, we're going to add a behavior that refreshes our data source, once a record is deleted. Click the **behaviors button (2)**.

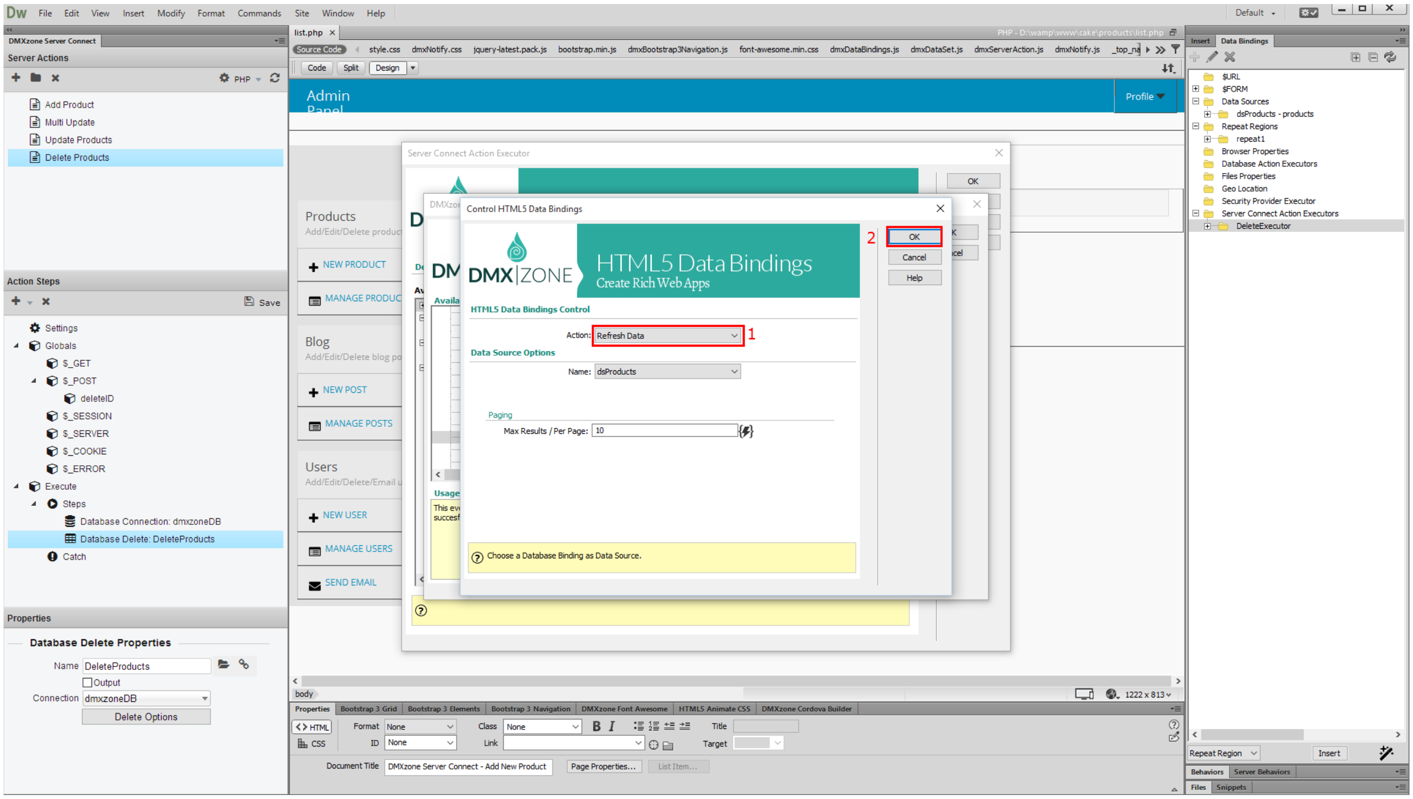


18. Select the **on success event (1)** and then **add (3)** the **control HTML5 Data Bindings behavior (2)** to this event.


Server Connect Action Executor
DMXzone Behaviors Connector
Behavior Connector
Interconnect Your Dreamweaver Extensions
OK
Cancel
Available Behaviors
Control Animated Thumbnail Gallery
Control CSS Image Gallery 2
Control DMXzone Calendar 3
Control DMXzone Database Action I
Control DMXzone Google Maps
Control DMXzone Nivo Slider 3
Control DMXzone Security Provider
Control DMXzone Server Connect A
Control DMXzone Supersized 2
Control Extended Repeater
Control HTML5 Data Bindings
Control HTML5 Geo Location
Control HTML5 Responsive Notify 2
Available Events
onBeforeSend - Before the action is executed
onUploadProgress - During the upload progress
onDownloadProgress - During the download progress
onComplete - When the action is finished
onSuccess - When the action is successfully executed
onError - When an error has occured
Attached Behaviors
Usage
1
2
3
DMXzone Server Connect
Server Actions
Add Product
Multi Update
Update Products
Delete Products
Action Steps
Settings
Globals
$_GET
$_POST
deleteID
$_SESSION
$_SERVER
$_COOKIE
$_ERROR
Execute
Steps
Database Connection: dmxzoneDB
Database Delete: DeleteProducts
Catch
Database Delete Properties
Name DeleteProducts
Output
Connection dmxzoneDB
Delete Options
Data Bindings
$URL
$FORM
Data Sources
dsProducts - products
Repeat Regions
repeat1
Browser Properties
Database Action Executors
Files Properties
Geo Location
Security Provider Executor
Server Connect Action Executors
DeleteExecutor

Usage: This event is fired when the action is successfully executed. You can show an error message here that the action was executed succesfully.

