**Contents**

The purpose of this document is to provide a help guide for making changes to either of the radio application forms and the steps that must be taken to ensure that those changes are carried out throughout the project code.

**1. Links for More Information**

**2. Adding a Field to a Form**

* **Update the database**
* **Update the form and view files**
* **Update the confirmation page**
* **Update the RadioApplication controller file**
* **Update the Ajax pages**
* **Update the userList (or organizationList) file**
* **Update the userDetails (or organizationDetails) file**
* **Update the Admin interface JavaScript file**

**3. Removing a Field from a Form**

**Links for More Information**

To learn more about the Zend framework, including examples and tutorials, go to: <http://framework.zend.com/manual/1.12/en/learning.html>

To learn more about Datatables, go to: <https://datatables.net>

**Adding a Field to a Form**

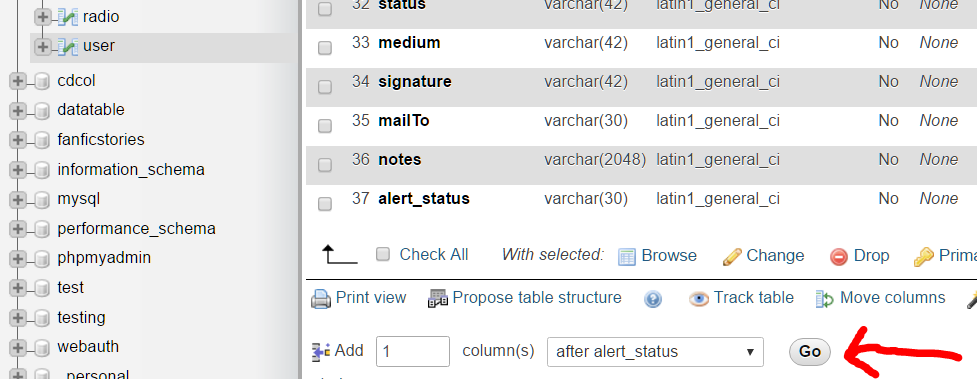
**Overview:** To add a field to either the individual or organization forms, a series of steps must be completed. These include:

* Update the database
* Update the corresponding form and view files
* Update the corresponding confirmation page file
* Update the RadioApplication controller file
* Update the Admin interface files concerning the ajax pages, the userList and userDetails files (or organizationList and organizationDetails), the import/export feature, and the admin.js file

As an example we will be adding a field called firstName to the database in the users table which corresponds to the first name of an individual listener.

**1. Update the Database**

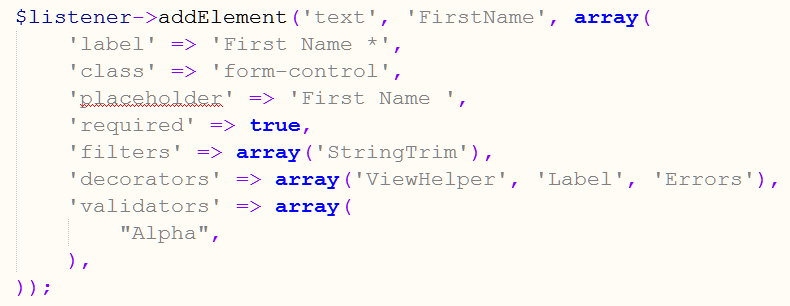
Go to phpmyadmin and navigate to the audioradingservice database. Go to either the user or organization table and add a new column. Click go and fill out the information regarding the new field.



**2. Update the Form and View Files**

Navigate to the forms/RadioApplicationForm.php file. There are four subforms in the file: $listener, $contact, $otherInfo, $statement. Since we are adding a field for the first name of a listener, we will use the $listener subform.

