2020

How to …

CODEIGNITER

Carmen Gagnon

January 2020

Content

TECHNOLOGY **4**

Code Igniter 4

Model – View – Controller 4

DATABASE **5**

Database Diagram 5

Relations between tables 5

JOIN (or INNER JOIN) *6*

LEFT JOIN *6*

RIGHT JOIN *6*

Comparison operator 7

Database and CodeIgniter 7

Database configuration *7*

Connecting the database *7*

Query *8*

Inserting data *8*

Updating data *8*

Deleting data *9*

STRUCTURE OF THE SYSTEM **10**

Dashboard and menu 10

News 10

Beekeeper 11

Apiaries *12*

Inquiries *12*

Documents *13*

Apiaries 13

Colonies *14*

Documents *14*

Map *14*

Colonies 15

Production *15*

Reports 16

Settings 17

Documentation 17

WORKING WITH MVC **18**

List of Nationality 18

Add nationality 21

Update nationality 23

Delete nationality 24

HOW TO ADD A NEW FIELD **25**

Create in the table 25

Change the controller – model and views 25

Modification of controller Association.php. *25*

Modification of the model *26*

Modification of the view in Association *26*

Verify the language translation 27

Modify the report 28

Modify the user guide 28

Testing 28

SECURITY IN THE SYSTEM **28**

PROFILE 28
USER 29

SESSION VARIABLES 30

FLASHDATA 30
USERDATA 30

TRANSLATION 31

LANGUAGE AND USER 31
LANGUAGE CLASS 31
SEQUENCE FOR TRANSLATION 31

JQUERY MODAL AND JAVASCRIPT 33

ORDER THE TABLE WITH JAVASCRIPT 33
PASSING PARAMETER 33

UPLOAD OF DOCUMENTS 34

CREATION OF THE DIRECTORY OF BEEKEEPER 34
TABLES DOCUMENT AND DOCUMENT TYPE 34
UPLOAD OF DOCUMENTS 35
VIEW OF DOCUMENTS 36
DELETE A DOCUMENT 36
DOWNLOAD AND ZIP FILE 37
1. Store files 37
2. Configuration 37
3. Controller 37
Add files – 37
Add directory files and sub-directory – 37
Completed Code 38
4. View 39
Completed Code 39

VALIDATION 40

LIBRARY FORM VALIDATION 40
CREATING OUR OWN VALIDATION WITH CALLBACK METHOD 40
CHECK INTEGRITY 41
FORM VALIDATION PARAMETER 42

REPORTS 43

TCPDF AND CODEIGNITER 43
ADD A NEW CRITERIA IN A REPORT 43
Add a field in a report 44
CREATION OF NEW REPORT IN THE SYSTEM 45
Step 1 - Report template needed 45
Step 2 – Create the new report in the report list 45
Step 3 – Integrate this new report in the controller Report.php and view /report/index.php 45
Step 4 – Create the SQL query for report 03 in /models/model_report.php 46
Step 5 – Create the report in /application/controllers/Report03.php 47
PRINT A TOTAL FOR THE REPORT 48
MAIN COMMAND TCPDF 49

TOOLS FOR FINDING ERRORS 51

PRINT_R IN PHP 51
ALERT() IN JAVASCRIPT 51
DEVELOPER TOOLS IN BROWSER 51

PHP BASICS 52

HTML BASICS **53**

FONT-AWESOME (FA) **56**

BOOTSTRAP BASICS **58**

GRID SYSTEM 58
ALERT 58
BUTTON 59
GLYPHICON 59
COLOR 61

GOOGLE MAP **62**

EMBED A MAP 62
GOOGLE MAP API WITH KEY 63
WORK IN CODEIGNITER 63

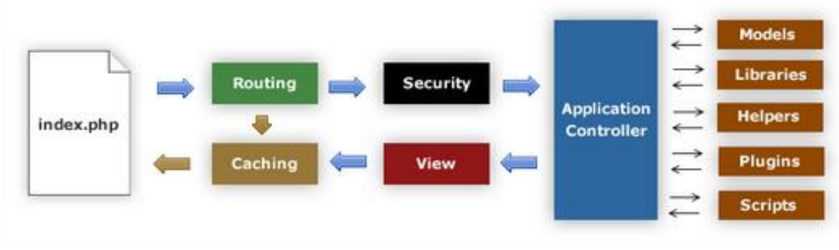
SOME TRICKS **65**

REMOVE TRAILING SPACES IN THE CODE 65
MIGRATION PROCESS 65
TAB CONTROL 66
MUTIPLE CHOICE FOR A DROP-DOWN LIST (JSON) 68
FUNCTION FOR CLEARING THE FORM WITH AJAX 69
BOOTSTRAP TOOLTIP 70
INSTALLATION OF A LOG 70

Technology

Code Igniter

CodeIgniter is an Application Development Framework - a toolkit - for people who build web sites using PHP. Its goal is to enable you to develop projects much faster than you could if you were writing code from scratch, by providing a rich set of libraries for commonly needed tasks, as well as a simple interface and logical structure to access these libraries. Your system is using the framework Code Igniter.



Model – View – Controller

CodeIgniter is based on the Model-View-Controller development pattern. MVC is a software approach that separates application logic from presentation. In practice, it permits your web pages to contain minimal scripting since the presentation is separate from the PHP scripting. Your system is based on these principles.

- The **Model** represents your data structures. Typically your model classes will contain functions that help you retrieve, insert, and update information in your database.
- The **View** is the information that is being presented to a user. A View will normally be a web page, but in CodeIgniter, a view can also be a page fragment like a header or footer.
- The **Controller** serves as an intermediary between the Model, the View, and any other resources needed to process the HTTP request and generate a web page.
- To get the **version** of the codeigniter used in your system look for define in system/core/CodeIgniter.php (define('CI_VERSION', '3.x.xxx');

Model	View	Controller	
Model_apiary.php	apiary	Apiary.php	Profile.php
Model_association.php	association	Association.php	Province.php
Model_auth.php	beekeeper	Auth.php	Region.php
Model_beekeeper.php	category	Backup.php	Report.php
Model_category.php	colony	Beekeeper.php	Report01.php
Model_colony.php	district	Category.php	Report02.php
Model_district.php	document_type	Colony.php	Report03.php
Model_document_type.php	documentation	Dashboard.php	Report04.php
Model_education.php	education	District.php	Report05.php
Model_fund_source.php	errors	Document_type.php	Report06.php
Model_gender.php	fund_source	Documentation.php	Report20.php
Model_inquiry.php	gender	Education.php	Report21.php
Model_inquiry_type.php	inquiry_type	Fund_Source.php	Report22.php
Model_municipality.php	municipality	Gender.php	Report23.php
Model_nationality.php	nationality	index.html	Report24.php
Model_phase.php	phase	Inquiry.php	Setting.php
Model_post.php	post	Inquiry_type.php	Source.php
Model_post_category.php	post_category	Migrate.php	Species.php
Model_product.php	product	Municipality.php	Support_type.php
Model_production.php	profile	Nationality.php	Topography.php
Model_profile.php	province	Phase.php	User.php
Model_province.php	region	Post.php	Website.php
Model_region.php	report	Post_category.php	
Model_report.php	setting	Product.php	
Model_source.php	source	Production.php	
Model_species.php	species		
Model_support_type.php	support_type		
Model_topography.php	templates		
Model_user.php	topography		
	user		
	website		

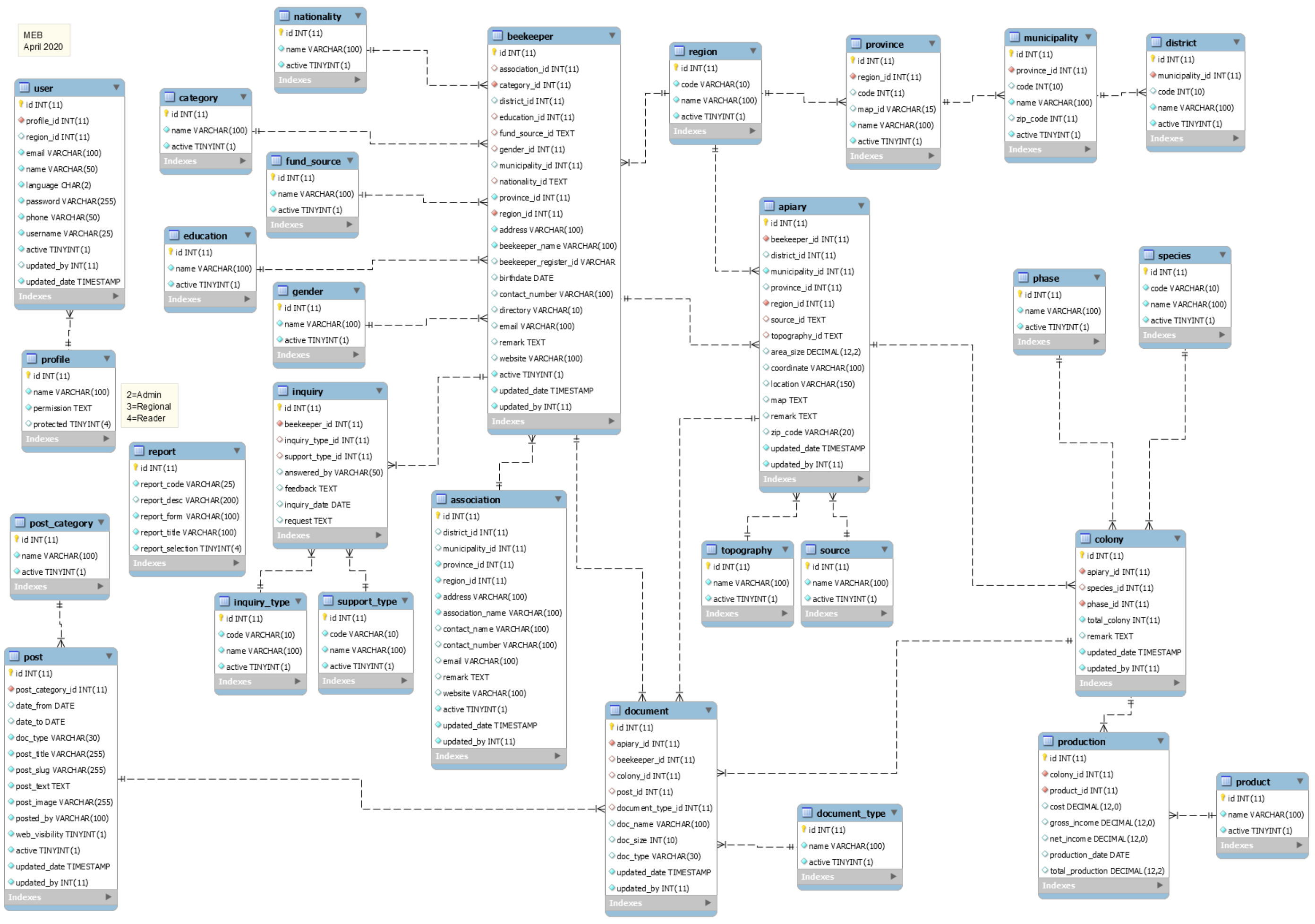
Database

MySQL is the world's most popular open source database. With its proven performance, reliability and ease-of-use, MySQL has become the leading database choice. Oracle drives MySQL innovation, delivering new capabilities to power next generation web, cloud, mobile and embedded applications. MySQL is a relational database, in that it allows tables to be joined together and also supports the concept of foreign keys.

The database has several relationships between the tables, which is illustrated by the following diagram. It's possible to relate the tables directly in MySQL but the technology choice for the database is to JOIN the tables when it's needed in the model. The performance is better and the management of the database will be easier.

Database Diagram

MySQL Workbench is used to create the diagram of the database. The relations are made manually after a synchronization with the database.

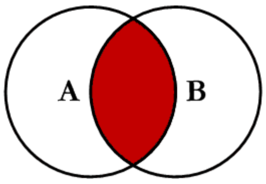

MEB April 2020
nationality: id INT(11), name VARCHAR(100), active TINYINT(1), Indexes
beekeeper: id INT(11), association_id INT(11), category_id INT(11), district_id INT(11), education_id INT(11), fund_source_id TEXT, gender_id INT(11), municipality_id INT(11), nationality_id TEXT, province_id INT(11), region_id INT(11), address VARCHAR(100), beekeeper_name VARCHAR(100), beekeeper_register_id VARCHAR, birthdate DATE, contact_number VARCHAR(100), directory VARCHAR(10), email VARCHAR(100), remark TEXT, website VARCHAR(100), active TINYINT(1), updated_date TIMESTAMP, updated_by INT(11), Indexes
region: id INT(11), code VARCHAR(10), name VARCHAR(100), active TINYINT(1), Indexes
province: id INT(11), region_id INT(11), code INT(11), map_id VARCHAR(15), name VARCHAR(100), active TINYINT(1), Indexes
municipality: id INT(11), province_id INT(11), code INT(10), name VARCHAR(100), zip_code INT(11), active TINYINT(1), Indexes
district: id INT(11), municipality_id INT(11), code INT(10), name VARCHAR(100), active TINYINT(1), Indexes
user: id INT(11), profile_id INT(11), region_id INT(11), email VARCHAR(100), name VARCHAR(50), language CHAR(2), password VARCHAR(255), phone VARCHAR(50), username VARCHAR(25), active TINYINT(1), updated_by INT(11), updated_date TIMESTAMP, Indexes
category: id INT(11), name VARCHAR(100), active TINYINT(1), Indexes
fund_source: id INT(11), name VARCHAR(100), active TINYINT(1), Indexes
education: id INT(11), name VARCHAR(100), active TINYINT(1), Indexes
gender: id INT(11), name VARCHAR(100), active TINYINT(1), Indexes
profile: id INT(11), name VARCHAR(100), permission TEXT, protected TINYINT(4), Indexes
2=Admin 3=Regional 4=Reader
apiary: id INT(11), beekeeper_id INT(11), district_id INT(11), municipality_id INT(11), province_id INT(11), region_id INT(11), source_id TEXT, topography_id TEXT, area_size DECIMAL(12,2), coordinate VARCHAR(100), location VARCHAR(150), map TEXT, remark TEXT, zip_code VARCHAR(20), updated_date TIMESTAMP, updated_by INT(11), Indexes
phase: id INT(11), name VARCHAR(100), active TINYINT(1), Indexes
species: id INT(11), code VARCHAR(10), name VARCHAR(100), active TINYINT(1), Indexes
inquiry: id INT(11), beekeeper_id INT(11), inquiry_type_id INT(11), support_type_id INT(11), answered_by VARCHAR(50), feedback TEXT, inquiry_date DATE, request TEXT, Indexes
report: id INT(11), report_code VARCHAR(25), report_desc VARCHAR(200), report_form VARCHAR(100), report_title VARCHAR(100), report_selection TINYINT(4), Indexes
association: id INT(11), district_id INT(11), municipality_id INT(11), province_id INT(11), region_id INT(11), address VARCHAR(100), association_name VARCHAR(100), contact_name VARCHAR(100), contact_number VARCHAR(100), email VARCHAR(100), remark TEXT, website VARCHAR(100), active TINYINT(1), updated_date TIMESTAMP, updated_by INT(11), Indexes
post_category: id INT(11), name VARCHAR(100), active TINYINT(1), Indexes
topography: id INT(11), name VARCHAR(100), active TINYINT(1), Indexes
source: id INT(11), name VARCHAR(100), active TINYINT(1), Indexes
colony: id INT(11), apiary_id INT(11), species_id INT(11), phase_id INT(11), total_colony INT(11), remark TEXT, updated_date TIMESTAMP, updated_by INT(11), Indexes
inquiry_type: id INT(11), code VARCHAR(10), name VARCHAR(100), active TINYINT(1), Indexes
support_type: id INT(11), code VARCHAR(10), name VARCHAR(100), active TINYINT(1), Indexes
post: id INT(11), post_category_id INT(11), date_from DATE, date_to DATE, doc_type VARCHAR(30), post_title VARCHAR(255), post_slug VARCHAR(255), post_text TEXT, post_image VARCHAR(255), posted_by VARCHAR(100), web_visibility TINYINT(1), active TINYINT(1), updated_date TIMESTAMP, updated_by INT(11), Indexes
document: id INT(11), apiary_id INT(11), beekeeper_id INT(11), colony_id INT(11), post_id INT(11), document_type_id INT(11), doc_name VARCHAR(100), doc_size INT(10), doc_type VARCHAR(30), updated_date TIMESTAMP, updated_by INT(11), Indexes
document_type: id INT(11), name VARCHAR(100), active TINYINT(1), Indexes
production: id INT(11), colony_id INT(11), product_id INT(11), cost DECIMAL(12,0), gross_income DECIMAL(12,0), net_income DECIMAL(12,0), production_date DATE, total_production DECIMAL(12,2), Indexes
product: id INT(11), name VARCHAR(100), active TINYINT(1), Indexes

Relations between tables

The type of JOIN will determine which data will be selected. There is many different ways you can return data from two relational tables. I am excluding cross Joins and self referencing Joins. For the database most of the time, only 3 JOIN methods have been used (JOIN – LEFT JOIN – RIGHT JOIN).

