

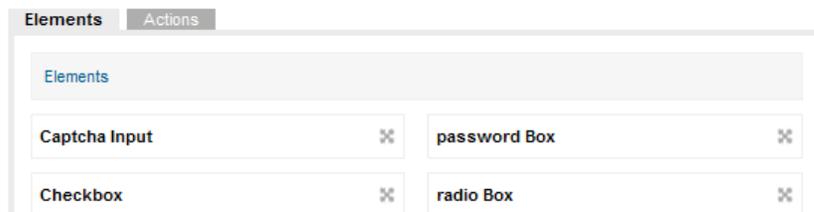
How to list table multiple rows and edit their data

In the last tutorial we have created a form which can load the data of one table record, now we are going to create another form to list all records and give us the ability to edit any one of them:

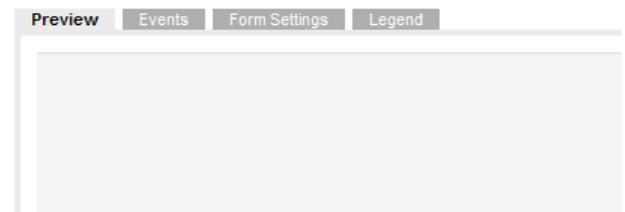
1. Create a new form using the form wizard.

New Form...

Drag

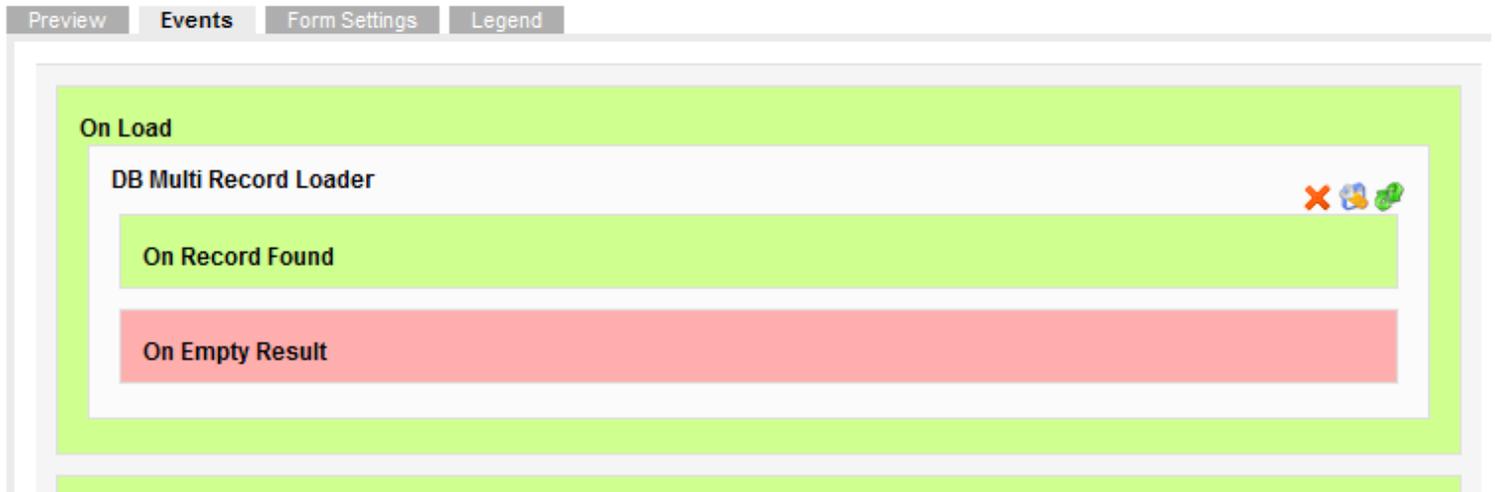


Drop



2. Drag a “DB Multi record loader” action to the “on Load” event, this action should help us load multiple rows data from any db table.

Drop



3. Open the new action settings and configure it as shown (there are many more advanced settings, please ignore them because we don't need them here) then apply and close.

Basic | **Advanced** | **Help**

DB Field

The field name which will be used to query the table record, if left empty then all records will be loaded.

Table

The table name to load the data from.

Request Param

The param name which will exist in the request url to the form, its value will be used to load the target db record.

Model ID

*The key under which the loaded record data will be stored in the form->data array.
this is obligatory, if left empty then a camilized version of the table name will be used, e.g: jos_my_table = JosMyTable*

4. To account for “no data” found cases (in case of empty table or empty results query) we are going to drag a “Custom Code” action and a “Show Stopper” to the “on Empty result” event, you can write in the “Custom Code” action “Empty table” or whatever you want.

Drop

Preview Events Form Settings Legend

On Load

DB Multi Record Loader ✕ 🔄 🟢

On Record Found

On Empty Result

Custom Code - ✕ 🔄 🟢

Show Stopper ✕ 🔄 🟢

Settings Help

Mode ▼

When should this code run ? during the controller code processing (early) or later when the output is viewed.

Action Label

Label for your action in the wizard.