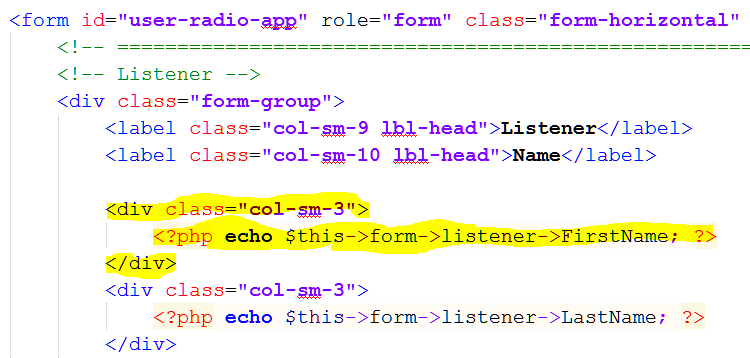
Find the addListenerFields($listener) function in the file and add the following code to the method:



This is the Zend standard for adding fields to a form. The information for each field can be modified at will. The ‘class’ => ‘form-control’ adds bootstrap styling to the new field. Take note of the value given after ‘text’. This is the class name for the field. For more information visit the Zend Framework site.

We still need to update the view file. Navigate to the application/views/scripts/radio-application/radio-form.phtml file.

Under the form in the ‘Listener’ section, add the highlighted code. $this->form->listener->FirstName goes into the corresponding form, looks in the listener section and adds the field that has the class name ‘FirstName’ to the view.



**3. Update the Confirmation Page**

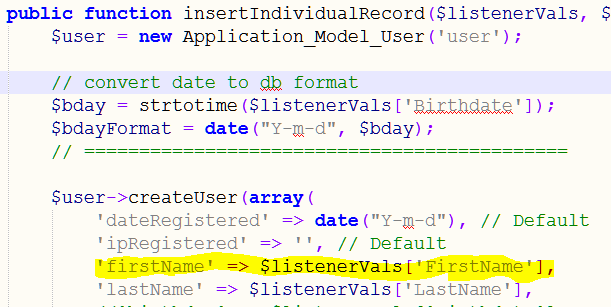
Navigate to the application/views/scripts/radio-application/individual-confirm.phtml file.

The first name field is stored in the $listenerData variable. Echo $listenerData[‘FirstName’] to display the value of the field entered by the user who filled out the application form so that it can be reviewed.

**4. Update the RadioApplication Controller File**

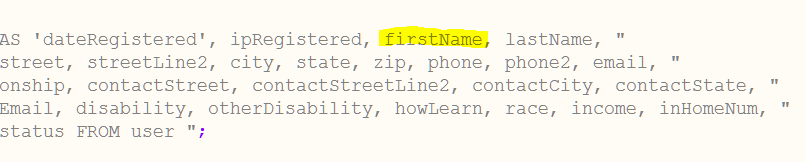
Navigate to the application/controllers/RadioApplicationController.php file.

Under the insertIndividualRecord function, add the highlighted code:

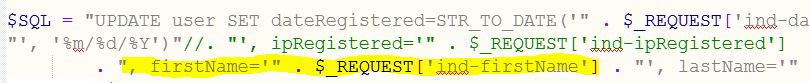


**5. Update the Ajax Pages**

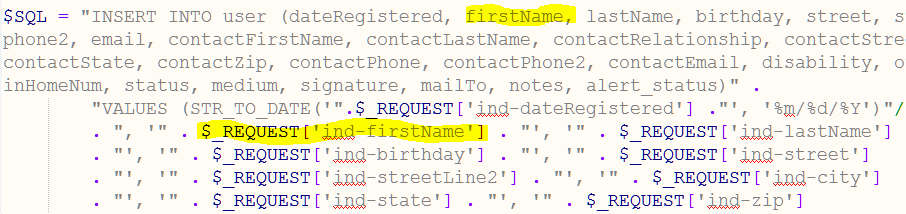
To update the admin/ajax/searchIndividualUsers.php file, add the highlighted code to the SQL statement:



To update the admin/ajax/updateIndividual.php file, add the highlighted code to the SQL statement:



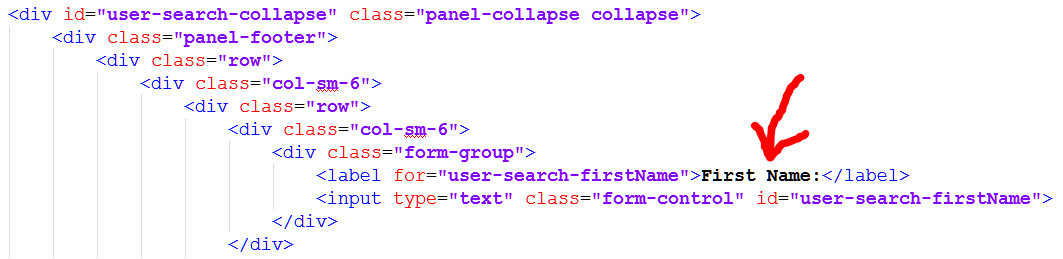
To update the admin/ajax/newIndividual.php file, add the highlighted code to the SQL statement:



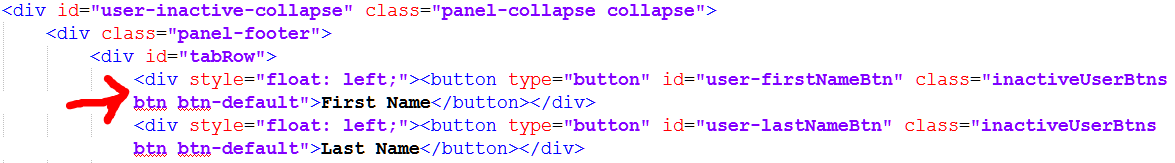
**6. Update the userList file**

Navigate to the admin/tabs/userList.php file.

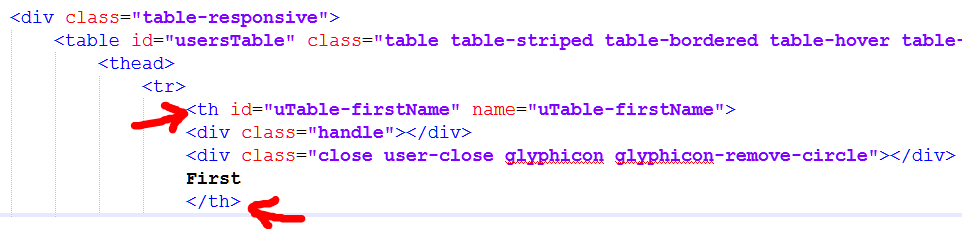
Add in the form group for the listener’s first name under the div for the search collapsible. Make sure to prefix the id of the input tag as ‘user-search-‘ followed by the name of the field which in this case is ‘firstName’.



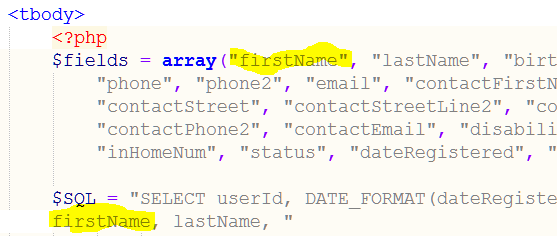
Add an inactive button for the new field to toggle its display in the table. Prefix the id with ‘user-‘ followed by the field name and ending with ‘Btn’ to get ‘user-firstNameBtn’. Make sure to add the inactiveUserBtns class.



In the table add in everything between the two red arrows in the figure below. Prefix the id with ‘uTable-‘ followed by the name, firstName.