19. The action should be set to **refresh data**. When you're done, **click OK (2)**.



Control HTML5 Data Bindings

HTML5 Data Bindings Control

Action: Refresh Data

Data Source Options

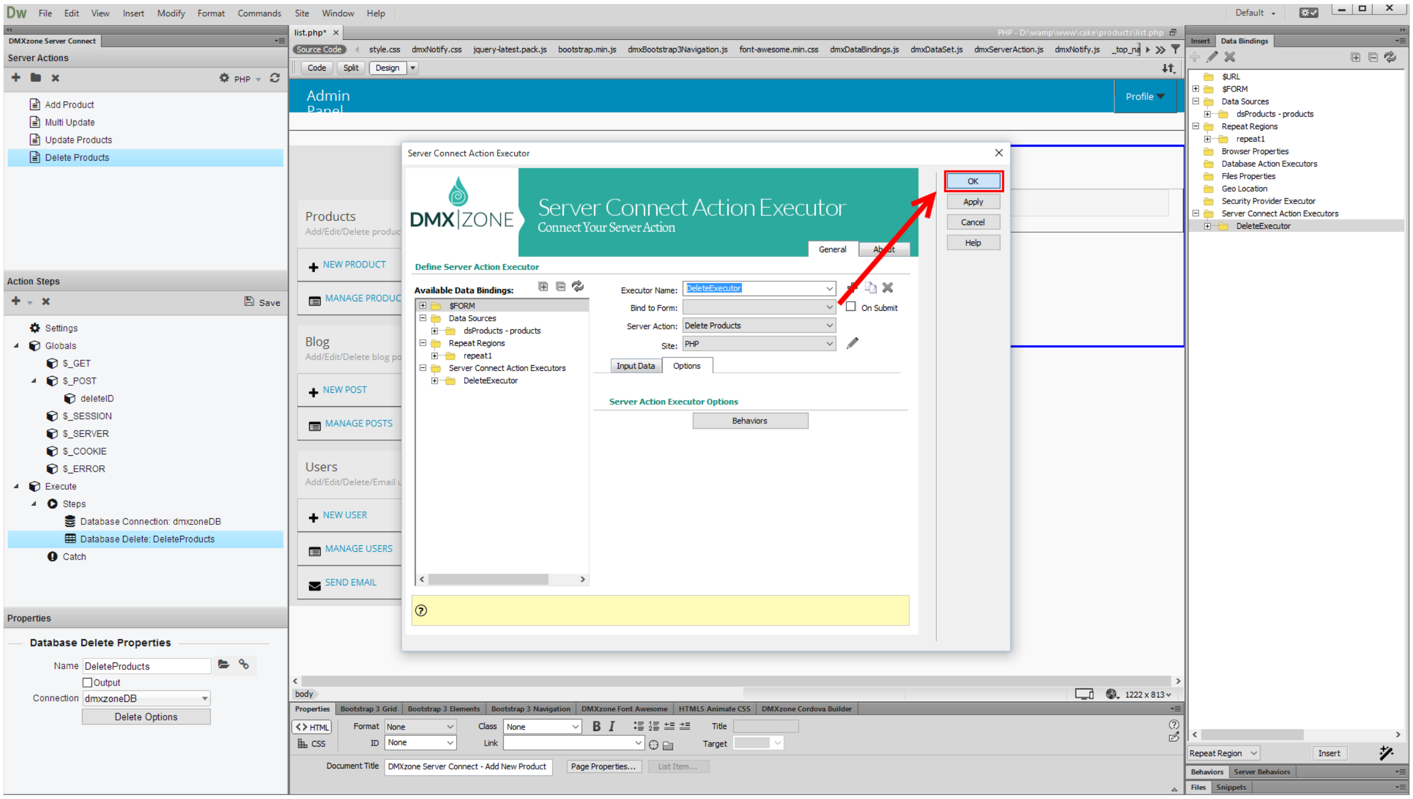
Name: dsProducts

Paging

Max Results / Per Page: 10

Choose a Database Binding as Data Source.