1. INNER JOIN OR JOIN
2. LEFT JOIN
3. RIGHT JOIN
4. OUTER JOIN
5. LEFT JOIN EXCLUDING INNER JOIN
6. RIGHT JOIN EXCLUDING INNER JOIN
7. OUTER JOIN EXCLUDING INNER JOIN

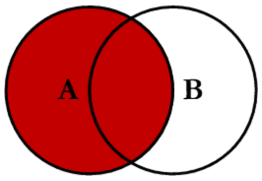
JOIN (or INNER JOIN)



This is the simplest, most understood Join and is the most common. This query will return all of the records in the left table (table A) that have a matching record in the right table (table B). This Join is written as follows:

SELECT <select_list>
FROM Table_A A
JOIN Table_B B
ON A.Key = B.Key

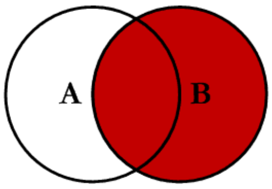
LEFT JOIN



This query will return all of the records in the left table (table A) regardless if any of those records have a match in the right table (table B). It will also return any matching records from the right table. This Join is written as follows:

SELECT <select_list>
FROM Table_A A
LEFT JOIN Table_B B
ON A.Key = B.Key

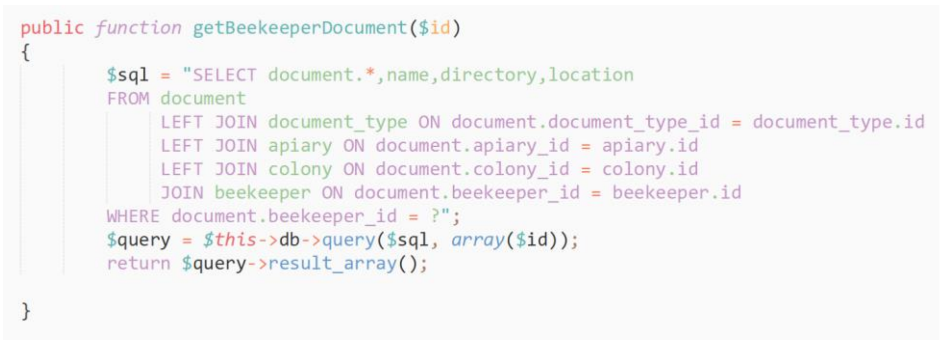
RIGHT JOIN



This query will return all of the records in the right table (table B) regardless if any of those records have a match in the left table (table A). It will also return any matching records from the left table. This Join is written as follows:

SELECT <select_list>
FROM Table_A A
RIGHT JOIN Table_B B
ON A.Key = B.Key

In the model_beekeeper, you find an example of JOIN. In this case there should be a match between beekeeper and and document, unless the line will not be qualified. But for apiary table, there should be a match or not. If there is no apiary in relation, the line will still be qualified.



```
public function getBeekeeperDocument($id)
{
        $sql = "SELECT document.*,name,directory,location
        FROM document
             LEFT JOIN document_type ON document.document_type_id = document_type.id
             LEFT JOIN apiary ON document.apiary_id = apiary.id
             LEFT JOIN colony ON document.colony_id = colony.id
             JOIN beekeeper ON document.beekeeper_id = beekeeper.id
        WHERE document.beekeeper_id = ?";
        $query = $this->db->query($sql, array($id));
        return $query->result_array();

}
```

Comparison operator

Name	Description
BETWEEN ... AND ...	Check whether a value is within a range of values
COALESCE()	Return the first non-NULL argument
=	Equal operator
<=>	NULL-safe equal to operator
>	Greater than operator
>=	Greater than or equal operator
GREATEST()	Return the largest argument
IN()	Check whether a value is within a set of values
INTERVAL()	Return the index of the argument that is less than the first argument
IS	Test a value against a boolean
IS NOT	Test a value against a boolean
IS NOT NULL	NOT NULL value test
IS NULL	NULL value test
ISNULL()	Test whether the argument is NULL
LEAST()	Return the smallest argument
<	Less than operator
<=	Less than or equal operator
LIKE	Simple pattern matching
NOT BETWEEN ... AND ...	Check whether a value is not within a range of values
!=, <>	Not equal operator
NOT IN()	Check whether a value is not within a set of values
NOT LIKE	Negation of simple pattern matching
STRCMP()	Compare two strings

Database and CodeIgniter

CodeIgniter comes with a full-featured and very fast abstracted database class that supports both traditional structures and Query Builder patterns. The database functions offer clear, simple syntax.

Database configuration

The database configuration is in **application/config/database.php**. You need to indicate where is the server of the database, the username, password and name of the database.

```
$db['default'] = array(
	'dsn'	=> '',
	'hostname' => 'localhost',
	'username' => 'root',
	'password' => '',
	'database' => 'meb',
	'dbdriver' => 'mysqli',
	'dbprefix' => '',
	'pconnect' => FALSE,
	'db_debug' => (ENVIRONMENT !== 'production'),
	'cache_on' => FALSE,
	'cachedir' => '',
	'char_set' => 'utf8',
	'dbcollat' => 'utf8_general_ci',
	'swap_pre' => '',
	'encrypt' => FALSE,
	'compress' => FALSE,
	'stricton' => FALSE,
	'failover' => array(),
	'save_queries' => TRUE
```

Connecting the database

In the system, the database is automatically connected when we start the system, in **application/config/autoload.php**

```
$autoload['libraries'] = array('session', 'form_validation', 'database', 'zip', 'calendar', 'table', 'Pdf');
```

Query

$this->db->query()

To submit a query, we use the query function. ($this_db->query). In this example, a parameter is passed to the query to get one row or all the rows. In the case we expect one row, we will use **row_array()** for the return. For many rows, it will be **result_array()**

```
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();
}
```

Inserting data

$this->db->insert()

Generates an insert string based on the data you supply, and runs the query. You can either pass an array or an object to the function.

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

In application/controllers/Nationality.php an array is created to pass $data.

```
$data = array(
    'name' => $this->input->post('nationality_name'),
    'active' => $this->input->post('active'),
);

$create = $this->model_nationality->create($data);
```

Updating data

$this->db->update()

Generates an update string and runs the query based on the data you supply. You can pass an array or an object to the function. Here is an example using an array:

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

In this case we pass the $data, which is an array created in application/controllers/Nationality.php and also the $id of the row of table Nationality we want to update.

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

    $update = $this->model_nationality->update($data, $id);

    if($update == true) {
        $response['success'] = true;
        $response['messages'] = $this->lang->line('Successfully updated');
    }
    else {
        $response['success'] = false;
        $response['messages'] = $this->lang->line('Error in the database while updating the information');

    }
```

Deleting data

$this->db->delete()

Generates a delete SQL string and runs the query.

```
public function remove($id)
{
    if($id) {
        $this->db->where('id', $id);
        $delete = $this->db->delete('nationality');
        return ($delete == true) ? true : false;
    }
}
```

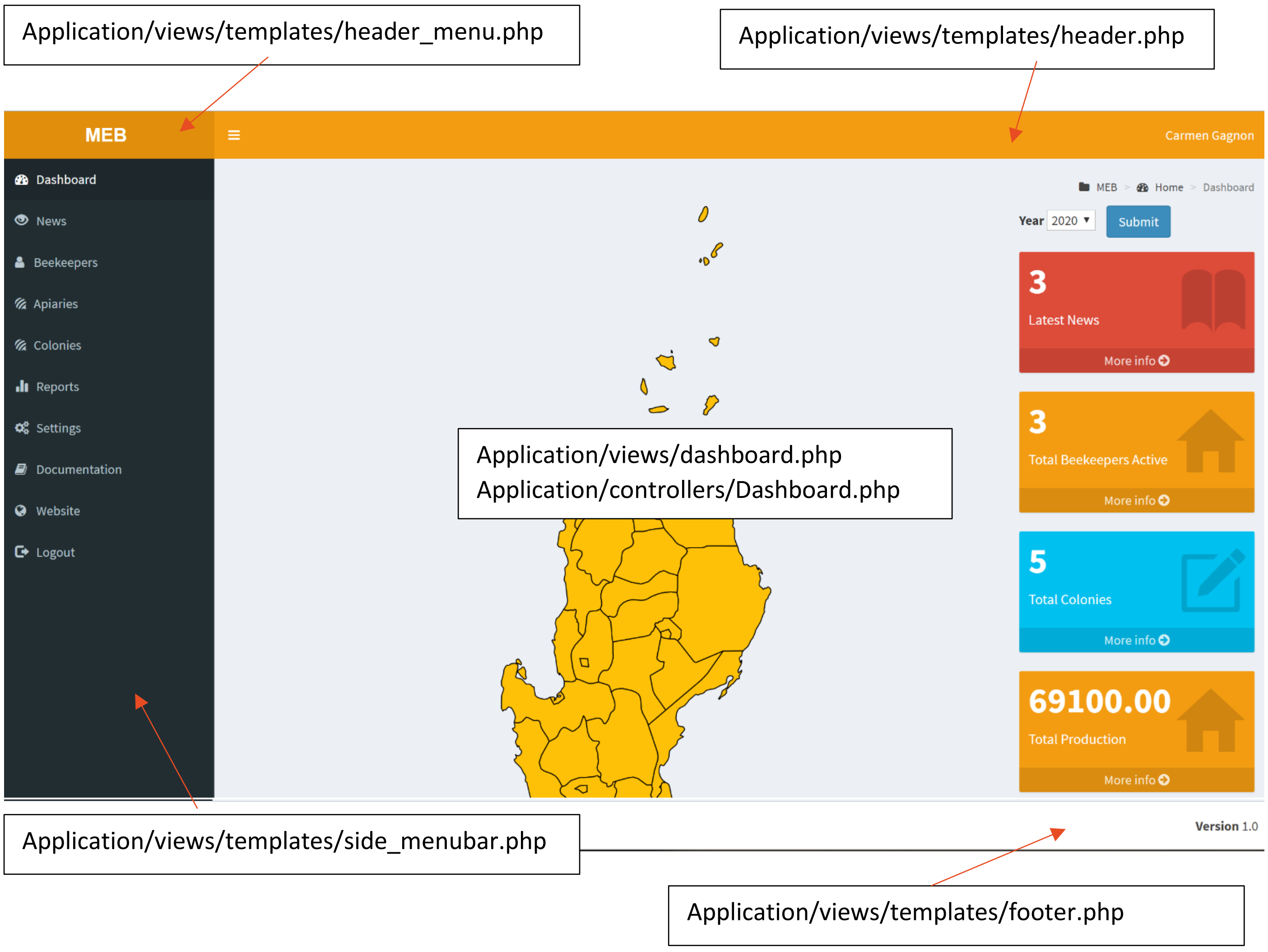
```
//---> Validate if the nationality is used in table Beekeeper
public function checkIntegrity($id)
{
    // select with the wildcard %.  It is possible to have more
    // than one nationality in beekeeper table.   In this case, the information
    // will appear between bracket ["1"].  The search 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();

}
```

In the controller, we check the integrity of the database before deleting a row.