Code

```
Empty table!!!!
```

5. It's supposed that our form is able to load the data from the table by now, so lets drag a "Custom Code" action to display this data, I will show you how to do this.



The screenshot shows the ChronoEngine form builder interface. At the top, there are four tabs: "Preview", "Events", "Form Settings", and "Legend". The "Events" tab is selected. Below the tabs, there is a large green-bordered area representing the form's event actions. Inside this area, there is a section titled "On Load" which contains a "DB Multi Record Loader" action. Below this, there is a section titled "On Record Found" which is currently empty. Below that, there is a section titled "On Empty Result" which contains two actions: "Custom Code -" and "Show Stopper". Each of these actions has a red 'X' icon, a blue refresh icon, and a green checkmark icon. At the bottom of the green-bordered area, there is another "Custom Code -" action, also with the same three icons.

6. We will use a mix of PHP and HTML code to loop through the records loaded and display them, since we have used "**UserDetails**" as our Model ID in the "multi record loader", all loaded records should be stored under that key inside the `$form->data` variable, as you can see, I'm using a simple foreach loop to display the results in a simply formatted HTML table.

Mode

When should this code run ? during the controller code processing (early) or later when the output is viewed.

Action Label

Label for your action in the wizard.

Code

```
<table>
<?php
foreach($form->data['UserDetails'] as $detail):
?>
<tr>
<td>Profession:<?php echo $detail['profession']; ?></td>
<td><a href="index.php?option=com_chronoforms&chronoform=user_details&
token=<?php echo $detail['cf_uid']; ?>">Edit</a></td>
</tr>
<?php
endforeach;
?>
</table>
```

- I'm displaying a table row (<tr>) for each record, and each table row has 2 cells (<td>), one of them has the profession record and the other has a link to the "user_details" form we created in the previous tutorial, at the end of the link I'm attaching the **token** parameter which should load the correct record data (go back to the "db record loader" tutorial for any details about this).
- Cool, lets apply then close the box, go to the "form settings" tab, add a form name and publish it then save and close.

Preview Events **Form Settings** Legend

Form name

Unique form name without spaces or any special characters, underscores _ or dashes -

Published

9. By displaying our newly created form, we can see a list of all table records.

Profession:Developer	Edit
Profession:Designer	Edit
Profession:Manager	Edit
Profession:Director	Edit
Profession:Sales Executive	Edit
Profession:System Administrator	Edit

10. Clicking on the “Edit” link of any record will simply load the “user_details” form and load that record data into it, very nice.

Profession

Subscribe?

Phone

11. So far so good, but you shouldn't try to edit and save your record now, why ? Because our "user_details" form was not ready to "update" records, it was only created to "view" existing records or "create" new ones, lets give it the ability to update them as well.
12. Open the "user_details" form for "wizard edit".
13. Drag a hidden field box to the end of the form, give it the name "cf_id", why this name ? Because its the name of the "primary key" field in the table we are loading the data from/updating.

Drop

Preview Events Form Settings Legend

Profession	<input type="text"/>	  
Subscribe?	<input type="text" value="No"/>	  
Phone	<input type="text"/>	  
Hidden field here		  
<input type="button" value="Submit"/>		  