20. And then apply the executor options, by clicking the OK button.



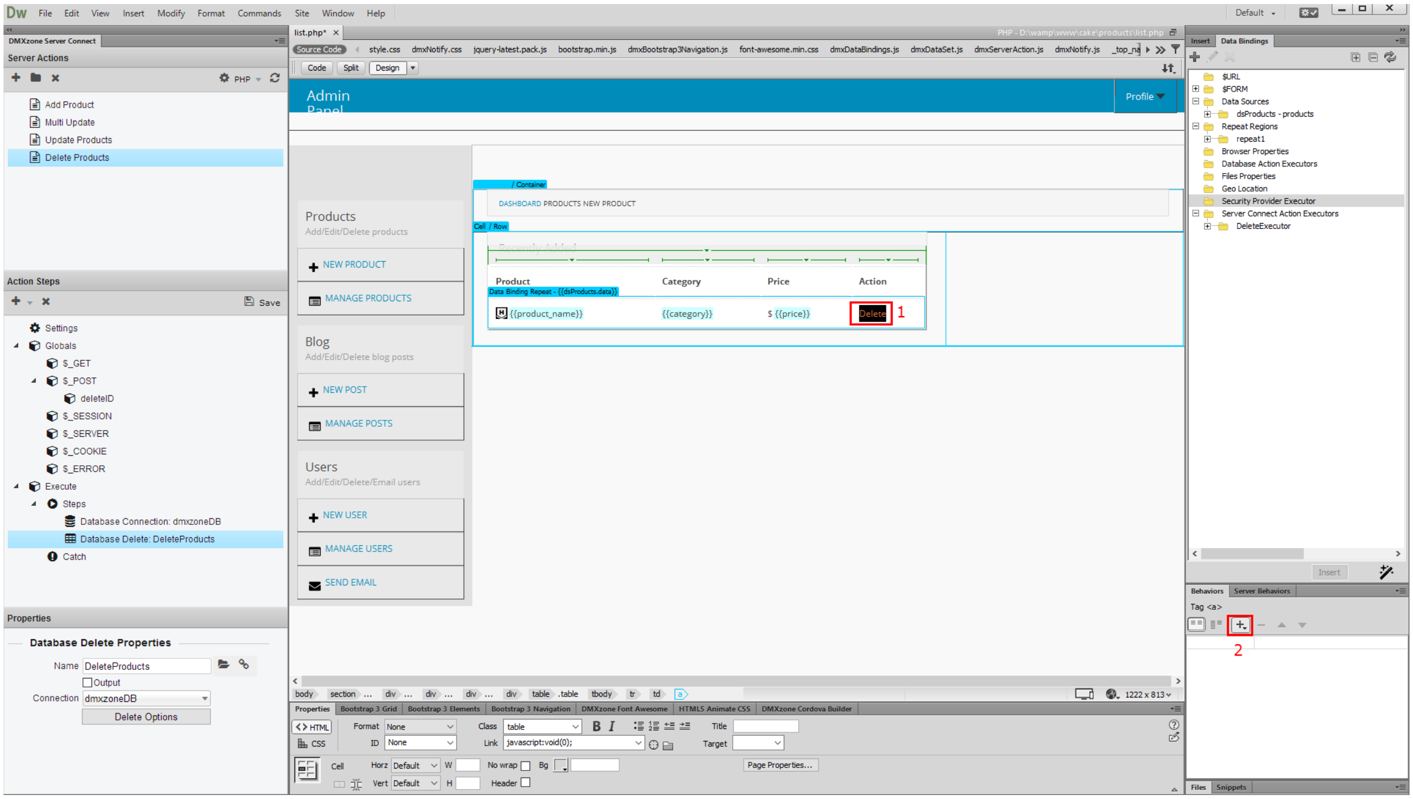
Server Connect Action Executor

Define Server Action Executor

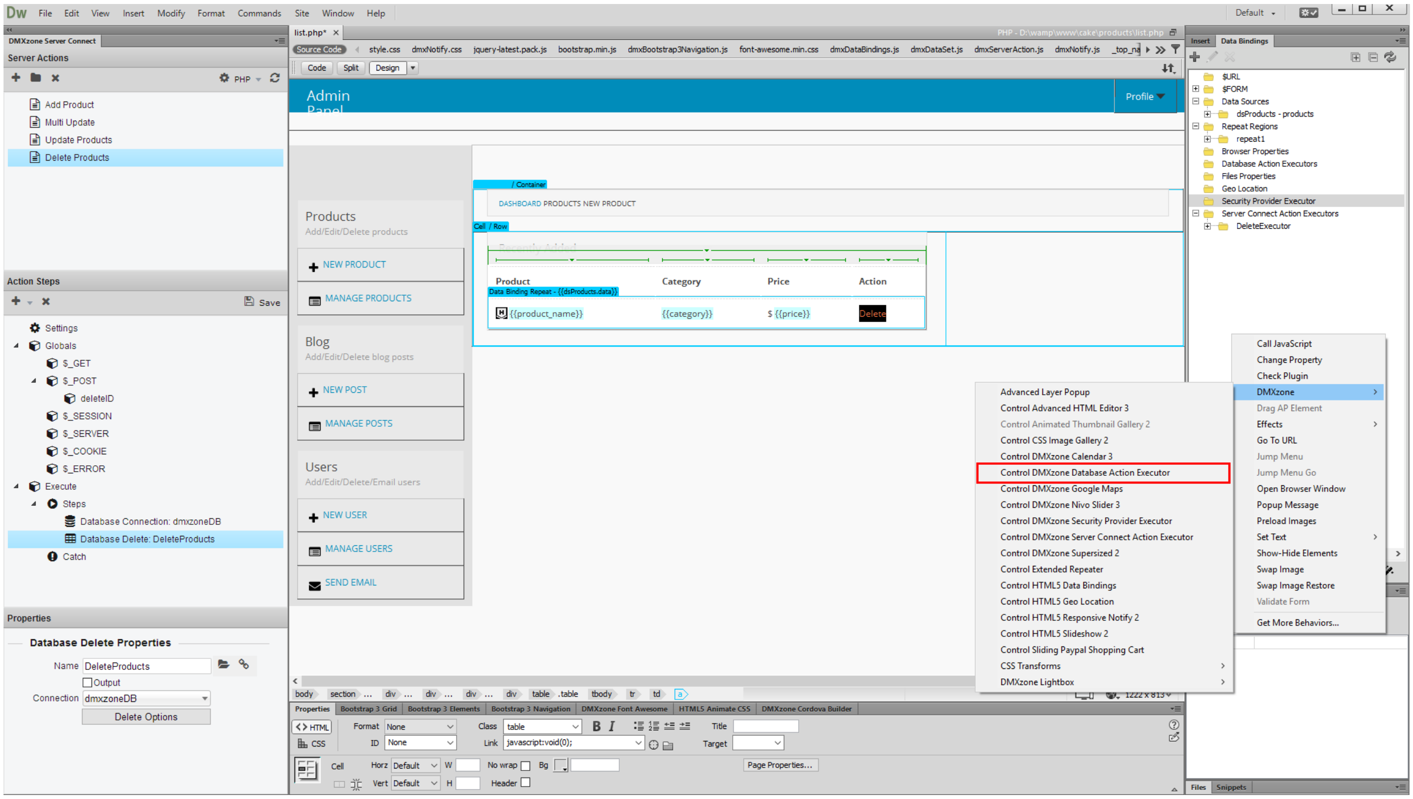
Executor Name: DeleteExecutor
Bind to Form:
□ On Submit
Server Action: Delete Products
Site: PHP

Server Action Executor Options

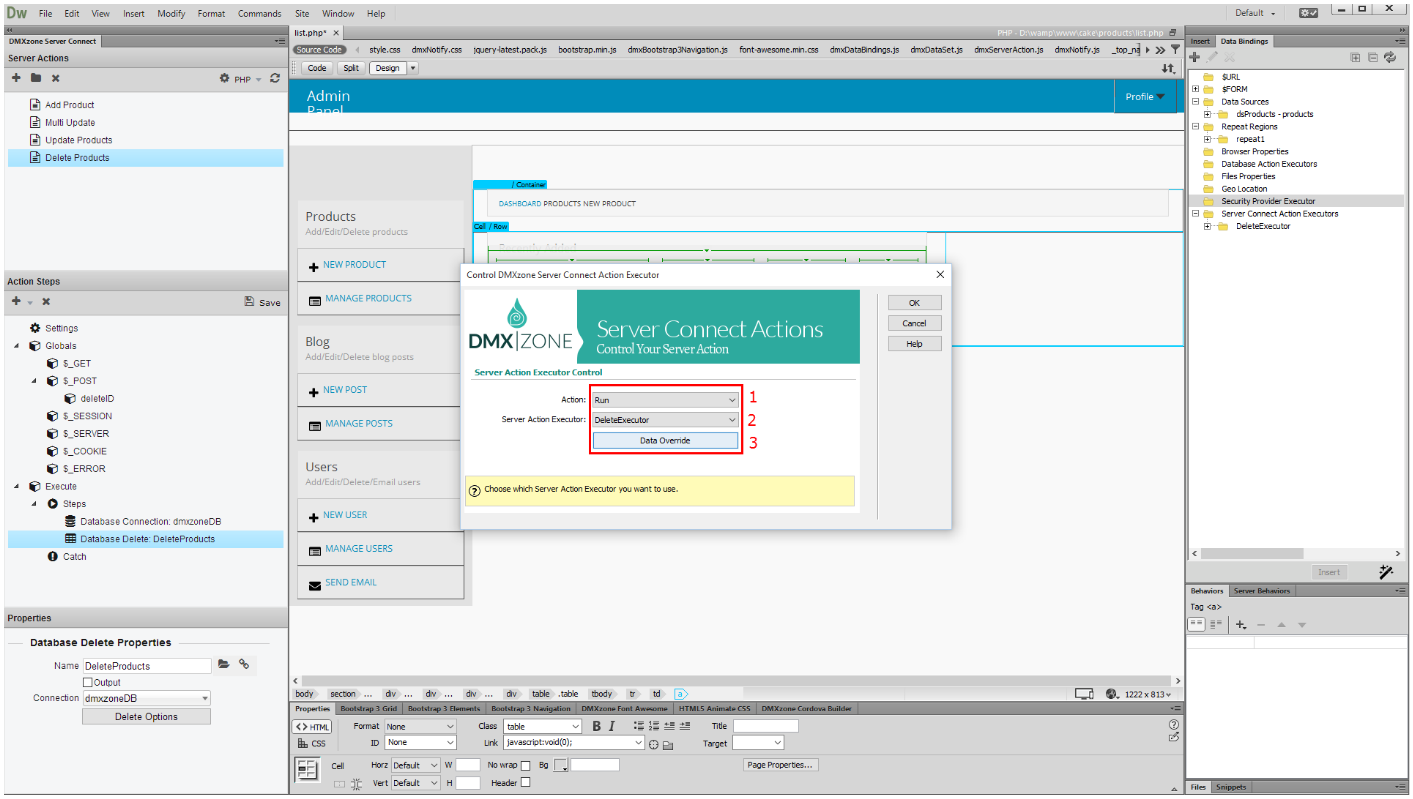
21. Now, let’s bind the delete action executor to the delete link. Select the **dummy delete link (1)** and **add a new behavior (2)**.



