

## Working with MVC

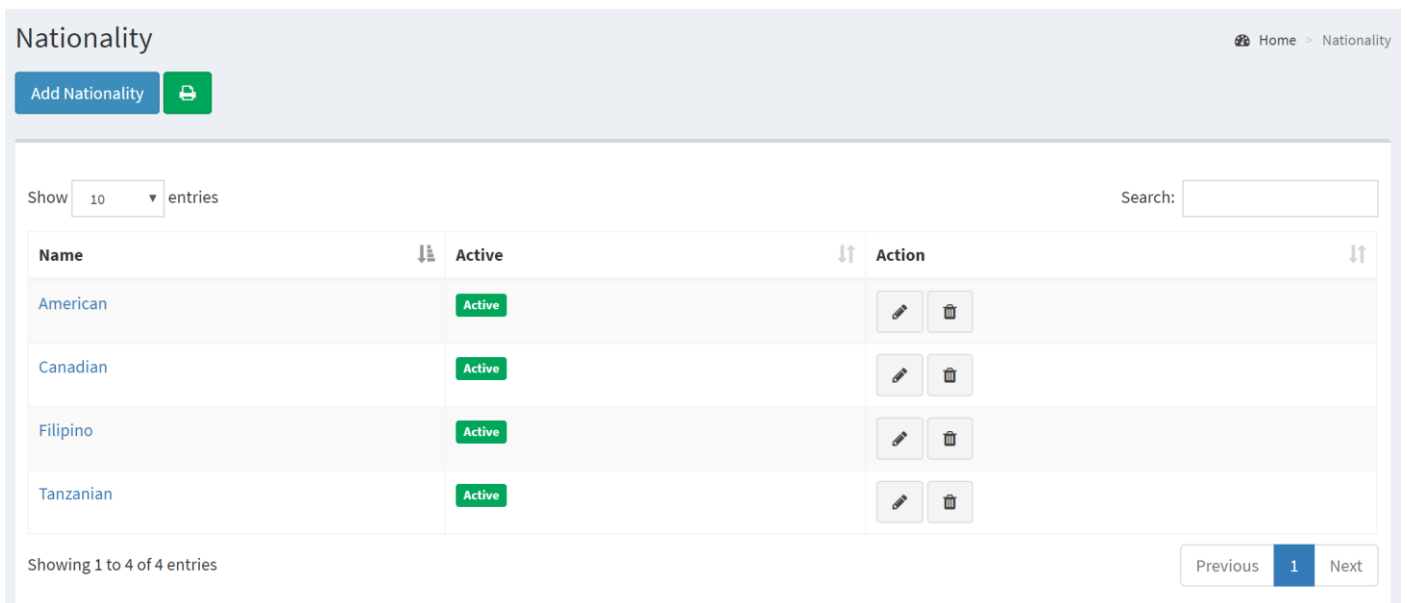
This example will introduce you to the CodeIgniter framework and the basic principles of MVC architecture. It will show you how a basic CodeIgniter module is constructed in step-by-step fashion. We will take a very simple example: The management of the table Nationality. (View – Add – Update and Delete)

Here is the structure of the table:

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<b>id</b>	int(11)			No	None		AUTO_INCREMENT
<b>name</b>	varchar(100)	utf8_general_ci		No	None		
<b>active</b>	tinyint(1)			No	None	1=active 2=inactive	

### List of Nationality

The view is the form that will be seen by the user. In the case of the management of nationality, all the process is inside /application/views/nationality/index.php. The list, creation, update and delete will use the facilities of datatables.net, javascript and ajax.

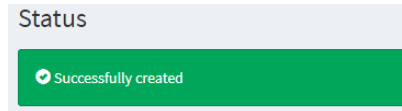


All the process will work using the MVC codeigniter principle. The controller /application/controllers/Nationality.php in charge of the control and the model /application/models/model\_nationality.php where all the queries to the database will be done. Finally, the view under /application/views/nationality/index.php.

To understand the whole process, open the controller Nationality.php, the model\_nationality and the view nationality/index.php in your editor.



1. First we evaluate if the last submit of the form had generate an error message or a confirmation of the success of the operation. The message will be presented to the user.