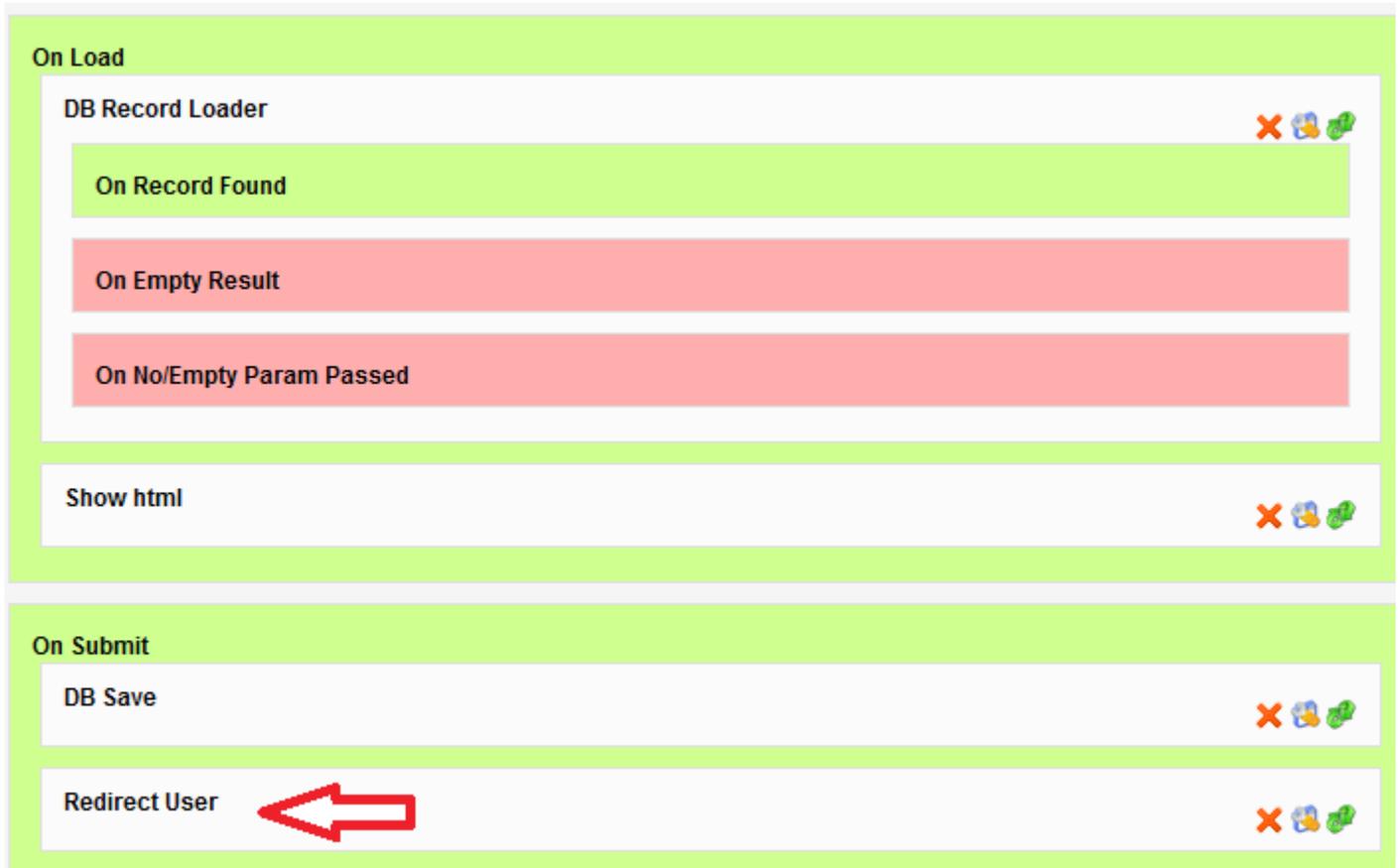
Field Name

cf_id

Field id

Field default value

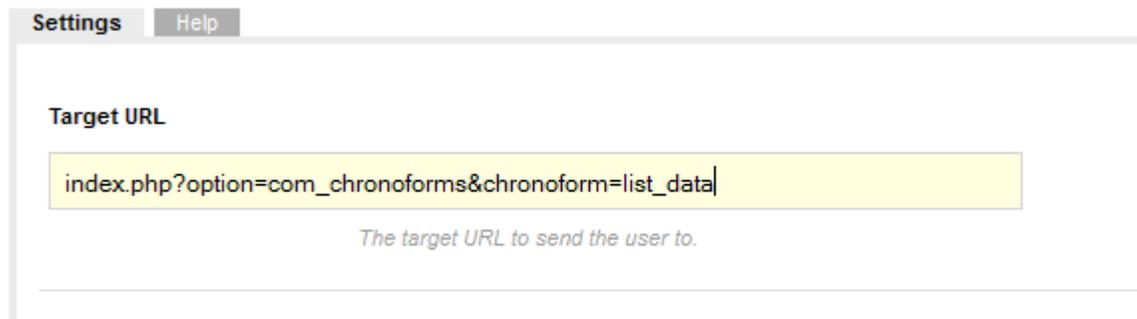
14. Then go to the “Events” tab and drag a “Redirect User” action to the bottom of the “on Submit” event, why ? So that when the user clicks the submit button, the record data will get updated then the user is taken back to the “list_data” form ;)



The screenshot shows the ChronoEngine event configuration interface. It is divided into two main sections: "On Load" and "On Submit".

- On Load:** Contains three stacked actions:
 - DB Record Loader:** A white box with a red 'X', a blue refresh icon, and a green checkmark icon.
 - On Record Found:** A light green box.
 - On Empty Result:** A light red box.
 - On No/Empty Param Passed:** A light red box.
- On Submit:** Contains two stacked actions:
 - DB Save:** A white box with a red 'X', a blue refresh icon, and a green checkmark icon.
 - Redirect User:** A white box with a red 'X', a blue refresh icon, and a green checkmark icon. A red arrow points to this action from the left.

15. We need to configure the “Redirect User” action, open the settings box and enter the url for the “list_data” form and hit apply then close.



The screenshot shows the settings dialog for the "Redirect User" action. It has a title bar with "Settings" and "Help" buttons. The main content area is titled "Target URL" and contains a text input field with the following URL: `index.php?option=com_chronoforms&chronoform=list_data`. Below the input field, there is a small italicized text label: "The target URL to send the user to."

16. Now lets view our listing form again and click “Edit” for any record then try to edit the profession field and click submit, we should be taken back to the “list_data” form and we should be able to see the



new updated data in the list.

Profession	<input type="text" value="Media Director"/>
Subscribe?	<input type="text" value="Yes"/> ▼
Phone	<input type="text" value="111-222-333"/>
<input type="button" value="Submit"/>	

Profession:Developer	Edit
Profession:Designer	Edit
Profession:Manager	Edit
Profession:Media Director	Edit
Profession:Sales Executive	Edit
Profession:System Administrator	Edit

17. I hope you enjoyed! :)