22. Select **Control DMXzone Server Connect Action Executor**, located in the DMXzone sub menu.



23. Select **run (1)** and choose your **delete action executor (2)** from the dropdown. Click the **data override button (3)**.



Control DMXzone Server Connect Action Executor
Server Connect Actions
Control Your Server Action
OK
Cancel
Help
1
2
3

Server Action Executor Control

Action: Run

Server Action Executor: DeleteExecutor

Data Override

Choose which Server Action Executor you want to use.

Database Delete Properties

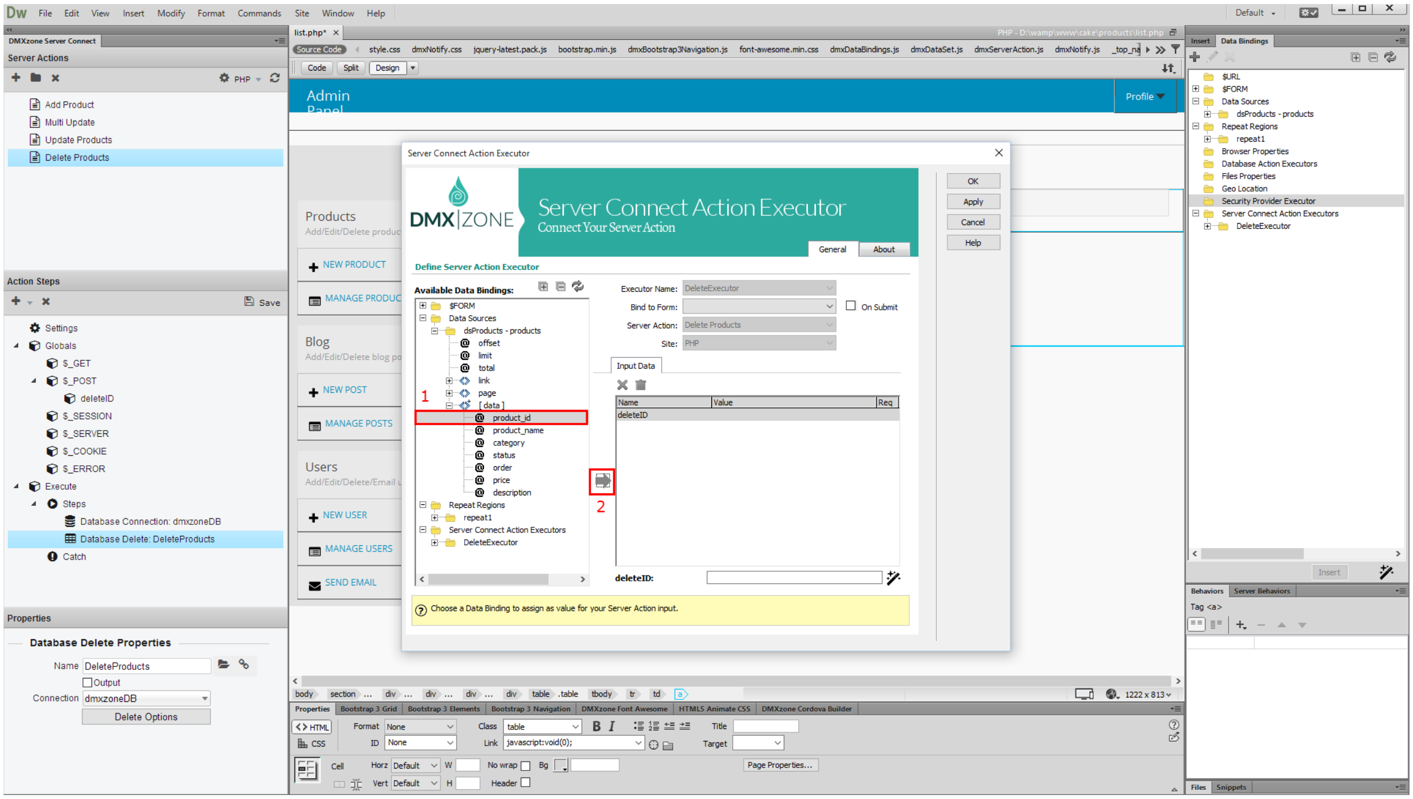
Name: DeleteProducts

□ Output

Connection: dmxzoneDB

Delete Options