2. Security: We verify if the user can create a new nationality. If yes, the button will be presented. The rest of the access (edit and delete) will be managed inside the controller where the line for each nationality will be composed with the data from the database.

```

<!------- View ----->
<section class="content">
  <div class="row">
    <div class="col-md-12 col-xs-12">
      <div id="messages"></div>
      <?php if($this->session->flashdata('success')): ?>
        <div class="alert alert-success alert-dismissible" role="alert">
          <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>
          <?php echo $this->session->flashdata('success'); ?>
        </div>
      <?php elseif($this->session->flashdata('error')): ?>
        <div class="alert alert-error alert-dismissible" role="alert">
          <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>
          <?php echo $this->session->flashdata('error'); ?>
        </div>
      <?php endif; ?>
      <?php if(in_array('createNationality', $user_permission)): ?>
        <button class="btn btn-primary" data-toggle="modal" onclick="createFunc()" data-target="#addModal"><?php echo $this->lang->line('Add
        Nationality'); ?></button>
      <?php endif; ?>
      <?php if(in_array('viewNationality', $user_permission)): ?>
        <?php echo '<a href="'.base_url('report20/REP20/nationality').'" target="_blank" class="btn btn-success"><i class="fa
        fa-print"></i></a>'; ?>
      <br /> <br />
      <?php endif; ?>
      <div class="box">
        <div class="box-header"></div>
        <div class="box-body">
          <table id="manageTable" class="table table-bordered table-striped">
            <thead>
              <tr>
                <th><?php echo $this->lang->line('Name'); ?></th>
                <th><?php echo $this->lang->line('Active'); ?></th>
                <?php if(in_array('updateNationality', $user_permission) // in_array('deleteNationality', $user_permission)): ?>
                  <th><?php echo $this->lang->line('Action'); ?></th>
                <?php endif; ?>
              </tr>
            </thead>
          </table>
        </div>
      </div>
    </div>
  </div>
</div>

```

1

2

3. The list is managed by javascript and Ajax. You will find the javascript part at the end of the form.

```

<script type="text/javascript">
var manageTable;

$(document).ready(function() {
  $("#nationalityNav").addClass('active');

  // initialize the datatable
  manageTable = $('#manageTable').DataTable({
    'ajax': 'fetchNationalityData',
    'language': { 'url': "<?php echo $this->session->link_language; ?>" },
    'order': [[0, 'asc']]
  });
});

```

3

manageTable is created here invoking the controller in /application/controllers/nationality.php where the function **fetchNationalityData** will organise the list and call the model Model\_nationality to reach the database.

This is where you can also modify the default order of the table.

- 4. Controller Nationality.php: This is where the function **getNationalityData** is called. You can see how the security is build giving permission or not to update or delete the entry in the table. You will find the function `getNationalityData` in `/application/models/Model_nationality.php`. The datas are treated by the controller and the result is back to the view `index.php` in json format.

```

//--> Fetches the nationality value from the nationality table
// This function is called from the datatable ajax function

public function fetchNationalityData()
{
    $result = array('data' => array());

    $data = $this->model_nationality->getNationalityData();

    foreach ($data as $key => $value) {

        $buttons = '';

        if(in_array('updateNationality', $this->permission)) {
            $buttons .= '<button type="button" class="btn btn-default" onclick="editFunc('.$value['id'].')" data-toggle="modal"
            data-target="#editModal"><i class="fa fa-pencil"></i></button>';
            $name= ' <a data-target="#editModal" onclick="editFunc('.$value['id'].')" data-toggle="modal" href="#editModal">'.$value['name'
            ].'</a>';
        }

        if(in_array('deleteNationality', $this->permission)) {
            $buttons .= ' <button type="button" class="btn btn-default" onclick="removeFunc('.$value['id'].')" data-toggle="modal"
            data-target="#removeModal"><i class="fa fa-trash"></i></button>';
        }

        $active = ($value['active'] == 1) ? '<span class="label label-success">'.$this->lang->line('Active').'</span>' : '<span
        class="label label-warning">'.$this->lang->line('Inactive').'</span>';

        $result['data'][$key] = array(
            $name,
            $active,
            $buttons
        );
    } // /foreach

    echo json_encode($result);
}

```

4

- 5. Model\_nationality.php: Here is the query to the database. If the ID is sent to the function, only the entry will be returned. But in the case of the list, all the entries in the table Nationality will be returned.

```

public function getNationalityData($id = null)
{
    if($id) {
        $sql = "SELECT * FROM nationality WHERE id = ?";
        $query = $this->db->query($sql, array($id));
        return $query->row_array();
    }

    $sql = "SELECT * FROM nationality";
    $query = $this->db->query($sql);
    return $query->result_array();
}

```

5



7. Function create of controller Nationality.php: This is where the validation will be done, in this case the name is required. If it's not filled, the validation will be false and an error message will be returned to the javascript form. If it's valid, the controller will call the function **create** in the model\_nationality

```
//--> It checks the nationality form validation
// and if the validation is true (valid) then it inserts the data into the database
// and returns the json format operation messages

public function create()
{
    if(!in_array('createNationality', $this->permission)) {redirect('dashboard', 'refresh');}

    $response = array();

    $this->form_validation->set_rules('nationality_name', $this->lang->line('Name'), 'trim|required');
    $this->form_validation->set_error_delimiters('<p class="text-danger">', '</p>');

    if ($this->form_validation->run() == TRUE) {
        $data = array(
            'name' => $this->input->post('nationality_name'),
            'active' => $this->input->post('active'),
        );

        $create = $this->model_nationality->create($data);
        if($create == true) {
            $response['success'] = true;
            $response['messages'] = $this->lang->line('Successfully created');
        }
        else {
            $response['success'] = false;
            $response['messages'] = $this->lang->line('Error in the database while creating the information');
        }
    }
    else {
        $response['success'] = false;
        foreach ($_POST as $key => $value) {
            $response['messages'][$key] = form_error($key);
        }
    }

    echo json_encode($response);
}
```