```
//---> 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');}
    }
```

Structure of the system

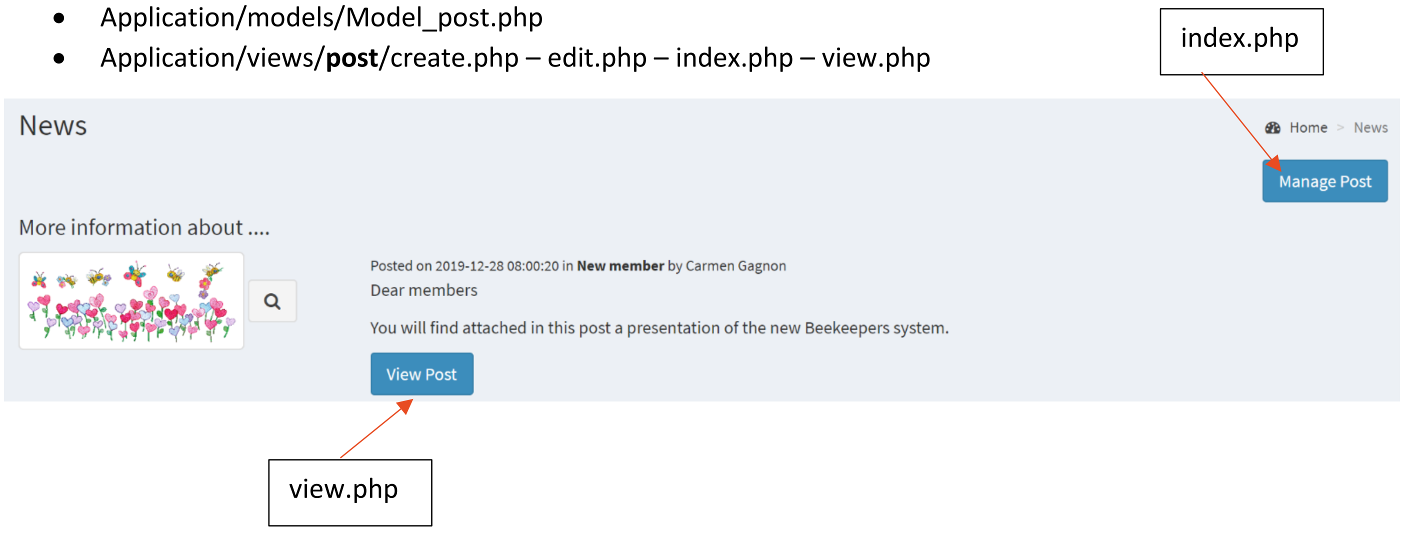
Dashboard and menu



News

MVC of News :

- Application/controllers/Post.php
- Application/models/Model_post.php
- Application/views/**post**/create.php – edit.php – index.php – view.php

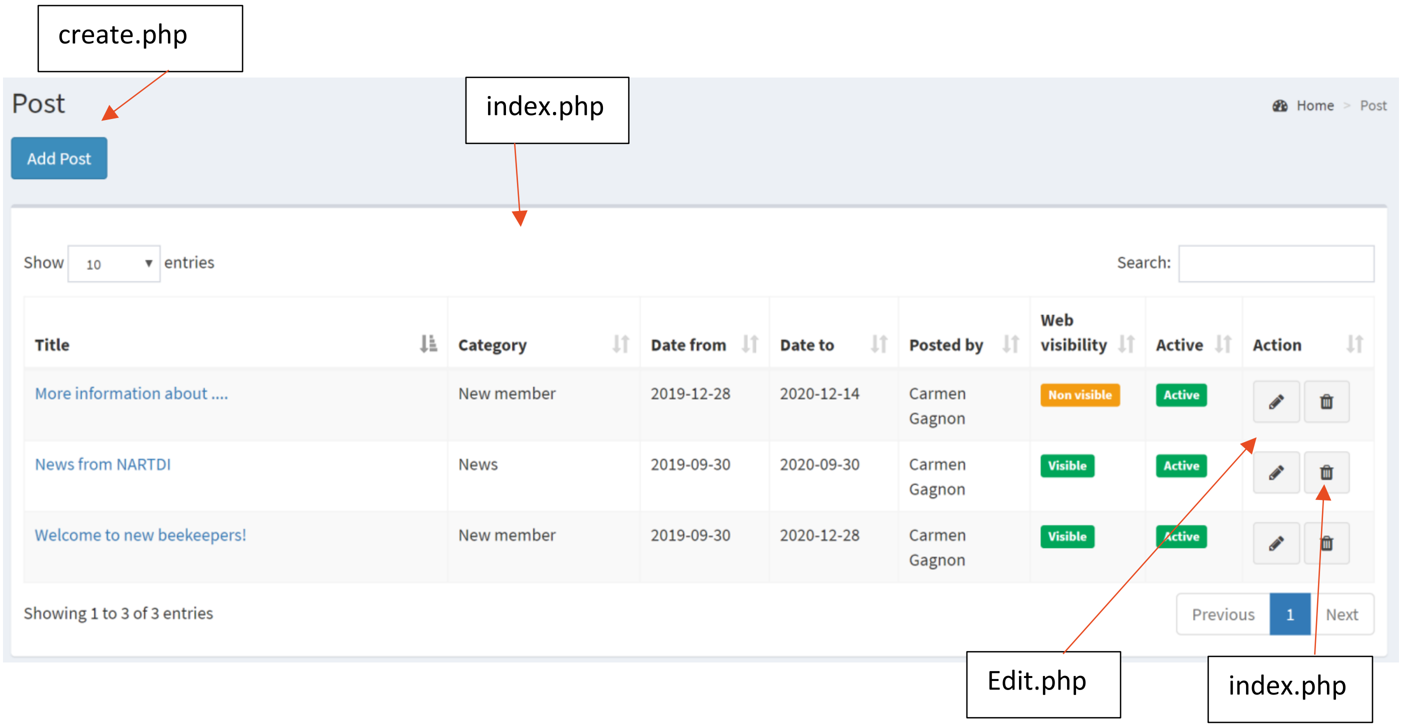


Posted on 2019-12-28 08:00:20 in **New member** by Carmen Gagnon

Dear members

You will find attached in this post a presentation of the new Beekeepers system.

The management of Post and Category are in the Settings of the system.



Post

Show 10 entries Search:

Title	Category	Date from	Date to	Posted by	Web visibility	Active	Action
More information about	New member	2019-12-28	2020-12-14	Carmen Gagnon	Non visible	Active	
News from NARTDI	News	2019-09-30	2020-09-30	Carmen Gagnon	Visible	Active	
Welcome to new beekeepers!	New member	2019-09-30	2020-12-28	Carmen Gagnon	Visible	Active	

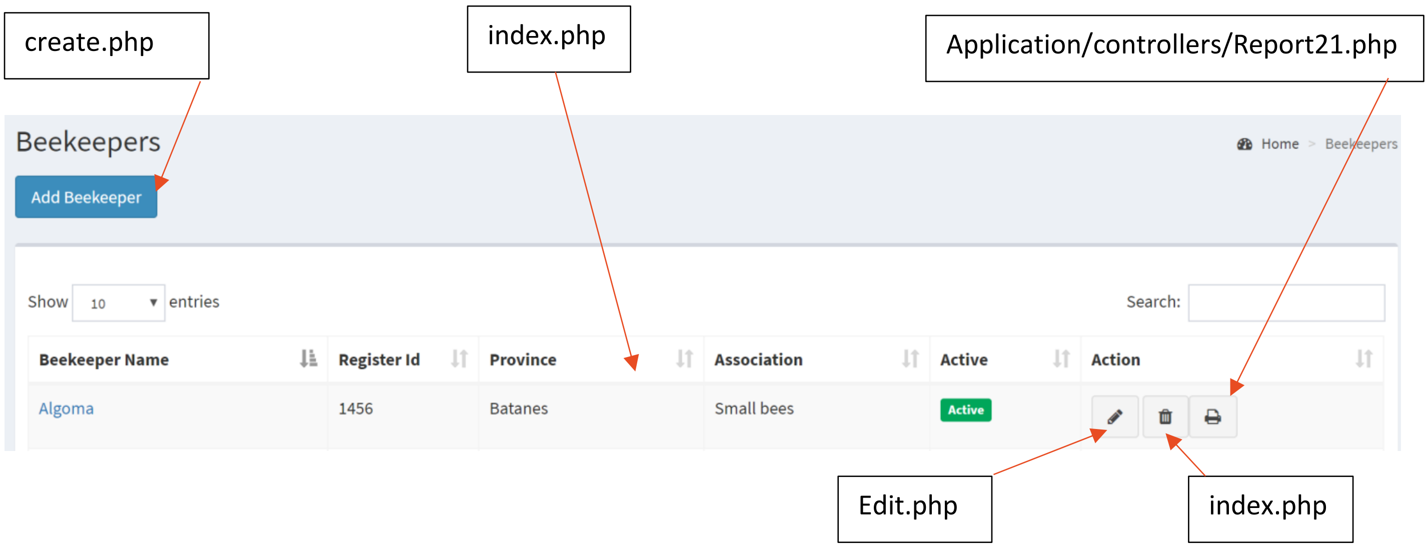
Showing 1 to 3 of 3 entries

Previous 1 Next

Beekeeper

MVC of Beekeeper :

- Application/controllers/Beekeeper.php
- Application/models/Model_beekeeper.php
- Application/views/**beekeeper**/create.php – edit.php – index.php

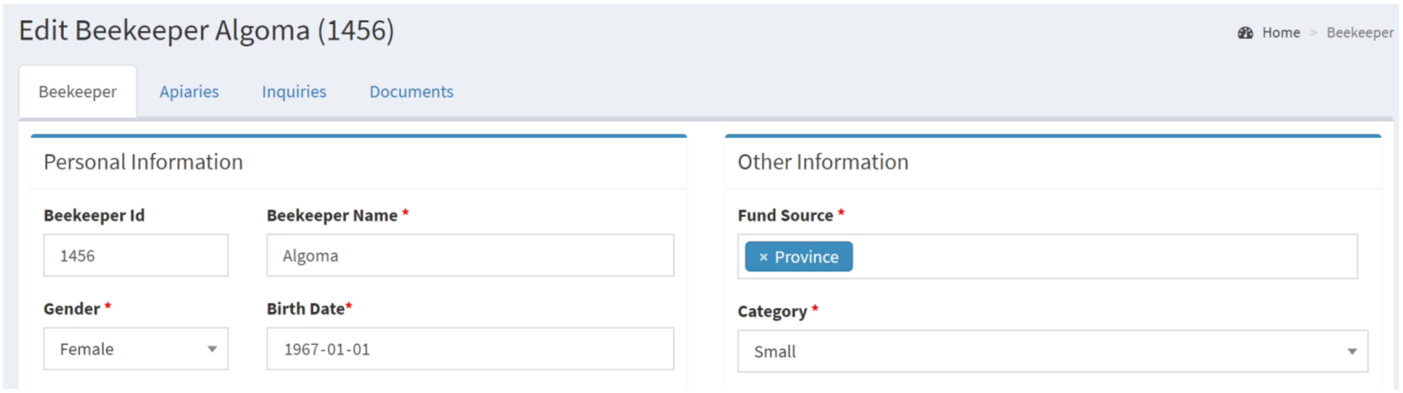


Beekeepers

Show 10 entries Search:

Beekeeper Name	Register Id	Province	Association	Active	Action
Algoma	1456	Batanes	Small bees	Active	

All the tabs of Edit a beekeeper are treated in a**pplication/views/beekeeper/edit.php** but might use model_apiary and model_colony and all the models for the creation of the drop-down list (model_nationality, model_fund_source, model_category, model_education, model_gender and model_region etc…).



Edit Beekeeper Algoma (1456)

Home > Beekeeper

Beekeeper | Apiaries | Inquiries | Documents

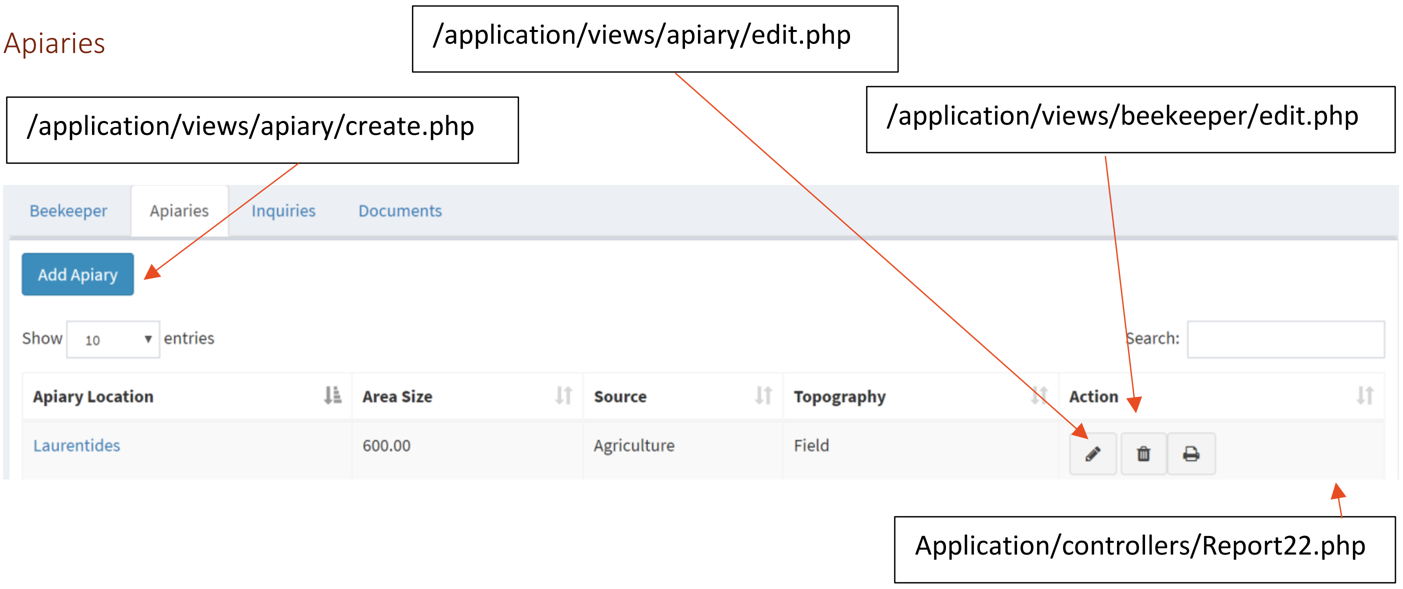
Personal Information

Beekeeper Id	1456
Beekeeper Name *	Algoma
Gender *	Female
Birth Date*	1967-01-01

Other Information

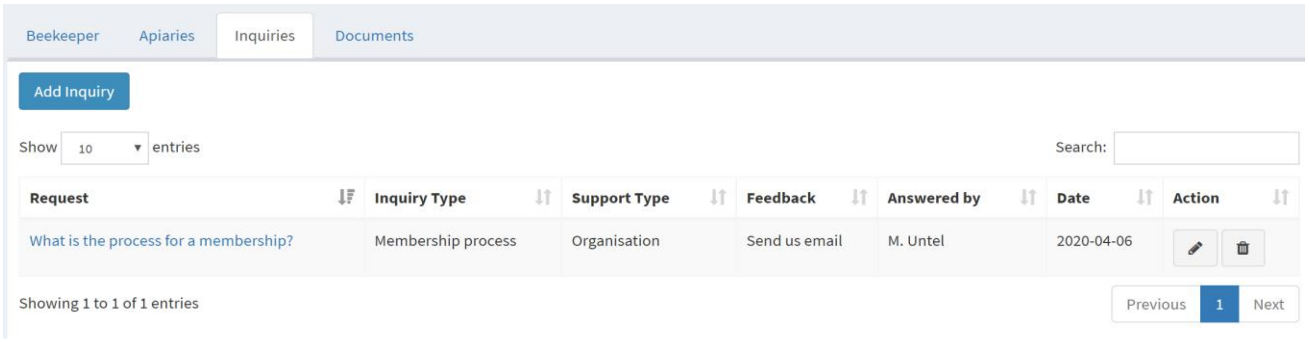
Fund Source *	× Province
Category *	Small

Apiaries


/application/views/apiary/edit.php
/application/views/apiary/create.php
/application/views/beekeeper/edit.php
Application/controllers/Report22.php
Beekeeper
Apiaries
Inquiries
Documents
Add Apiary
Show 10 entries
Search:

Apiary Location	Area Size	Source	Topography	Action
Laurentides	600.00	Agriculture	Field	

Inquiries

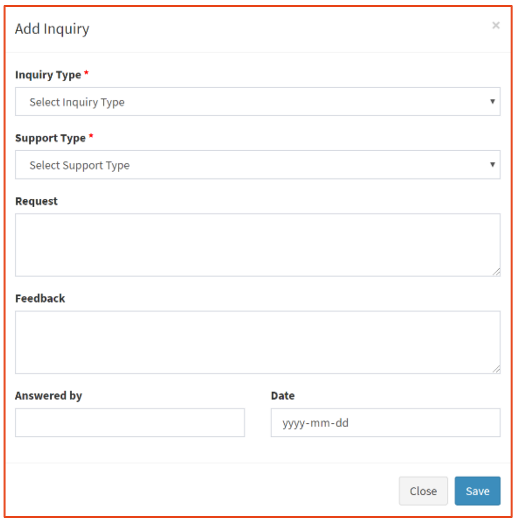


Request	Inquiry Type	Support Type	Feedback	Answered by	Date	Action
What is the process for a membership?	Membership process	Organisation	Send us email	M. Untel	2020-04-06	

Showing 1 to 1 of 1 entries

Previous 1 Next

List, add, update and delete of Inquiries are in **application/views/beekeeper/edit.php** but the treatment are in the controller Inquiry.php and model_inquiry

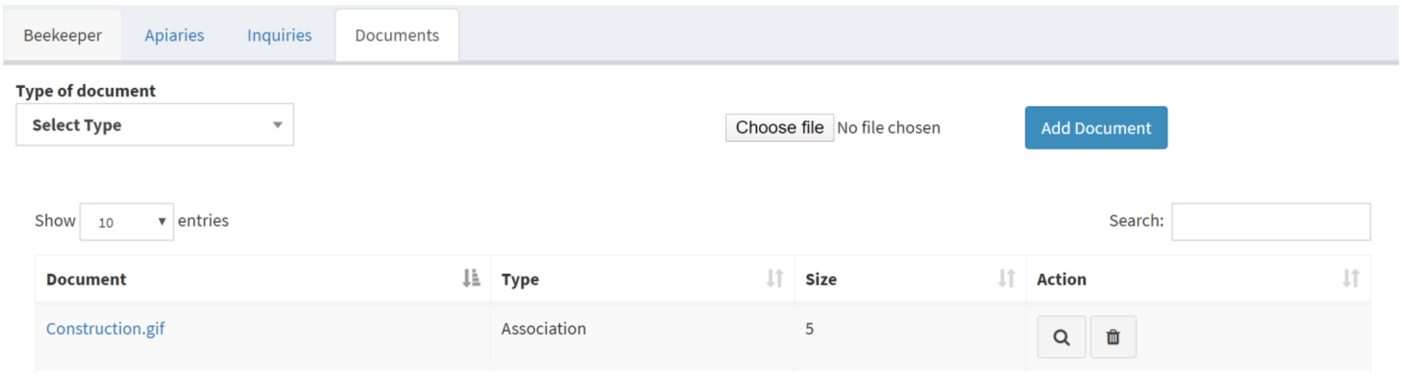


Add Inquiry

Inquiry Type *	Select Inquiry Type
Support Type *	Select Support Type
Request	
Feedback	
Answered by	
Date	yyyy-mm-dd

Documents

Documents are in **application/views/beekeeper/edit.php**. All the management of documents (upload, view and delete) are done in Document tab.

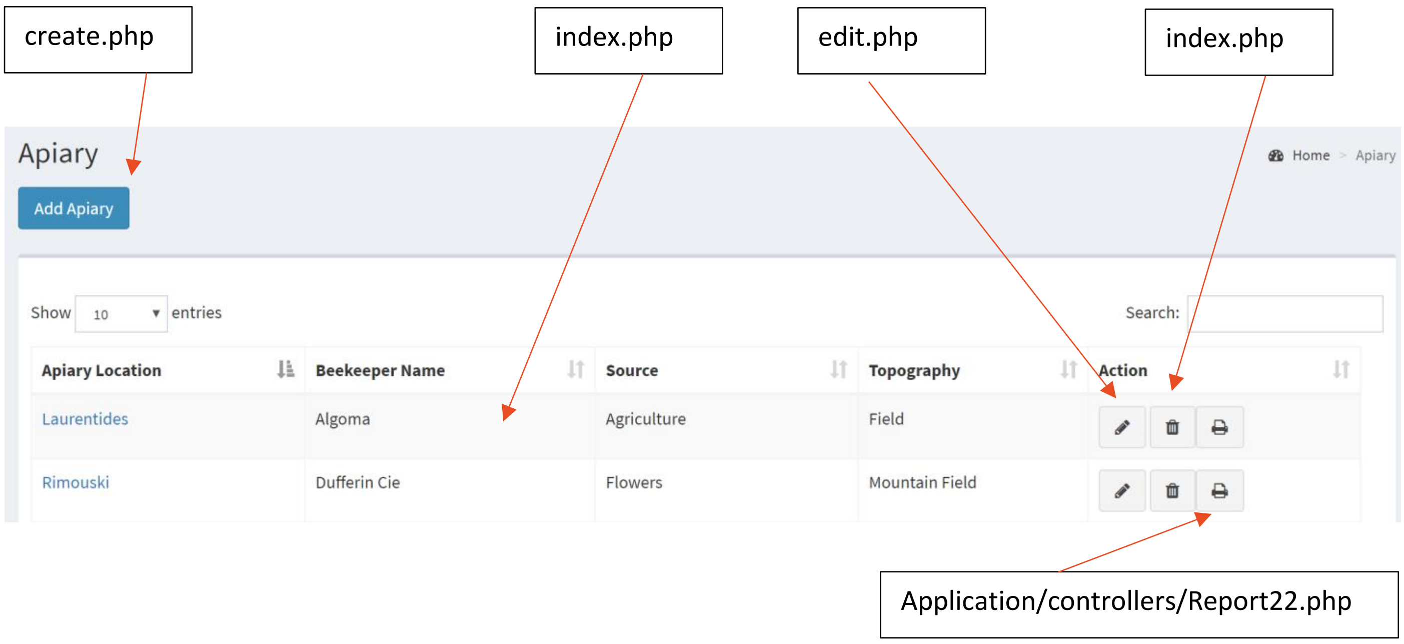


Document	Type	Size	Action
Construction.gif	Association	5	

Apiaries

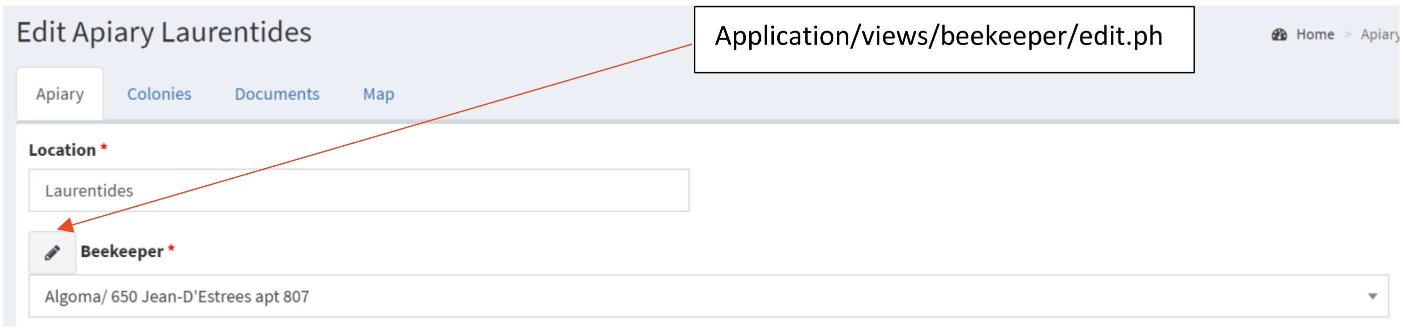
MVC of Apiary:

- Application/controllers/Apiary.php
- Application/models/Model_apiary.php
- Application/views/**apiary**/create.php – edit.php – index.php



Apiary Location	Beekeeper Name	Source	Topography	Action
Laurentides	Algoma	Agriculture	Field	
Rimouski	Dufferin Cie	Flowers	Mountain Field	

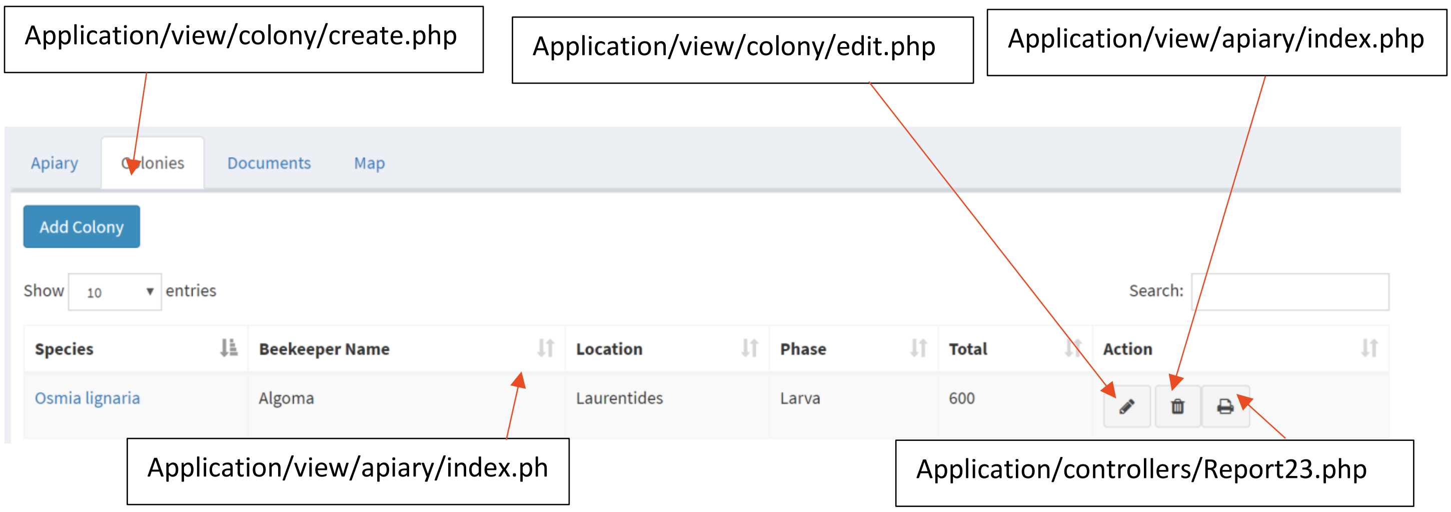
Other models are called for the creation of the drop-down list (model_source, model_beekeeper, model_topography, model_region, model_province, model_district).



Location *
Laurentides
Beekeeper *
Algoma/ 650 Jean-D'Estrees apt 807

Colonies

Some parts are treated in the MVC colony, but also in the MVC apiary


Application/view/colony/create.php
Application/view/colony/edit.php
Application/view/apiary/index.php
Apiary
Colonies
Documents
Map
Add Colony
Application/view/apiary/index.ph
Application/controllers/Report23.php

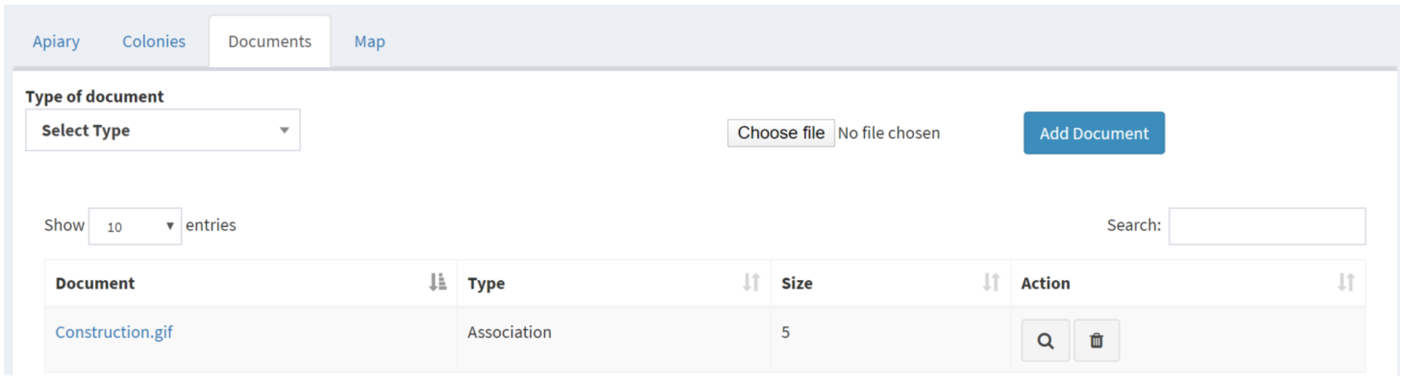
Show 10 entries

Search:

Species	Beekeeper Name	Location	Phase	Total	Action
Osmia lignaria	Algoma	Laurentides	Larva	600	

Documents

Documents are in **application/views/apiary/edit.php**. All the management of documents (upload, view and delete) are done in Document tab.



Type of document

Select Type

No file chosen

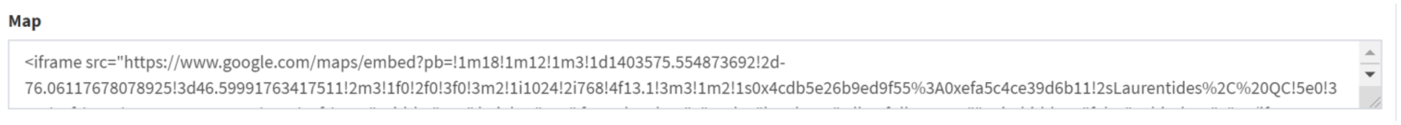
Show 10 entries

Search:

Document	Type	Size	Action
Construction.gif	Association	5	

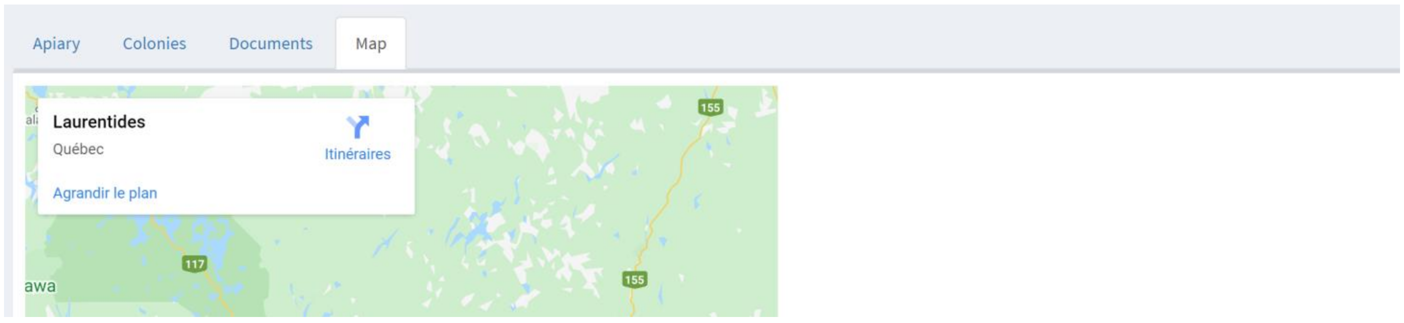
Map

The map is only a view of the link indicated in the field map in the edit part of the apiary.



Map

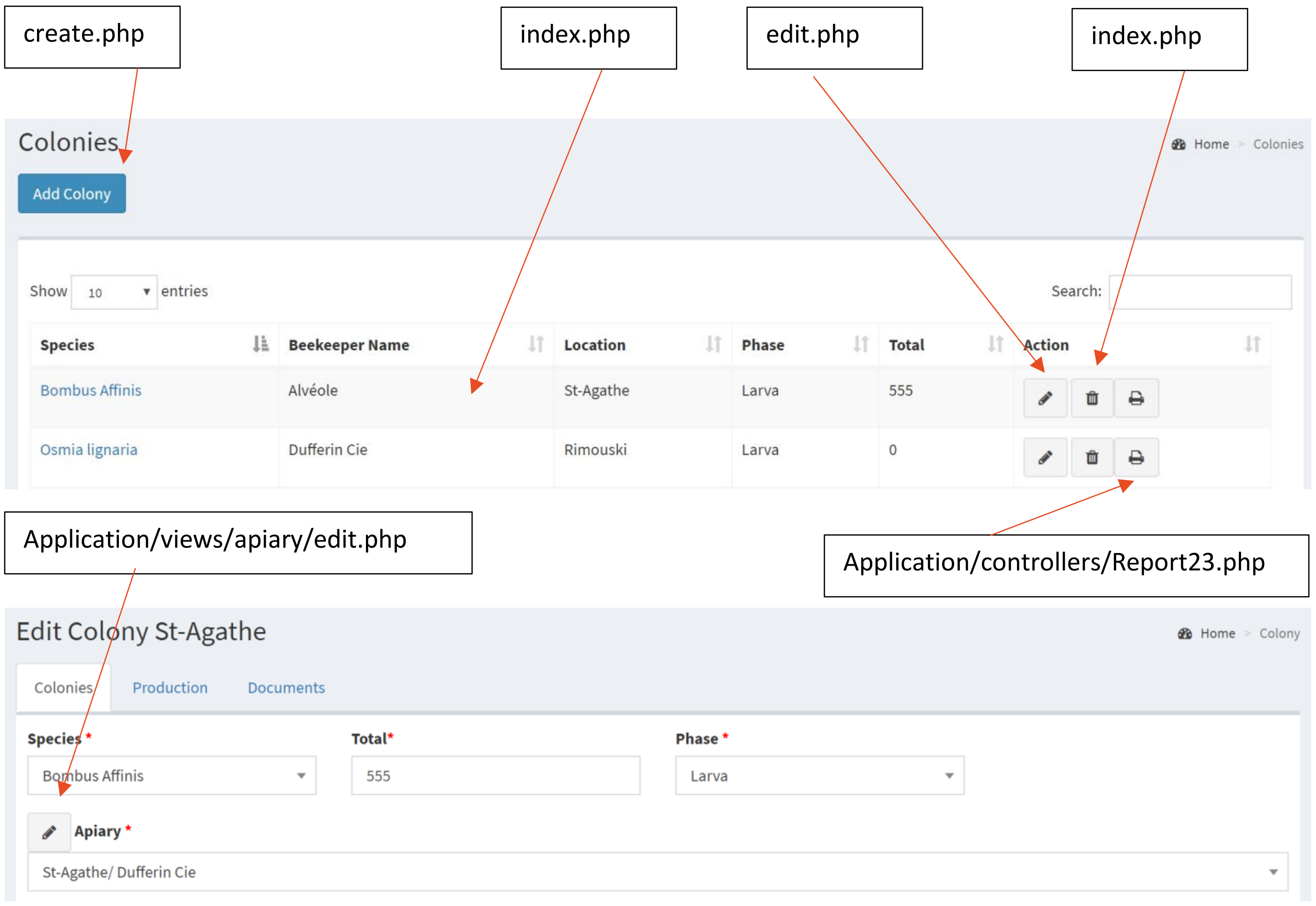
```
<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d1403575.554873692!2d-
76.06117678078925!3d46.59991763417511!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x4cdb5e26b9ed9f55%3A0xefa5c4ce39d6b11!2sLaurentides%2C%20QC!5e0!3
```



Colonies

MVC of Colony:

- Application/controllers/Colony.php
- Application/models/Model_colony.php
- Application/views/**colony**/create.php – edit.php – index.php



Colonies

Show 10 entries Search:

Species	Beekeeper Name	Location	Phase	Total	Action
Bombus Affinis	Alvéole	St-Agathe	Larva	555	
Osmia lignaria	Dufferin Cie	Rimouski	Larva	0	

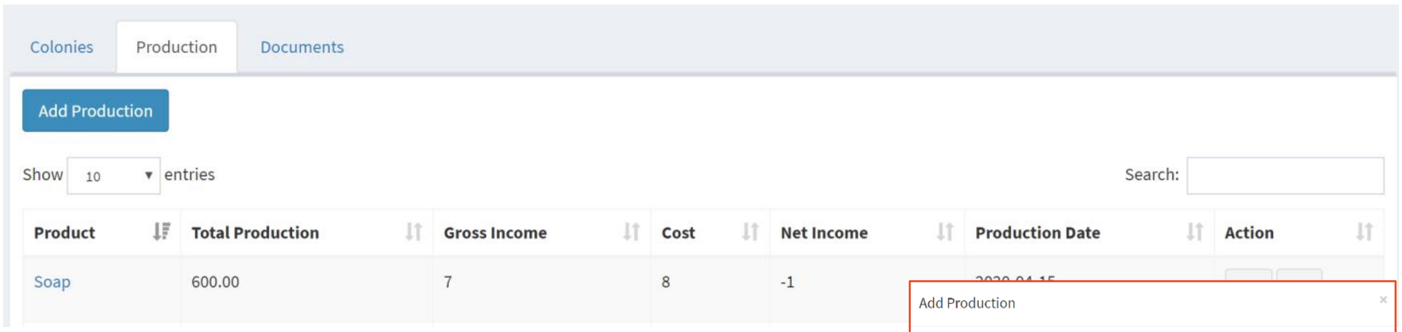
Edit Colony St-Agathe

Species *	Total*	Phase *
Bombus Affinis	555	Larva

Apiary *: St-Agathe/ Dufferin Cie

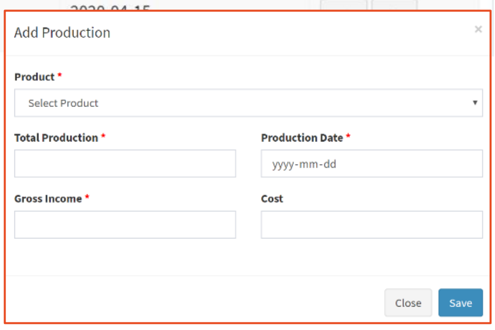
Production

Other models are called for the creation of the drop-down list (model_product, model_colony)



Show 10 entries Search:

Product	Total Production	Gross Income	Cost	Net Income	Production Date	Action
Soap	600.00	7	8	-1	2020-04-15	



Add Production

Product *	Select Product
Total Production *	
Production Date *	yyyy-mm-dd
Gross Income *	
Cost	

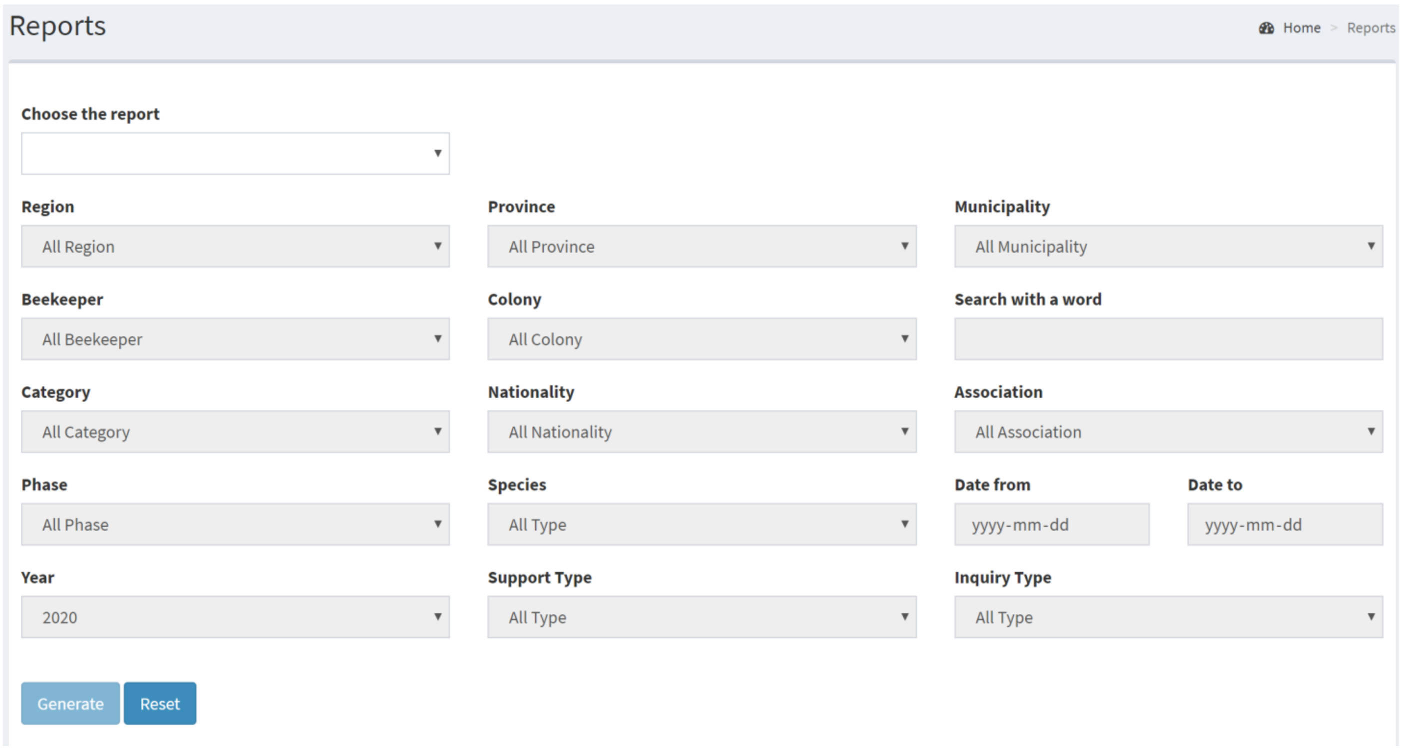
List, add, update and delete of Production are in **application/views/colony/edit.php** but the treatment are in the controller Production.php and model_production

Reports

MVC of Report:

- Application/controllers/Report.php
- Application/models/Model_report.php
- Application/views/**report**/index.php
- Each reports are in application/controllers/report01.php report02.php etc…
- Models used for the creation of all the dropdown list: model_province, model_district, model_beekeeper, model_municipality, model_species , model_phase, model_nationality etc…

Choose the report will open or not the parameters for each report. This is done in javascript found in application/views/report/index.php

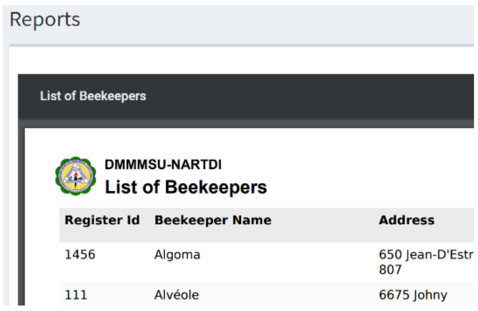


Reports

Home > Reports

Choose the report			
Region: All Region	**Province**: All Province	**Municipality**: All Municipality	
Beekeeper: All Beekeeper	**Colony**: All Colony	**Search with a word**:	
Category: All Category	**Nationality**: All Nationality	**Association**: All Association	
Phase: All Phase	**Species**: All Type	**Date from**: yyyy-mm-dd	**Date to**: yyyy-mm-dd
Year: 2020	**Support Type**: All Type	**Inquiry Type**: All Type	

Generate the report submits the form /application/views/report/index.php to compose and print the appropriate report found in /appplication/controllers/reportxx.php. It opens the pdf viewer of the browser.



Reports

List of Beekeepers

DMMMSU-NARTDI
List of Beekeepers

Register Id	Beekeeper Name	Address
1456	Algoma	650 Jean-D'Estr 807
111	Alvéole	6675 Johny

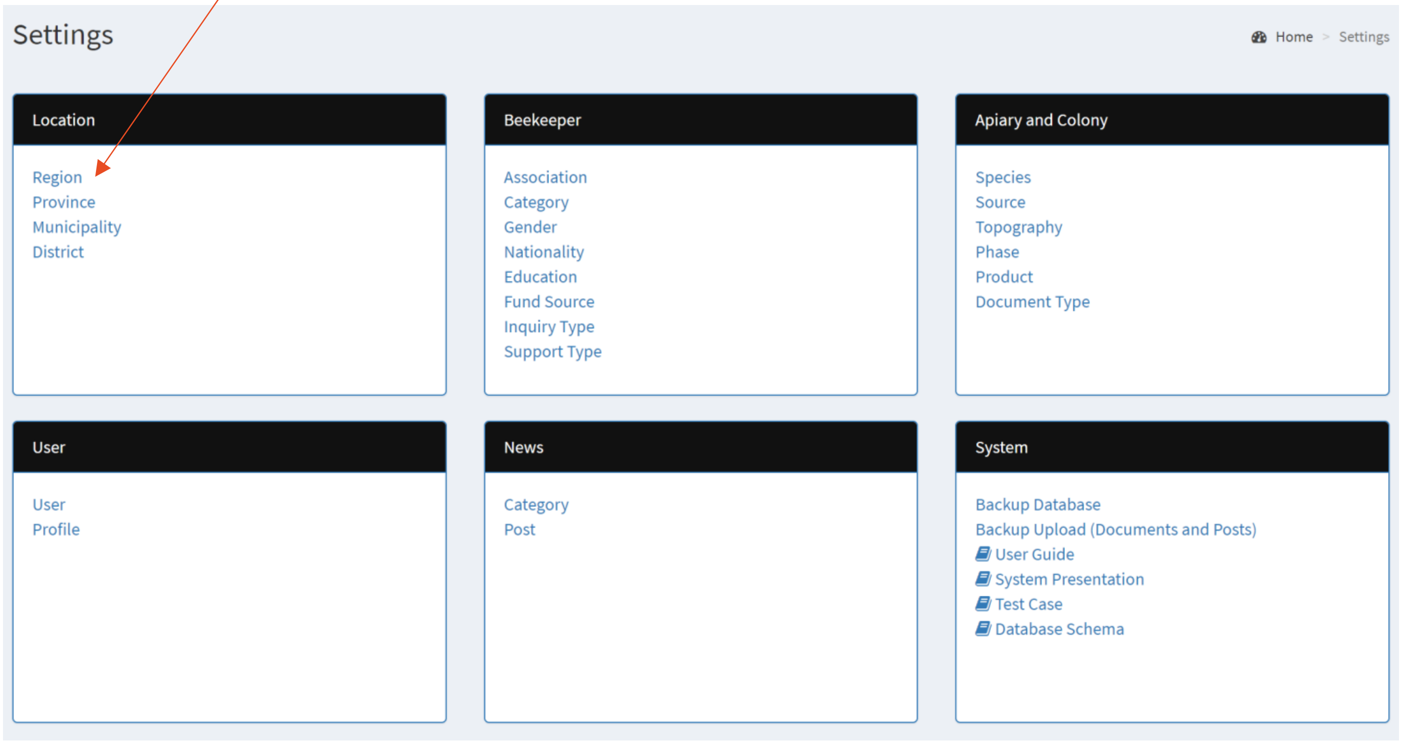
```
<!-------------------------------------------------  P R I N T   R E P O R T S  ------------------------------------------ -->
        <?php if($this->session->printREP01 == 'yes') : ?>
          <object data="<?php echo  base_url("report01/REP01"); ?>" width="100%" height="1000px" type="application/pdf"> </object>
        <?php endif; ?>