24. We need to bind a value to the hidden field that we use to filter the delete action. So, just select the **ID binding (1)** of your data source and **bind it as a value** for the hidden field **(2)**. Then click OK.



Server Connect Action Executor

Server Connect Action Executor
Connect Your Server Action

Define Server Action Executor

Available Data Bindings:

Executor Name: DeleteExecutor
Bind to Form:
☐ On Submit
Server Action: Delete Products
Site: PHP

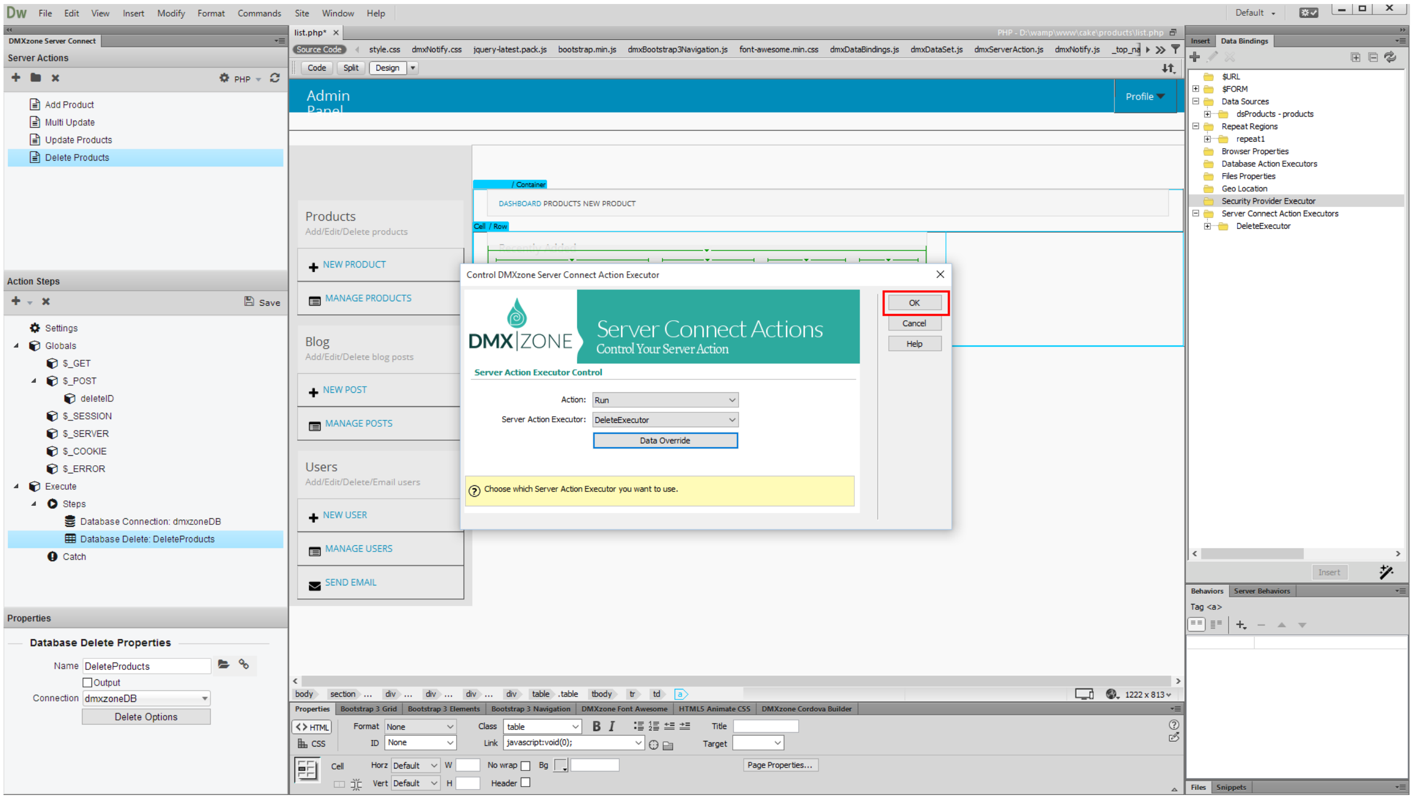
Input Data

Name	Value	Req
deleteID		

deleteID:

Choose a Data Binding to assign as value for your Server Action input.