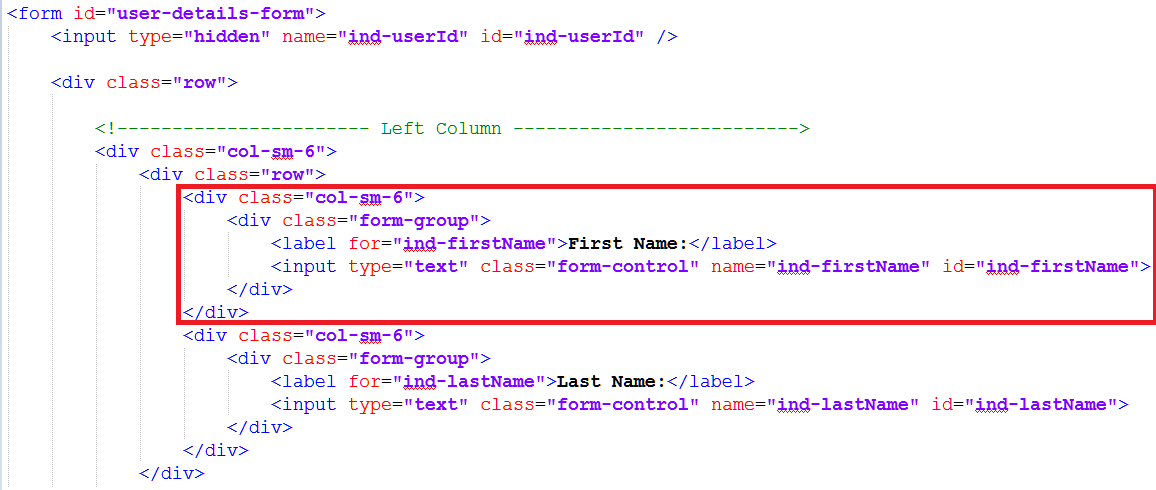
Under the tbody tag add the highlighted code:



**7. Update the userDetails File**

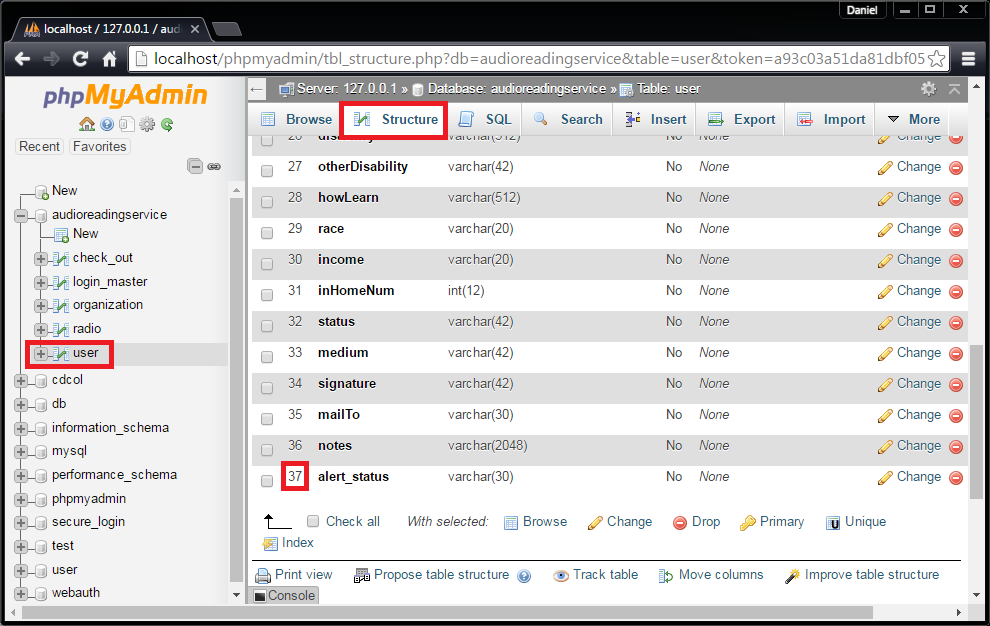
Navigate to admin/tabs/userDetails.php.

Add the code within the red rectangle.



**8. Steps to update Import in the event of a database change**

1. Go to the database and check the structure of the table that has been modified.
2. Scroll to the bottom and check the total number of columns.