        <?php if($this->session->printREP02 == 'yes') : ?>
          <object data="<?php echo  base_url("report02/REP02"); ?>" width="100%" height="1000px" type="application/pdf"> </object>
        <?php endif; ?>
```

Settings

MVC of Settings:

- Application/controllers/Setting.php
- Application/views/**setting**/index.php
- There is not model for settings because it's only commands to call the MVC of each settings.
- Each command have the same treatment. You will find a more detailed examples in the section Example MVC.

Application/views/region/index.php



Documentation

Documentation is calling the user guide with the pdf Viewer. The user guide are in:

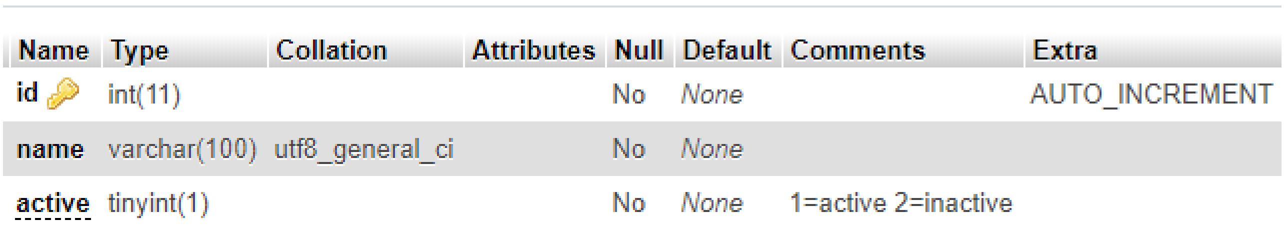
/assets/documentation/meb_user_guide_fr.pdf or meb_user_guide_en.pdf depending on the language of the user.

You will find other documentation related to the system in this directory but not available for users. Only for the super administrator of the system.

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)

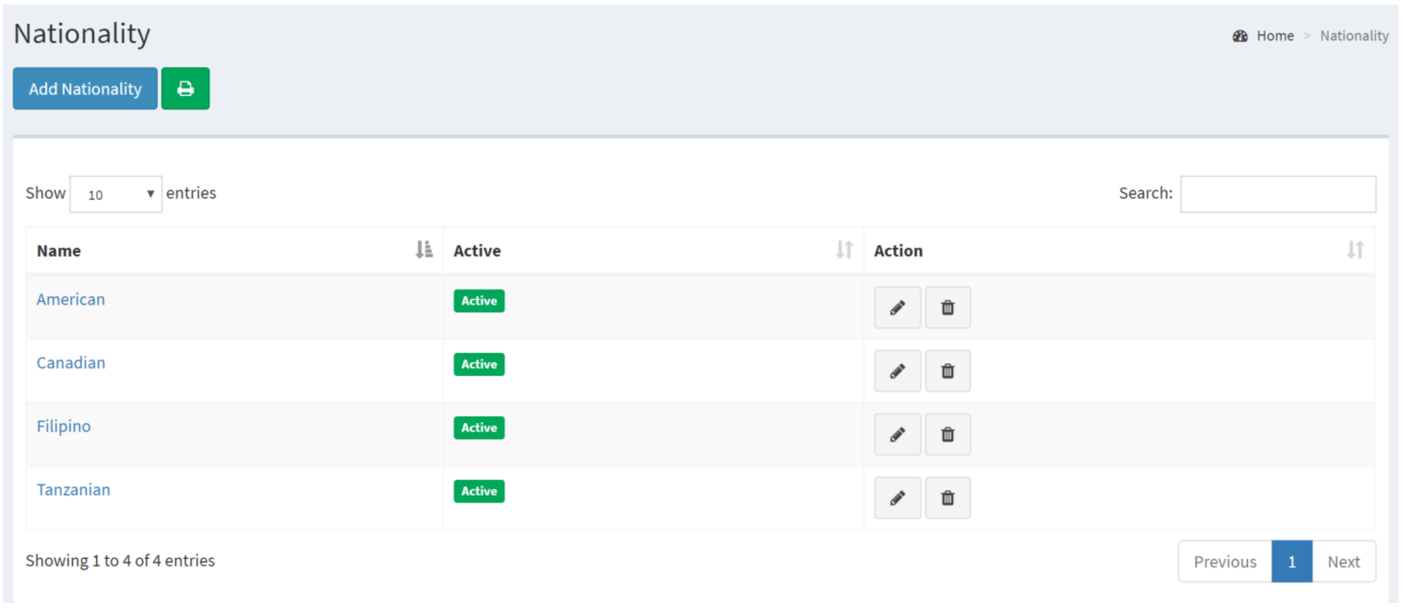
Here is the structure of the table:



Name	Type	Collation	Attributes	Null	Default	Comments	Extra
id	int(11)			No	*None*		AUTO_INCREMENT
name	varchar(100)	utf8_general_ci		No	*None*		
active	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.



Nationality

Home > Nationality

Add Nationality

Show 10 entries Search:

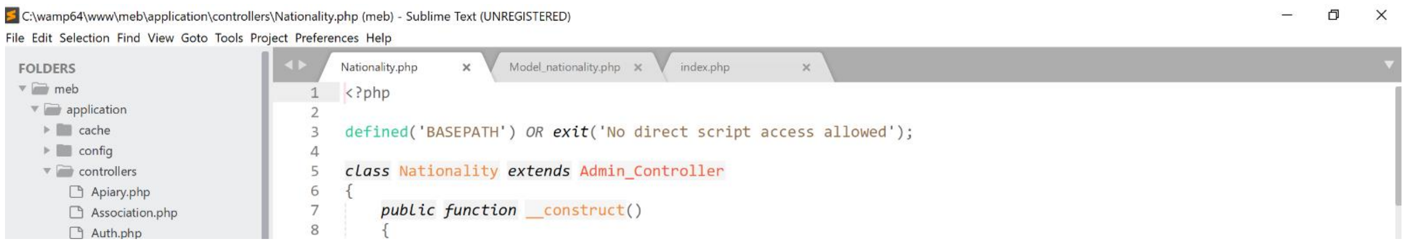
Name	Active	Action
American	Active	
Canadian	Active	
Filipino	Active	
Tanzanian	Active	

Showing 1 to 4 of 4 entries

Previous 1 Next

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.

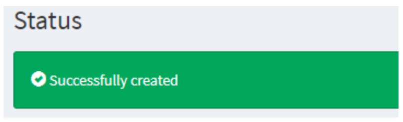
To understand the whole process, open the controller Nationality.php, the model_nationality and the view nationality/index.php in your editor.



C:\wamp64\www\meb\application\controllers\Nationality.php (meb) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
meb
application
cache
config
controllers
Apiary.php
Association.php
Auth.php
Nationality.php
Model_nationality.php
index.php