7

```
public function create($data)
{
    if($data) {
        $insert = $this->db->insert('nationality', $data);
        return ($insert == true) ? true : false;
    }
}
```



## Delete nationality

- 11. Same process, delete is in the view index.php, calling the javascript removeModal
- 12. On submit, in this case authorization for the delete, the function **remove** from the controller Nationality.php will be called.

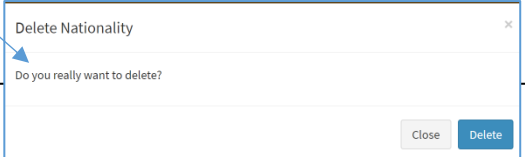
```

<!------- Delete ----->
<?php if(in_array('deleteNationality', $user_permission)): ?>
<!-- word -->
<div class="modal fade" tabindex="-1" role="dialog" id="removeModal">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
        <h4 class="modal-title"><?php echo $this->lang->line('Delete Nationality'); ?></h4>
      </div>
      <form role="form" action="<?php echo base_url('nationality/remove') ?>" method="post" id="removeForm">
        <div class="modal-body">
          <p><?php echo $this->lang->line('Do you really want to delete?'); ?></p>
        </div>
        <div class="modal-footer">
          <button type="button" class="btn btn-default" data-dismiss="modal"><?php echo $this->lang->line('Close'); ?></button>
          <button type="submit" class="btn btn-primary"><?php echo $this->lang->line('Delete'); ?></button>
        </div>
      </form>
    </div><!-- /.modal-content -->
  </div><!-- /.modal-dialog -->
</div><!-- /.modal -->
<?php endif; ?>

```

11

12



```

// remove functions
function removeFunc(id)
{
  if(id) {
    $("#removeForm").on('submit', function() {
      var form = $(this);
      // remove the text-danger
      $(".text-danger").remove();
      $.ajax({
        url: form.attr('action'),
        type: form.attr('method'),
        data: { nationality_id:id },
        dataType: 'json',
        success: function(response) {

```

/application/views/nationality/ Index.php -

- 13. The **remove** function of the controller nationality.php will verify the integrity of the database. The nationality can be already used in the table license and if an entry exists, the delete will not be possible. The function is **checkIntegrity** in the model\_nationality
- 14. If the information is not in license table, then the delete can be done, calling the function **remove** in the model\_nationality.

```

//--> It removes the nationality information from the database
// and returns the json format operation messages
public function remove()
{
  if(!in_array('deleteNationality', $this->permission)) {
    redirect('dashboard', 'refresh');
  }
  $nationality_id = $this->input->post('nationality_id');
  $response = array();
  if($nationality_id) {
    //--- Validate if the information is used in beekeeper table
    $total_beekeeper = $this->model_nationality->checkIntegrity($nationality_id);
    //--- If no beekeeper have this information, we can delete
    if ($total_beekeeper == 0) {
      $delete = $this->model_nationality->remove($nationality_id);
      if($delete == true) {
        $response['success'] = true;
        $response['messages'] = $this->lang->line('Successfully deleted');
      } else {
        $response['success'] = false;
        $response['messages'] = $this->lang->line('Error in the database while deleting the information');
      }
    } else {
      //--- There is at least one beekeeper having this information
      $response['success'] = false;
      $response['messages'] = $this->lang->line('At least one beekeeper uses this information.
    }
  } else {
    $response['success'] = false;
    $response['messages'] = $this->lang->line('Refresh the page again');
  }
  echo json_encode($response);
}

```

13

```

public function remove($id)
{
  if($id) {
    $this->db->where('id', $id);
    $delete = $this->db->delete('nationality');
    return ($delete == true) ? true : false;
  }
  //--- Validate if the nationality is used in table Beekeeper
  public function checkIntegrity($id)
  {
    // select with the wildcard %. It is possible to have more
    // than one nationality in beekeeper table. In this case, the information
    // will appear between bracket [1]. The search will be
    // SELECT * FROM beekeeper WHERE nationality_id LIKE '%[1]%'
    $this->db->select('*');
    $this->db->from('beekeeper');
    $this->db->like('nationality_id', $id, 'both');
    $query = $this->db->get();
    return $query->num_rows();
  }
}

```

14