**Figure 1**

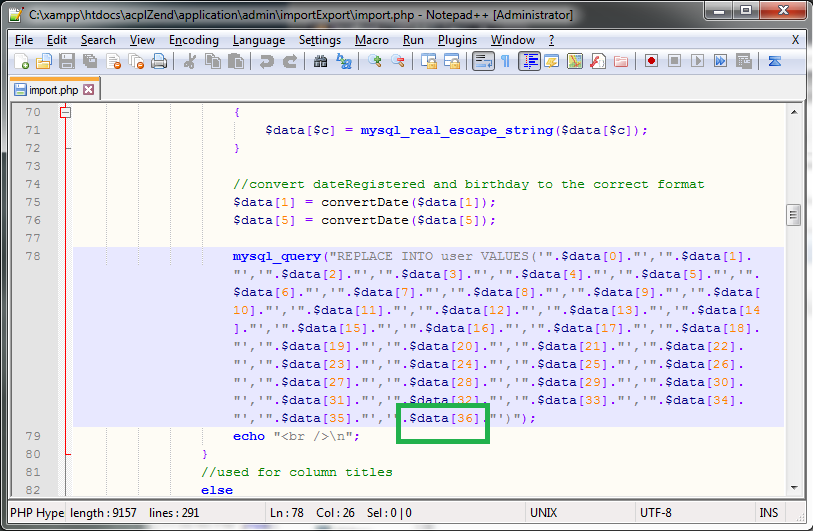
Select the user table, navigate to the Structure tab. Scroll to the bottom, and check the total number of columns. In this case, the total number of columns is 37.

1. Now that we have the total number of columns, we are ready to modify the import.php code.
2. Navigate to the ../admin/importExport/import.php file.
3. Open import.php in your preferred text editor.
4. To update users, navigate to the line “mysql\_query(“REPLACE INTO user VALUES…”)” which is roughly around line 78.

To update radios, navigate to the line “mysql\_query("REPLACE INTO radio VALUES…”)” which is roughly around line 166.

To update organizations, navigate to the line “mysql\_query("REPLACE INTO organization VALUES…”)” which is roughly around line 253.

1. Once you have located the query you wish to update, check the last element of the array.
2. Compare this number to the number of columns we obtained previously.
3. You must subtract 1 from the number we obtained previously to determine the correct value for the array.



**Figure 2**

This is the query for the user table. As we can see, we have an array index of 36, which is one less than the current size of the table column number.

1. If the array index is too low, extend the current array to be the desired length.

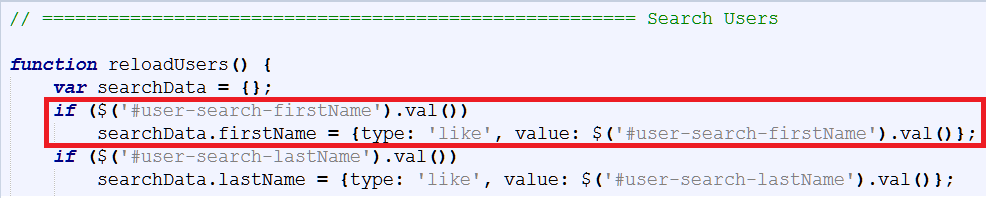
If the array index is too high, delete the excess array elements to the desired length.

1. Once the array matches the structure of the table, import will work again.

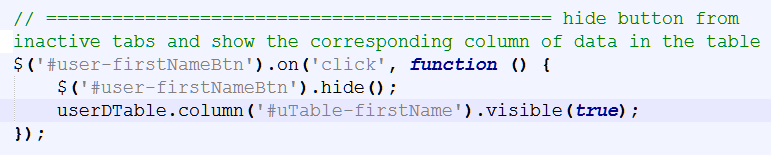
**9. Update the Admin Interface JavaScript File**

Navigate to the ../public/js/admin.js file.

Go to the reloadUsers() function and add the code in the rectangle below.



Add the following code after the comment. This will hide the button from the inactive tabs and display the corresponding column of data in the table.



After this step, everything regarding the new field should be up to date and work as expected.

**Removing a Field from a Form**

To remove a field from either the individual or organization radio application forms, the process will be similar to adding a field, just opposite. See the guide for ‘Adding a Field to the Form’. Instead of adding the given lines of code, remove them. Likewise, delete the field from the database table in phpmyadmin.