```php
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Nationality extends Admin_Controller
{
    public function __construct()
    {
```

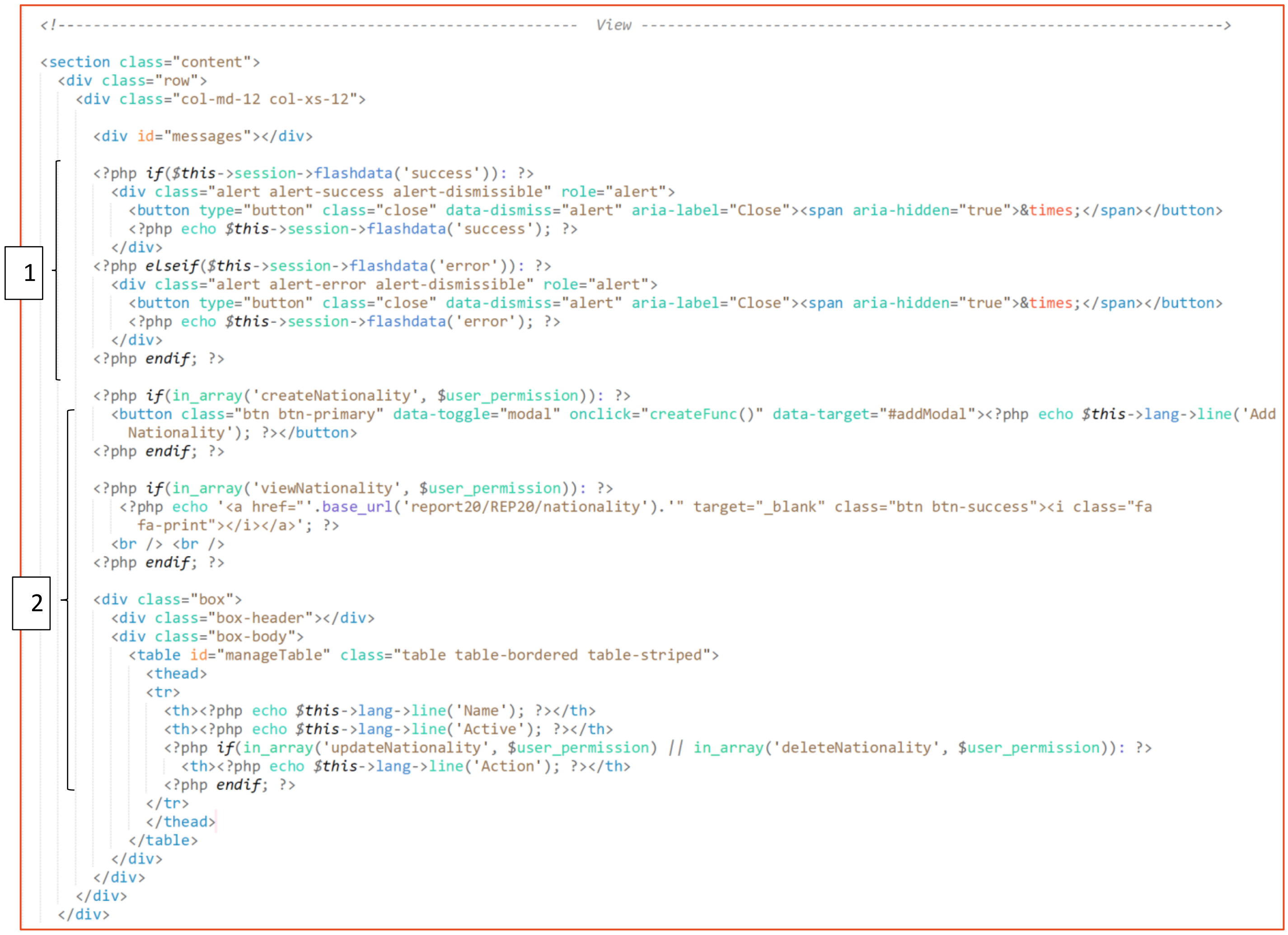
1. First we evaluate if the last submit of the form had generate an error message or a confirmation of the success of the operation. The message will be presented to the user.



Status

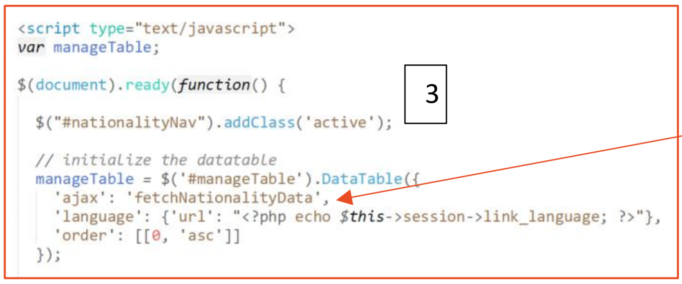
Successfully created

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>
```

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

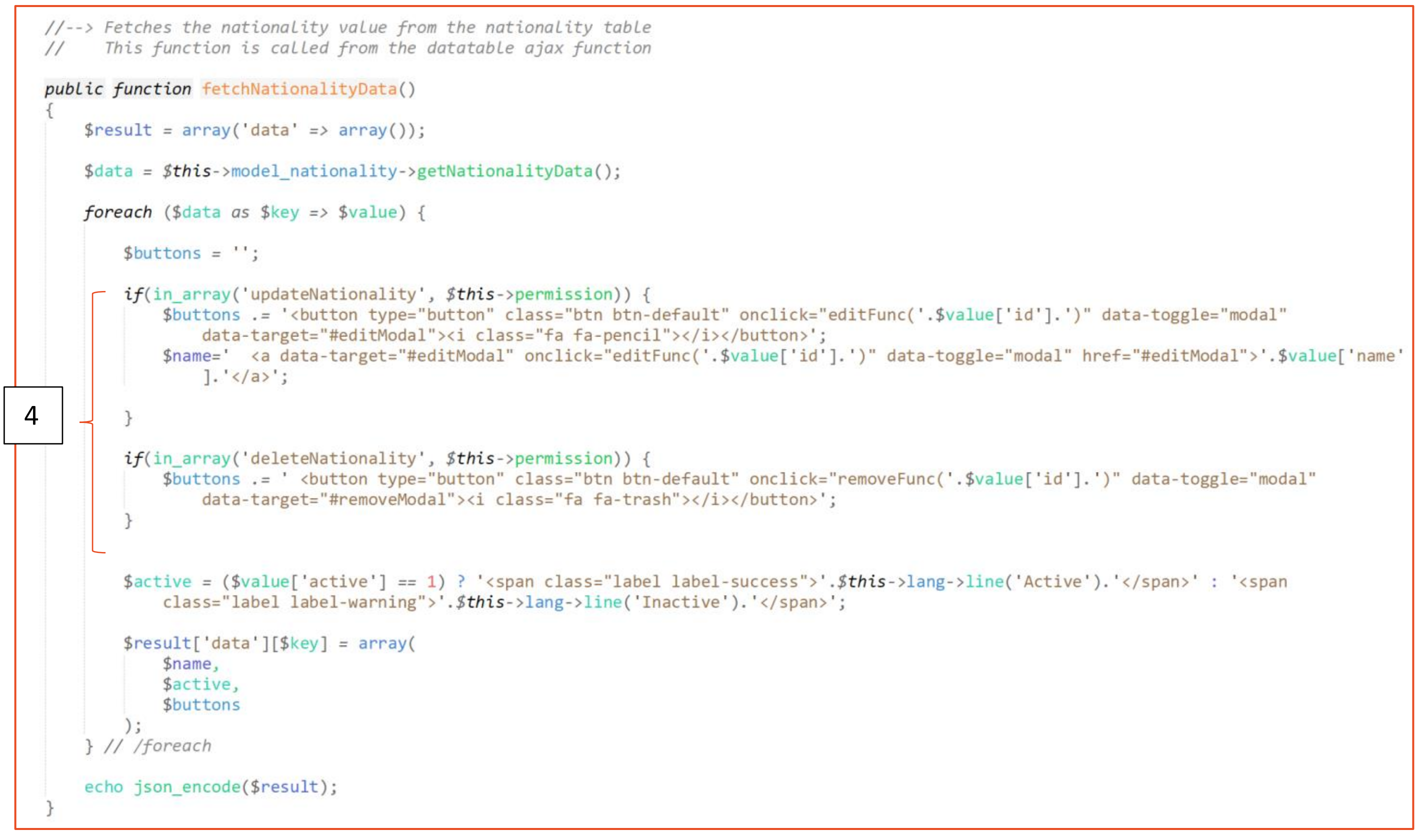


```
<script type="text/javascript">
var manageTable;

$(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']]
  });
```

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.