25. And **apply the behavior**.



Control DMXzone Server Connect Action Executor

Server Connect Actions
Control Your Server Action

Server Action Executor Control

Action: Run

Server Action Executor: DeleteExecutor

Data Override

Choose which Server Action Executor you want to use.

OK
Cancel
Help

Database Delete Properties

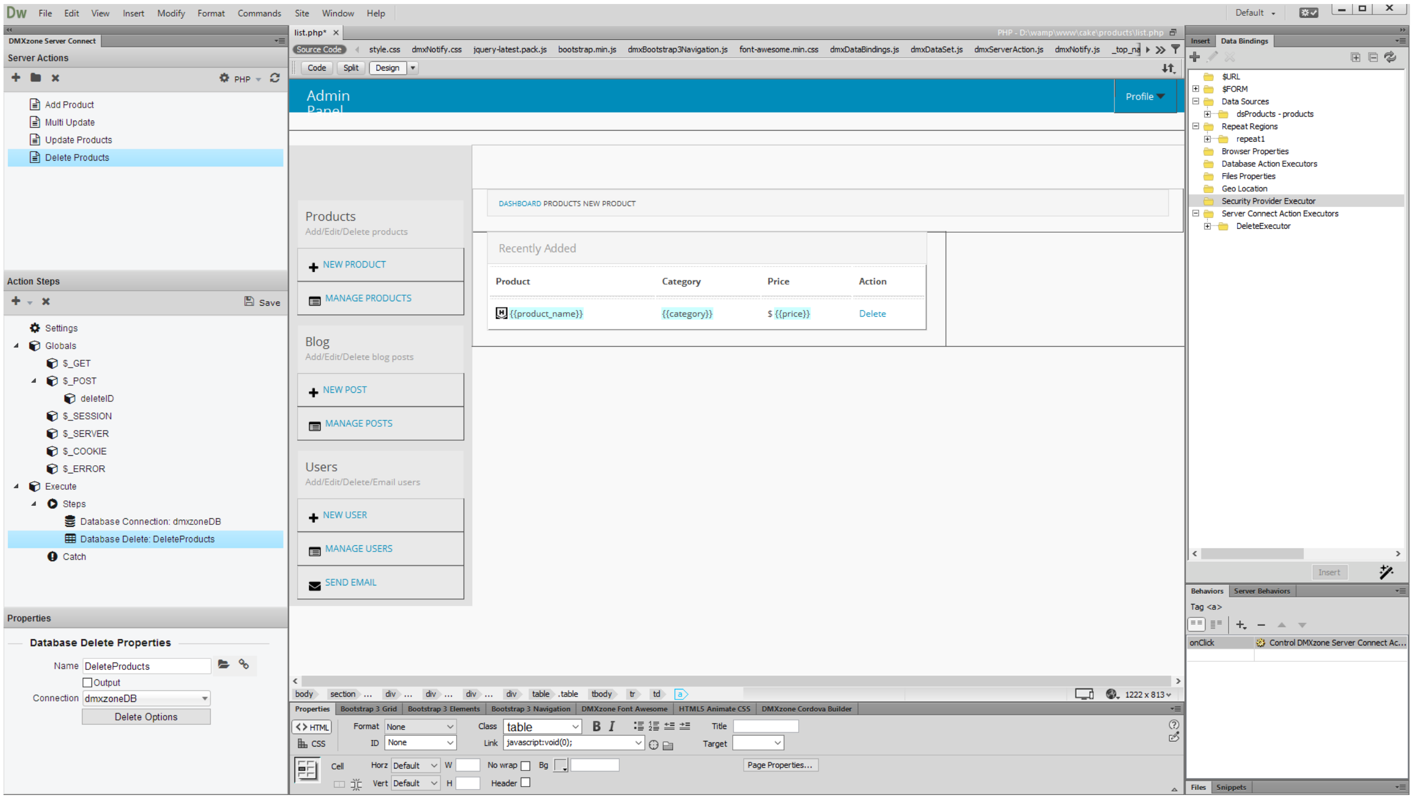
Name: DeleteProducts

□ Output

Connection: dmxzoneDB

Delete Options

26. Let's preview the results!



Product	Category	Price	Action
{{product_name}}	{{category}}	$ {{price}}	Delete

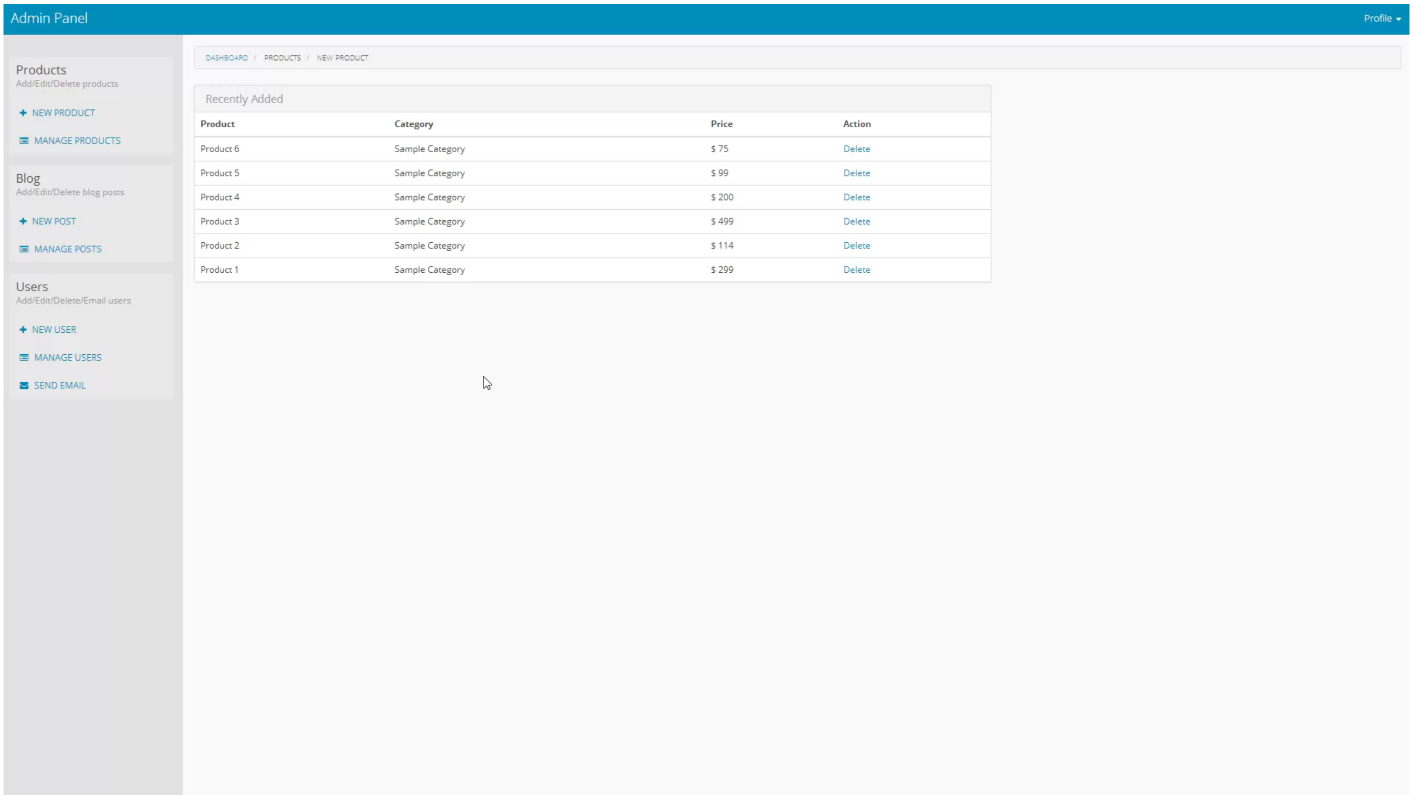
Database Delete Properties

Name DeleteProducts

□ Output

Connection dmxzoneDB

27. As soon as you click the delete button, the action is executed and the data source refreshes. That’s how easy it is to delete database records using DMXzone Server Connect.



Recently Added

Product	Category	Price	Action
Product 6	Sample Category	$ 75	Delete
Product 5	Sample Category	$ 99	Delete
Product 4	Sample Category	$ 200	Delete
Product 3	Sample Category	$ 499	Delete
Product 2	Sample Category	$ 114	Delete
Product 1	Sample Category	$ 299	Delete

Video: Accessing Server Actions Builder Panel

With this movie we will show you how to access and enable DMXzone Server Connect in Dreamweaver.

Video: Adding Form Inputs to DMXzone Server Connect

With this movie we will show you how to add your form fields to the global input variables of DMXzone Server Connect.

Video: Insert Records into a Database

With this movie we will show you how to how to insert records in your database, using DMXzone Server Connect. Note, that the insert record functionality requires the HTML5 Data Bindings, DMXzone Database Connector PHP/ASP and DMXzone Database Updater PHP/ASP extensions.

Video: Update Database Records

With this movie we will show you how to update database records, using DMXzone Server Connect. In order to do that you'll also need HTML5 Data Bindings, DMXzone Database Updater PHP/ASP and DMXzone Database Connector PHP/ASP extensions.

Video: Update Multiple Database Records

With this movie we will show you how to update multiple database records, using DMXzone Server Connect. In order to do that you'll also need HTML5 Data Bindings, DMXzone Database Updater PHP/ASP and DMXzone Database Connector PHP/ASP extensions.

Video: Delete Database Records

With this movie we will show you how to delete multiple database records, using DMXzone Server Connect. In order to do that you'll also need HTML5 Data Bindings, DMXzone Database Updater PHP/ASP and DMXzone Database Connector PHP/ASP extensions.