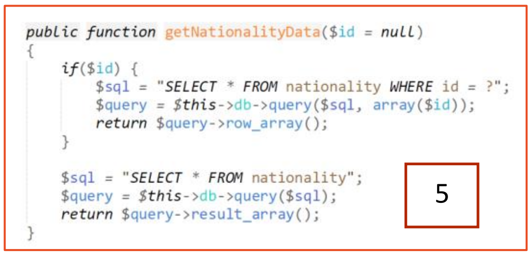
4. Controller Nationality.php: This is where the function **getNationalityData** is called. You can see how the security is build giving permission or not to update or delete the entry in the table. You will find the function getNationalityData in /application/models/Model_nationality.php. The datas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



4

```php
//--> 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);
}
```

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.

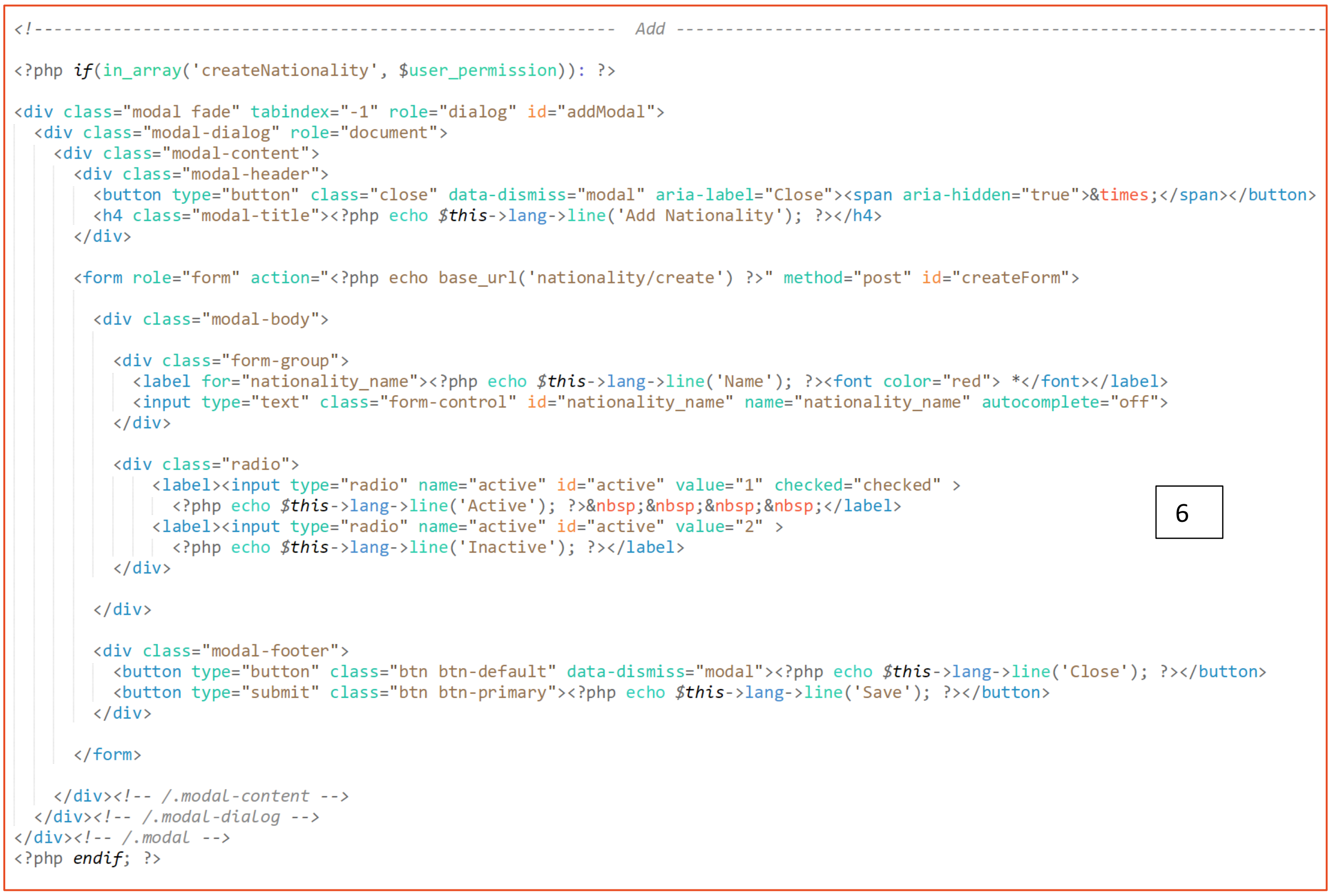


```php
public function getNationalityData($id = null)
{
    if($id) {
        $sql = "SELECT * FROM nationality WHERE id = ?";
        $query = $this->db->query($sql, array($id));
        return $query->row_array();
    }

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

5

Add nationality

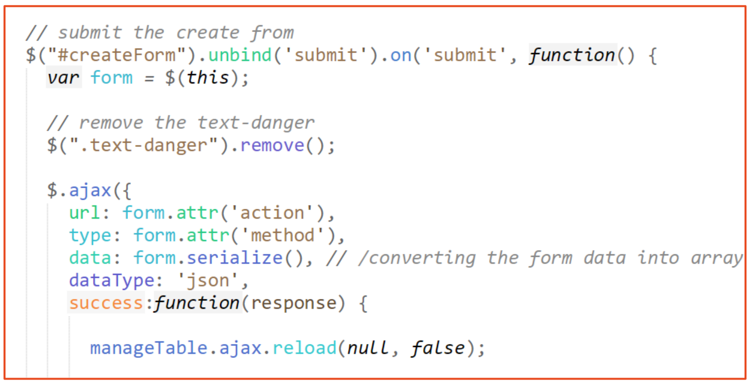
6. For adding a nationality, we will use the createForm of Ajax/Javascript and submit the form calling the function **create** in the controller nationality.php. Here is the view for adding a nationality.



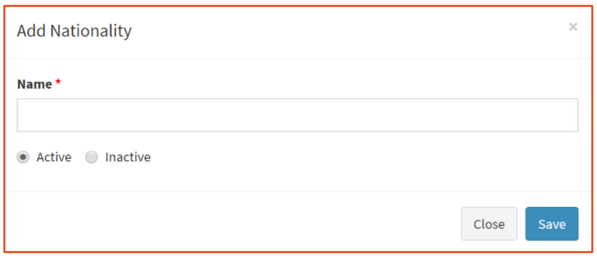
```
<!------------------------------------------------------------ Add ------------------------------------------------------------------

<?php if(in_array('createNationality', $user_permission)): ?>

<div class="modal fade" tabindex="-1" role="dialog" id="addModal">
  <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('Add Nationality'); ?></h4>
      </div>

      <form role="form" action="<?php echo base_url('nationality/create') ?>" method="post" id="createForm">

        <div class="modal-body">

          <div class="form-group">
            <label for="nationality_name"><?php echo $this->lang->line('Name'); ?><font color="red"> *</font></label>
            <input type="text" class="form-control" id="nationality_name" name="nationality_name" autocomplete="off">
          </div>

          <div class="radio">
              <label><input type="radio" name="active" id="active" value="1" checked="checked" >
                <?php echo $this->lang->line('Active'); ?>&nbsp;&nbsp;&nbsp;&nbsp;</label>
              <label><input type="radio" name="active" id="active" value="2" >
                <?php echo $this->lang->line('Inactive'); ?></label>
          </div>

        </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('Save'); ?></button>
        </div>

      </form>

    </div><!-- /.modal-content -->
  </div><!-- /.modal-dialog -->
</div><!-- /.modal -->
<?php endif; ?>
```

6

CreateForm javascript.



```
// submit the create from
$("#createForm").unbind('submit').on('submit', function() {
  var form = $(this);

  // remove the text-danger
  $(".text-danger").remove();

  $.ajax({
    url: form.attr('action'),
    type: form.attr('method'),
    data: form.serialize(), // /converting the form data into array
    dataType: 'json',
    success:function(response) {

      manageTable.ajax.reload(null, false);
```



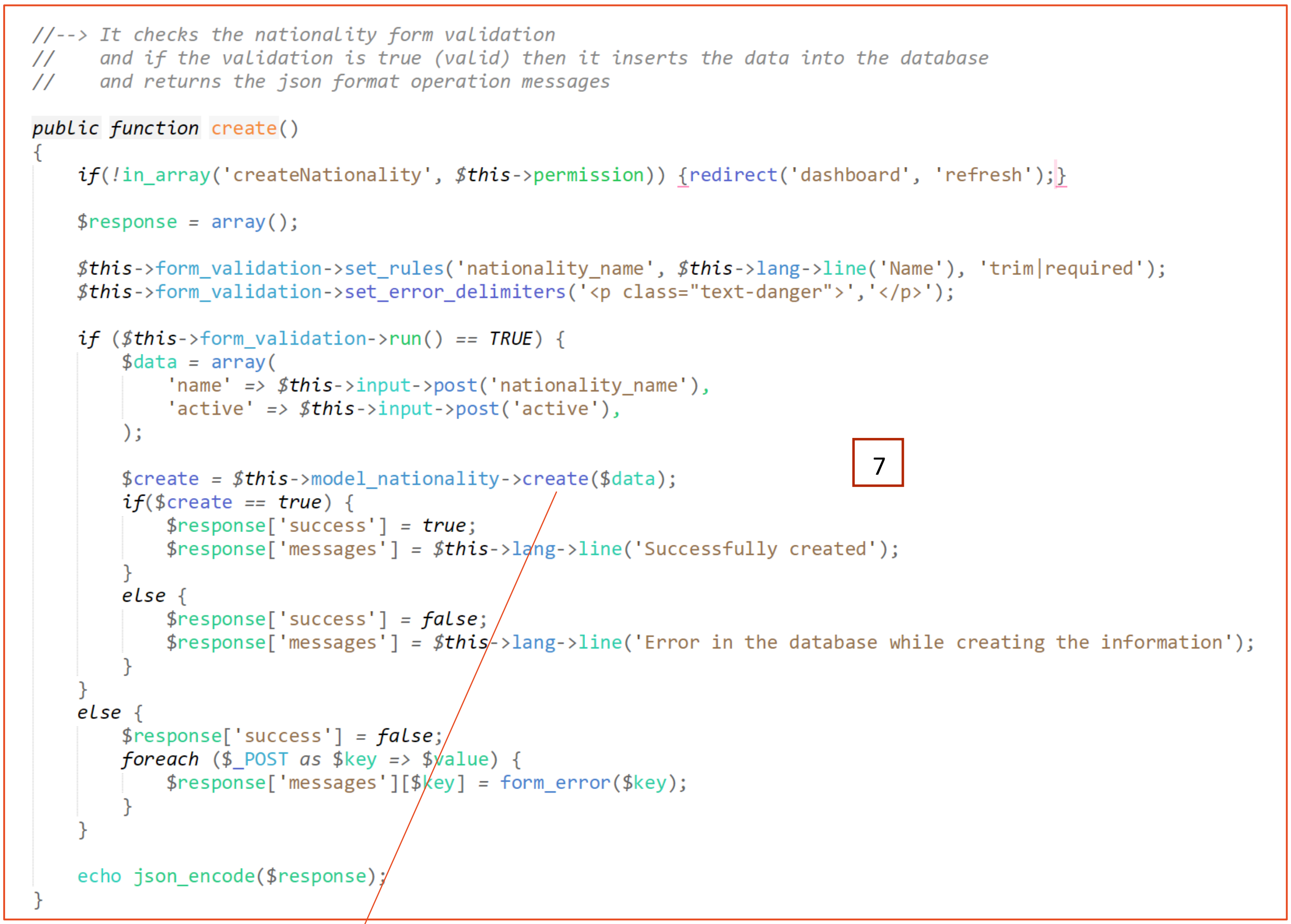
Add Nationality ×

Name *

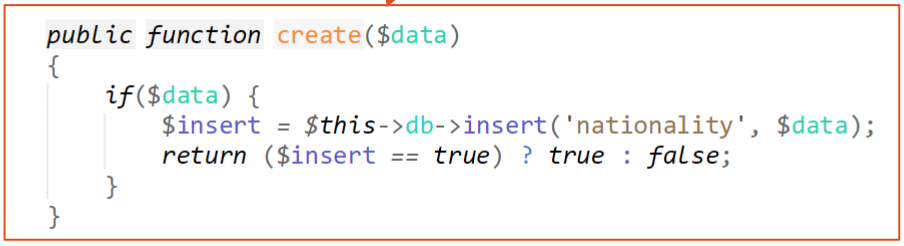
● Active ○ Inactive

Close Save

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



```php
//--> 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);
}
```



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

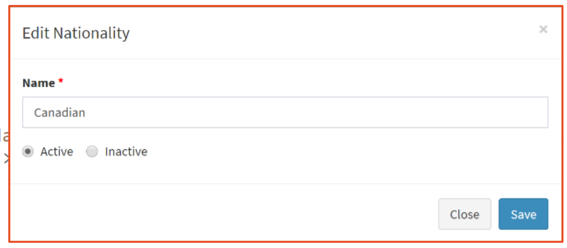
Update nationality

8. The process will be pratically the same for the edit (update) of the nationality. Javascript editModal will manage the update.
9. The submit at the end will call the function **update** of the controller Nationality.php

```
<!------------------------------------------------------------  Edit ----------------------------------------------------------------------------

<?php if(in_array('updateNationality', $user_permission)): ?>
<!-- edit modal -->
<div class="modal fade" tabindex="-1" role="dialog" id="editModal">
  <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('Edit Nationality'); ?></h4>
      </div>

      <form role="form" action="<?php echo base_url('nationality/update') ?>" method="post" id="updateForm">

        <div class="modal-body">
          <div id="messages"></div>

          <div class="form-group">
            <label for="edit_nationality_name"><?php echo $this->lang->line('Name'); ?><font color="red"> *</font></label>
            <input type="text" class="form-control" id="edit_nationality_name" name="edit_nationality_name" autocomplete="off">
          </div>

          <div class="radio">
              <label><input type="radio" name="edit_active" id="edit_active" value="1" >
                <?php echo $this->lang->line('Active'); ?>&nbsp;&nbsp;&nbsp;&nbsp;</label>
              <label><input type="radio" name="edit_active" id="edit_inactive" value="2" >
                <?php echo $this->lang->line('Inactive'); ?></label>
          </div>


        </div>

        <div class="modal-footer">
          <button type="button" class="btn btn-default" data-dismiss="moda
          <button type="submit" class="btn btn-primary"><?php echo $this->
        </div>

      </form>
    </div><!-- /.modal-content -->
  </div><!-- /.modal-dialog -->
</div><!-- /.modal -->
<?php endif; ?>
```

8

9


Edit Nationality

Name *

Canadian

● Active ○ Inactive

10. The form should be filled with the information from the database and the function **fetchNationalityDataById** will be called from the controller nationality.php, javascript modal passing the id from the list to the function **getNationalityData** in the model_nationality

/application/views/nationality/ Index.php – editFunc

```
// edit function
function editFunc(id)
{
  $("#updateForm")[0].reset();
  $("#updateForm .form-group").removeClass('has-error').removeClas
  $(".text-danger").remove();
  $.ajax({
    url: 'fetchNationalityDataById/'+id,
    type: 'post',
    dataType: 'json',
    success:function(response) {

      $("#edit_nationality_name").val(response.name);
      if(response.active==1){
          $('input:radio[id=edit_active]')[0].checked = true;
          $('input:radio[id=edit_inactive]')[0].checked = false;
        }else{
          $('input:radio[id=edit_active]')[0].checked = false;
          $('input:radio[id=edit_inactive]')[0].checked = true;
        }
```

10

/application/controllers/ nationality.php

```
public function fetchNationalityDataById($id)
{
    if($id) {
        $data = $this->model_nationality->getNationalityData($id);
        echo json_encode($data);
    }

    return false;
}
```

/application/models/ model_nationality.php

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

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

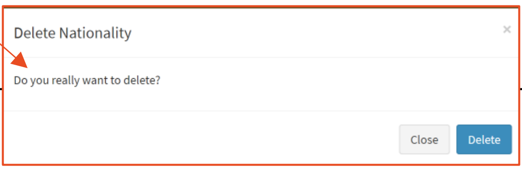
Delete nationality

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


Delete Nationality

Do you really want to delete?

```
// 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 apiary 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 apiary 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();

}
```

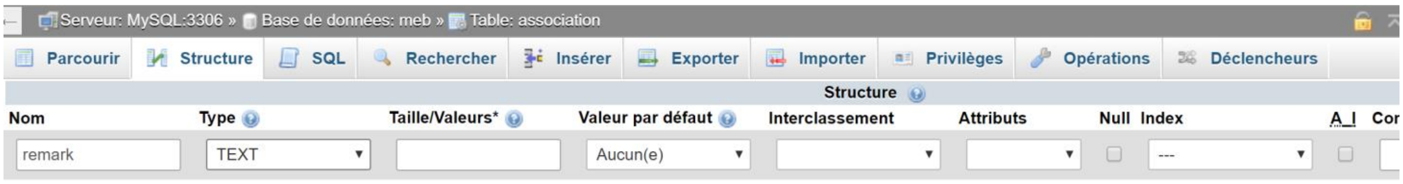
14

How to add a new field

This is the method for adding a new field in the database and in the system. The example here is to add the field remark to the table association.

Create in the table

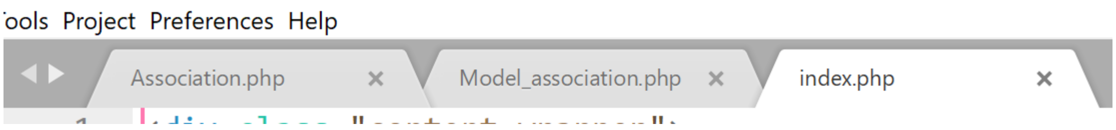
Add the field **remark** to the table **association** with phpMyAdmin. You add the field in alphabetical order when possible.



Nom	Type	Taille/Valeurs*	Valeur par défaut	Interclassement	Attributs	Null	Index	A_I	Cor
remark	TEXT		Aucun(e)			☐	---	☐	

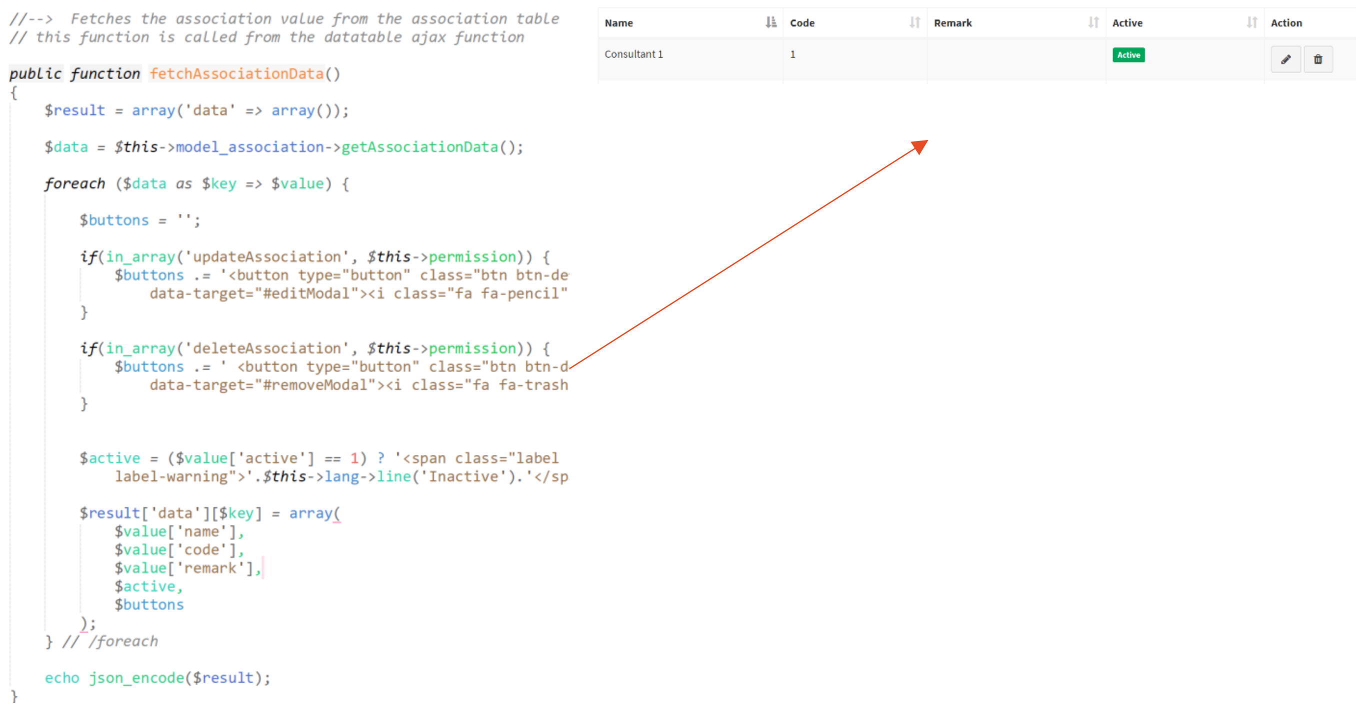
Change the controller – model and views

The treatment of association (add, edit, delete) is called in the view Index of Association. Open controller Association.php, model_association and views/association/index.php



Modification of controller Association.php.

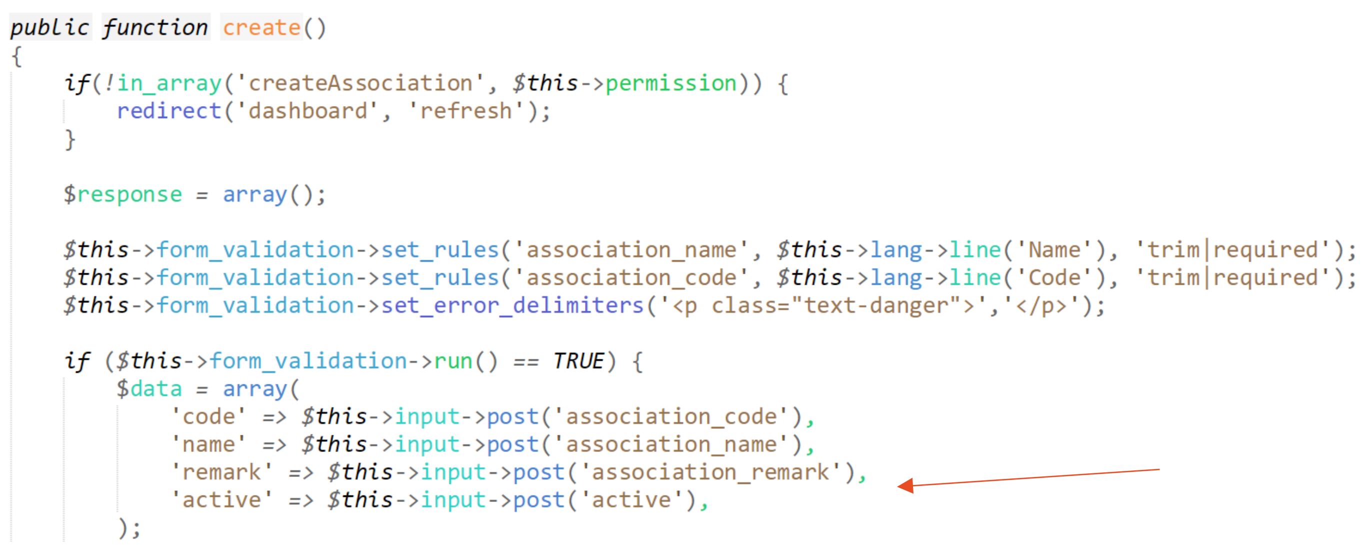
We must add the field remark in the list of association that appear in Association form. In the function fetchAssociationData



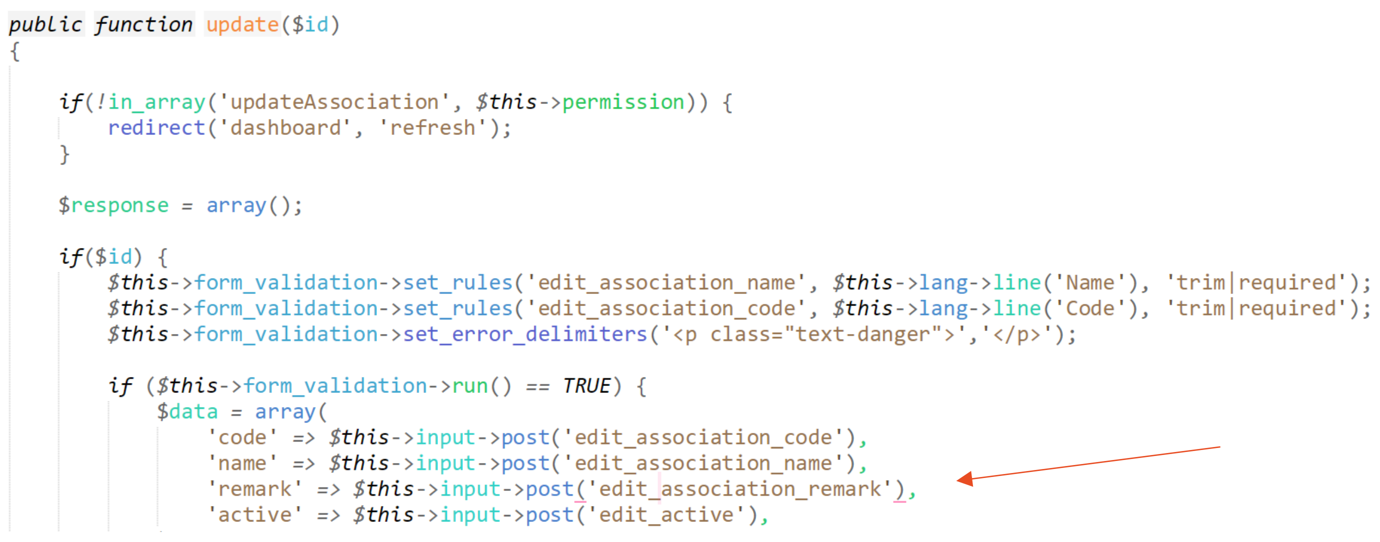
```
//-->  Fetches the association value from the association table
// this function is called from the datatable ajax function

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

    $data = $this->model_association->getAssociationData();

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

        $buttons = '';

        if(in_array('updateAssociation', $this->permission)) {
            $buttons .= '<button type="button" class="btn btn-de
                data-target="#editModal"><i class="fa fa-pencil"
        }

        if(in_array('deleteAssociation', $this->permission)) {
            $buttons .= ' <button type="button" class="btn btn-d
                data-target="#removeModal"><i class="fa fa-trash
        }


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

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

    echo json_encode($result);
}
```

Name	Code	Remark	Active	Action
Consultant 1	1		Active	

Add the new field in the functions Create and Update.



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

    $response = array();

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

    if ($this->form_validation->run() == TRUE) {
        $data = array(
            'code' => $this->input->post('association_code'),
            'name' => $this->input->post('association_name'),
            'remark' => $this->input->post('association_remark'),
            'active' => $this->input->post('active'),
        );
```



```
public function update($id)
{

    if(!in_array('updateAssociation', $this->permission)) {
        redirect('dashboard', 'refresh');
    }

    $response = array();

    if($id) {
        $this->form_validation->set_rules('edit_association_name', $this->lang->line('Name'), 'trim|required');
        $this->form_validation->set_rules('edit_association_code', $this->lang->line('Code'), 'trim|required');
        $this->form_validation->set_error_delimiters('<p class="text-danger">','</p>');

        if ($this->form_validation->run() == TRUE) {
            $data = array(
                'code' => $this->input->post('edit_association_code'),
                'name' => $this->input->post('edit_association_name'),
                'remark' => $this->input->post('edit_association_remark'),
                'active' => $this->input->post('edit_active'),
```

Modification of the model

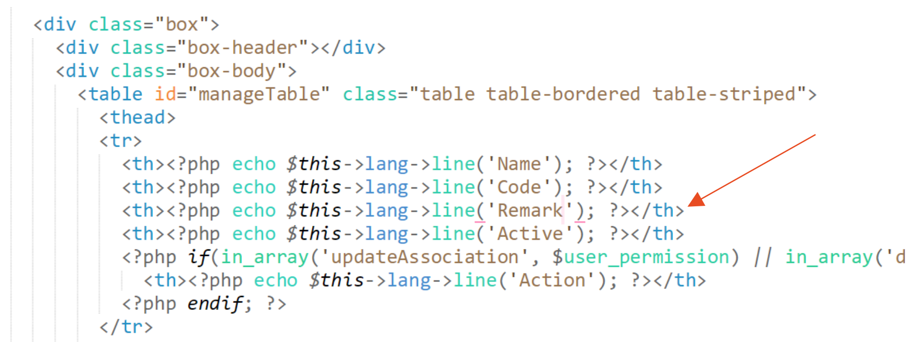
Verify if the new field might be involved in the treatment of model_association.php. The query is SELECT * so automatically the remark will be available in the recordset

NO MODIFICATION TO DO

Modification of the view in Association

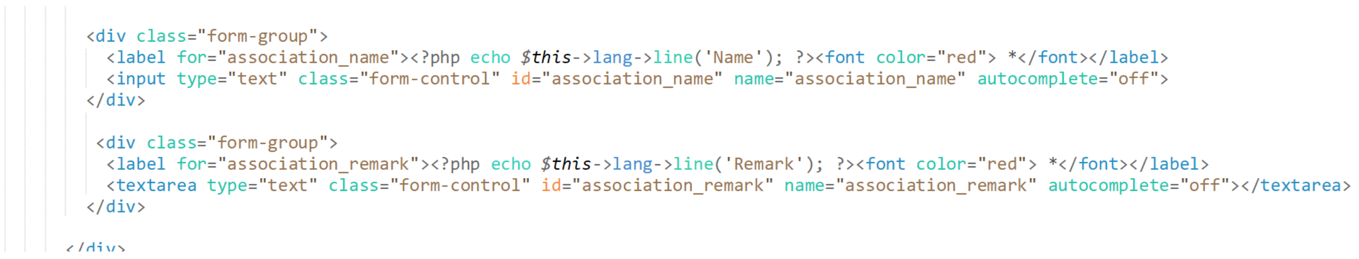
Find the jquery modal for performance in the views/association/index.php

In the list, we must add the remark



```
<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('Code'); ?></th>
        <th><?php echo $this->lang->line('Remark'); ?></th>
        <th><?php echo $this->lang->line('Active'); ?></th>
        <?php if(in_array('updateAssociation', $user_permission) || in_array('d
          <th><?php echo $this->lang->line('Action'); ?></th>
        <?php endif; ?>
      </tr>
```

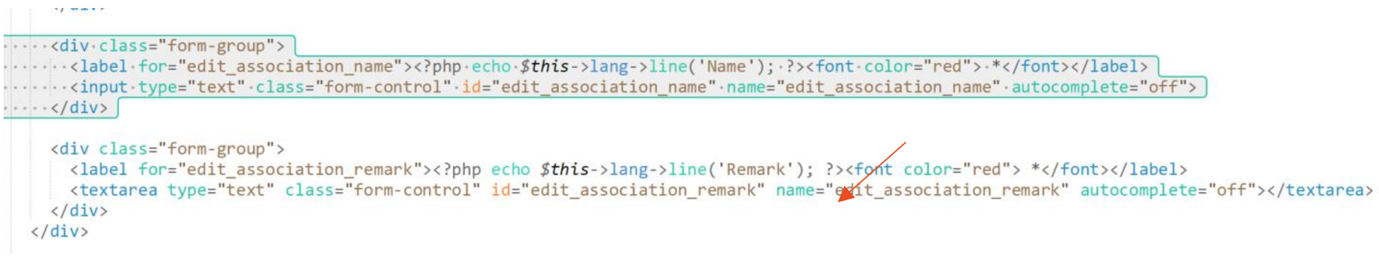
In the create, a textarea field for remark must be added after the name of the association



```
    <div class="form-group">
      <label for="association_name"><?php echo $this->lang->line('Name'); ?><font color="red"> *</font></label>
      <input type="text" class="form-control" id="association_name" name="association_name" autocomplete="off">
    </div>

     <div class="form-group">
      <label for="association_remark"><?php echo $this->lang->line('Remark'); ?><font color="red"> *</font></label>
      <textarea type="text" class="form-control" id="association_remark" name="association_remark" autocomplete="off"></textarea>
    </div>

  </div>
```

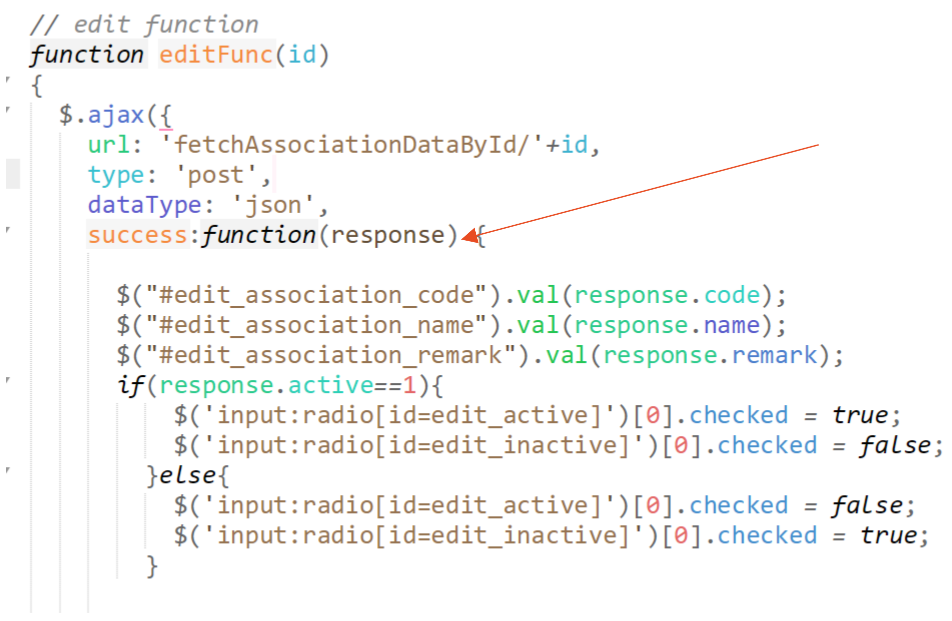
No change in the query javascript part of the create of association

In the edit part:



```
    <div class="form-group">
      <label for="edit_association_name"><?php echo $this->lang->line('Name'); ?><font color="red"> *</font></label>
      <input type="text" class="form-control" id="edit_association_name" name="edit_association_name" autocomplete="off">
    </div>

    <div class="form-group">
      <label for="edit_association_remark"><?php echo $this->lang->line('Remark'); ?><font color="red"> *</font></label>
      <textarea type="text" class="form-control" id="edit_association_remark" name="edit_association_remark" autocomplete="off"></textarea>
    </div>
  </div>
```

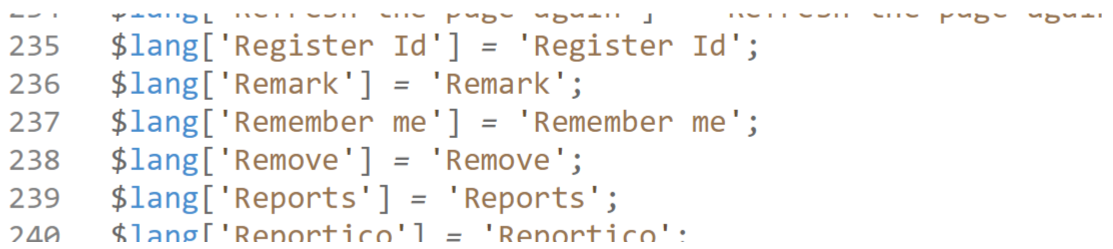
In the query javascript part, the field remark must be filled from the database:



```
// edit function
function editFunc(id)
{
  $.ajax({
    url: 'fetchAssociationDataById/'+id,
    type: 'post',
    dataType: 'json',
    success:function(response){

      $("#edit_association_code").val(response.code);
      $("#edit_association_name").val(response.name);
      $("#edit_association_remark").val(response.remark);
      if(response.active==1){
          $('input:radio[id=edit_active]')[0].checked = true;
          $('input:radio[id=edit_inactive]')[0].checked = false;
        }else{
          $('input:radio[id=edit_active]')[0].checked = false;
          $('input:radio[id=edit_inactive]')[0].checked = true;
        }
```

Verify the language translation

Verify if the title *Remark* is in the translation part of the system. Application/language/english/english_lang.php and application/language/french/french_lang.php



```
$lang['Register Id'] = 'Register Id';
$lang['Remark'] = 'Remark';
$lang['Remember me'] = 'Remember me';
$lang['Remove'] = 'Remove';
$lang['Reports'] = 'Reports';
$lang['Reportico'] = 'Reportico';
```

Modify the report

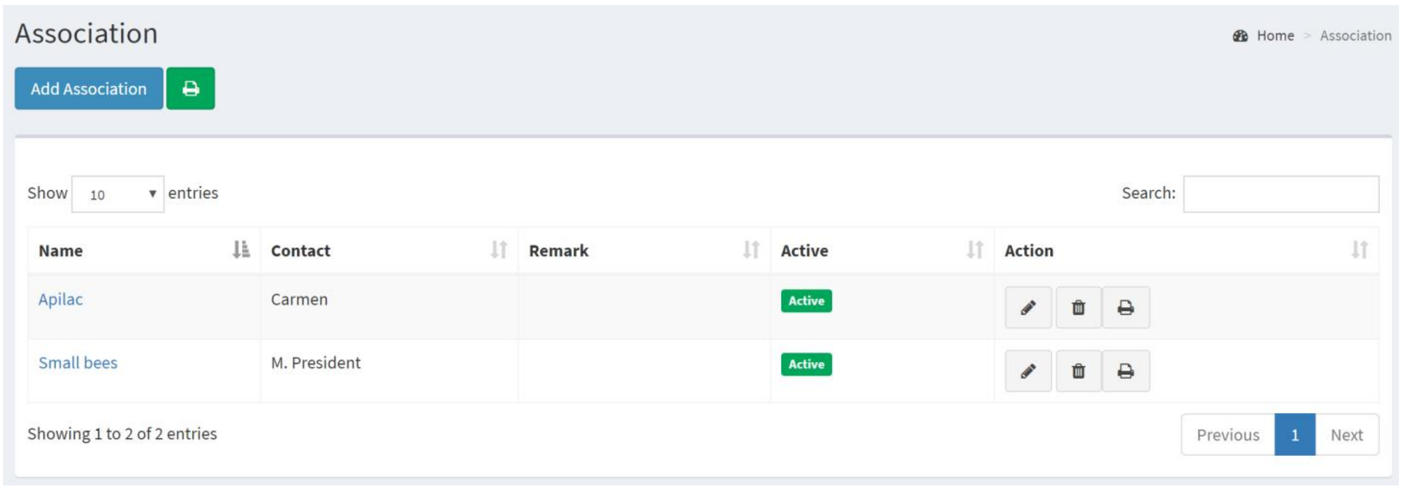
If a report for the association exists, verify if the field Remark should be included. NO REPORT

Modify the user guide

User guide: modify the capture of the screen on association in the user guide. Save the user guide in pdf format assets/documentation/user_guide_en.doc and user_guide_fr.doc. In this case, there is no screen in the user guide specifically for the association.

Testing

You should see the remark on the list of Association in Settings part.



Association

Home > Association

Add Association

Show 10 entries Search:

Name	Contact	Remark	Active	Action
Apilac	Carmen		Active	
Small bees	M. President		Active	

Showing 1 to 2 of 2 entries

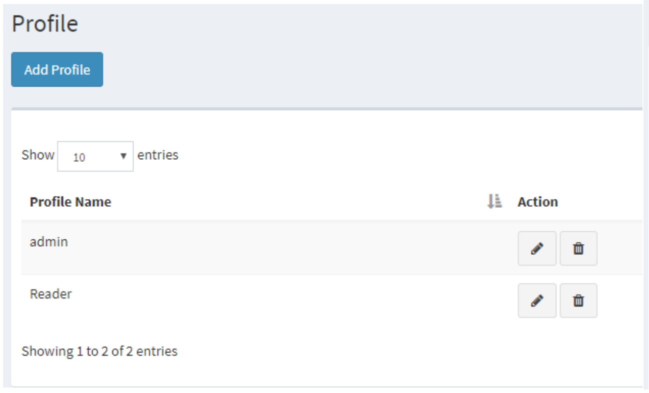
Previous 1 Next

And test the Add Association and Edit Association. Verify as well if the delete is still working good.

Security in the system

The security of the system is part of the profile. The Super-administrator can decide what will be the access of the profile, giving after to each user the appropriate profile.

Profile



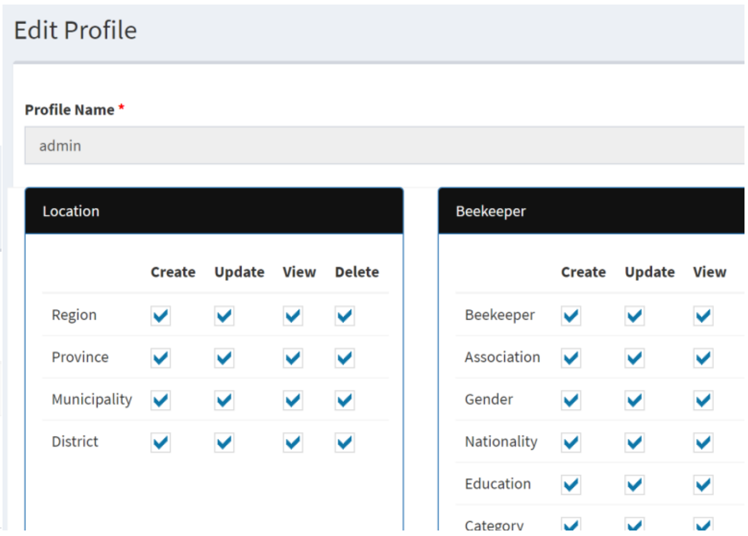
Profile

Add Profile

Show 10 entries

Profile Name	Action
admin	
Reader	

Showing 1 to 2 of 2 entries



Edit Profile

Profile Name *

admin

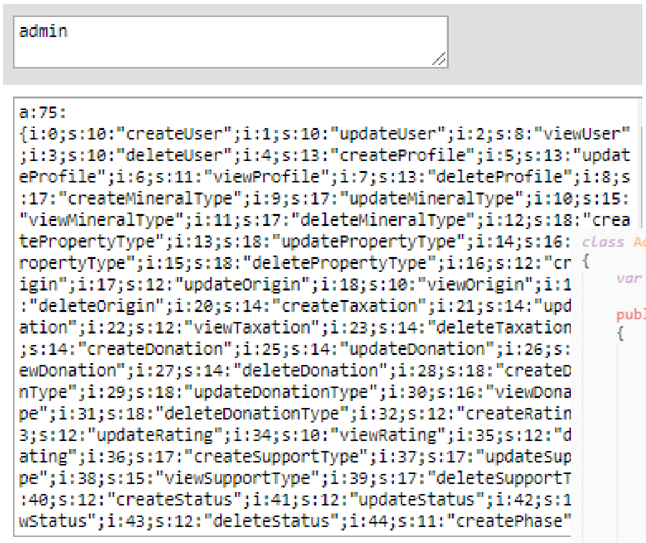
Location

	Create	Update	View	Delete
Region	✓	✓	✓	✓
Province	✓	✓	✓	✓
Municipality	✓	✓	✓	✓
District	✓	✓	✓	✓

Beekeeper

	Create	Update	View
Beekeeper	✓	✓	✓
Association	✓	✓	✓
Gender	✓	✓	✓
Nationality	✓	✓	✓
Education	✓	✓	✓
Category	✓	✓	✓

The profile admin have all the access. Only the administrator of the system should have all these access. Once the profile is created, it’s in the table profile.

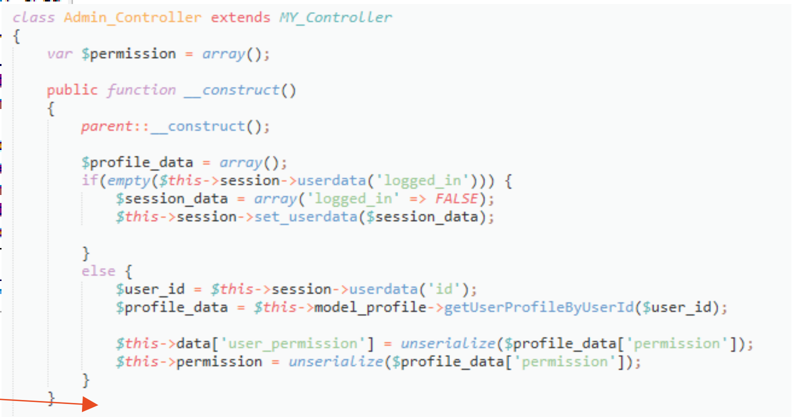


```
admin
```

```
a:75:
{i:0;s:10:"createUser";i:1;s:10:"updateUser";i:2;s:8:"viewUser"
;i:3;s:10:"deleteUser";i:4;s:13:"createProfile";i:5;s:13:"updat
eProfile";i:6;s:11:"viewProfile";i:7;s:13:"deleteProfile";i:8;s
:17:"createMineralType";i:9;s:17:"updateMineralType";i:10;s:15:
"viewMineralType";i:11;s:17:"deleteMineralType";i:12;s:18:"crea
tePropertyType";i:13;s:18:"updatePropertyType";i:14;s:16:
ropertyType";i:15;s:18:"deletePropertyType";i:16;s:12:"cr
igin";i:17;s:12:"updateOrigin";i:18;s:10:"viewOrigin";i:1
:"deleteOrigin";i:20;s:14:"createTaxation";i:21;s:14:"upd
ation";i:22;s:12:"viewTaxation";i:23;s:14:"deleteTaxation
;s:14:"createDonation";i:25;s:14:"updateDonation";i:26;s:
ewDonation";i:27;s:14:"deleteDonation";i:28;s:18:"createD
nType";i:29;s:18:"updateDonationType";i:30;s:16:"viewDona
pe";i:31;s:18:"deleteDonationType";i:32;s:12:"createRatin
3;s:12:"updateRating";i:34;s:10:"viewRating";i:35;s:12:"d
ating";i:36;s:17:"createSupportType";i:37;s:17:"updateSup
pe";i:38;s:15:"viewSupportType";i:39;s:17:"deleteSupportT
:40;s:12:"createStatus";i:41;s:12:"updateStatus";i:42;s:1
wStatus";i:43;s:12:"deleteStatus";i:44;s:11:"createPhase"
```

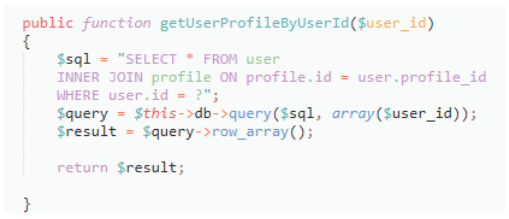
You can see in the entry of admin profile, all the access. Each module of the system will verify the profile given to the user and find the appropriate access.

It's in **application/core/My_controller.php** that the permission will be copied in the class permission



```
class Admin_Controller extends MY_Controller
{
    var $permission = array();

    public function __construct()
    {
        parent::__construct();

        $profile_data = array();
        if(empty($this->session->userdata('logged_in'))) {
            $session_data = array('logged_in' => FALSE);
            $this->session->set_userdata($session_data);

        }
        else {
            $user_id = $this->session->userdata('id');
            $profile_data = $this->model_profile->getUserProfileByUserId($user_id);

            $this->data['user_permission'] = unserialize($profile_data['permission']);
            $this->permission = unserialize($profile_data['permission']);
        }
    }
```

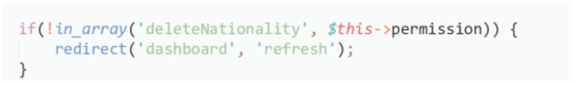
model_profile.php



```
public function getUserProfileByUserId($user_id)
{
    $sql = "SELECT * FROM user
    INNER JOIN profile ON profile.id = user.profile_id
    WHERE user.id = ?";
    $query = $this->db->query($sql, array($user_id));
    $result = $query->row_array();

    return $result;

}
```

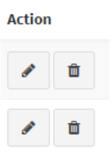
In all the module, before the treatment, there will always be a verification of the permission:

In this example, if the array deleteNationality is not found in $this->permission the delete will not be available and the user will be redirected to dashboard.

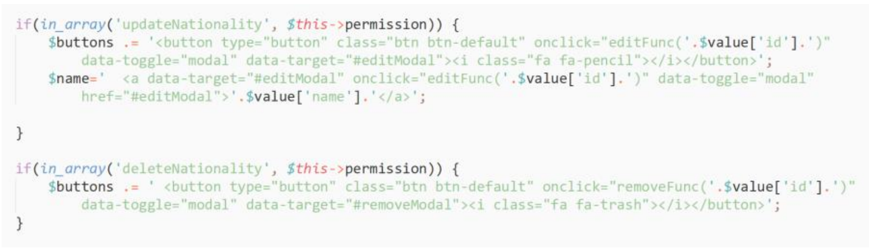


```
if(!in_array('deleteNationality', $this->permission)) {
    redirect('dashboard', 'refresh');
}
```

This is an example of the composition of the list of nationality. The icones edit and delete will not be visible if it's not in the array profile of the user.



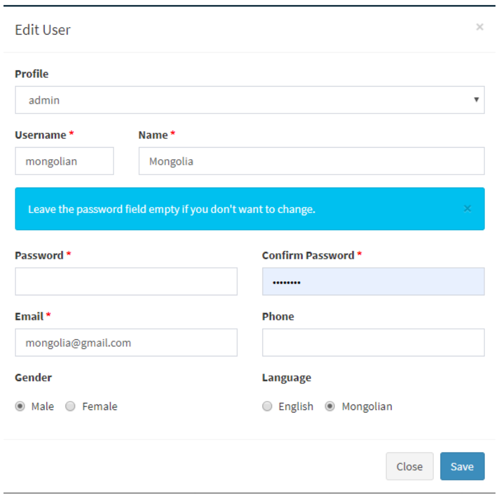
Action



```
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>';
}
```

User

It's the profile given to the user that will give the permission.



Edit User

Profile	admin
Username *	mongolian
Name *	Mongolia

Leave the password field empty if you don't want to change.

Password *	
Confirm Password *	••••••••
Email *	mongolia@gmail.com
Phone	
Gender	[✓ Male] Female
Language	English [✓ Mongolian]

Session variables

Session variables are special variables that exist only while the user's session with your application is active. Session variables are specific to each visitor to your site. They are used to store user-specific information that needs to be accessed by multiple pages in a web application. Like for example, the language used by the user or even the user code. Codeigniter have a library for the session variable. There is 2 possibilities.

Flashdata

The variable is temporary and will last only for one process. In this case, no need to unset the variable. The error message use this method.

```
$msg_error = $this->lang->line('Error occurred');
$this->session->set_flashdata('errors', $msg_error);
```

Userdata

The variable is available all the time, unless you unset the session variable

First, you have to create the session variables. In this example, I want to keep the beekeeper id for future treatment.

- Clear the content of the session variable beekeeper_id in case it already exists
- If it's empty, then we can fill
- We fill by the creation of an array with the name of the session variable (beekeeper_id) and the content, in this case $beekeeper_data['id']
- The session variable beekeeper_id is then filled with the name and content.

```
<!-- Creation of a session field to keep the beekeeper  -->
        <?php $this->session->unset_userdata('beekeeper_id');?>
        <?php if(empty($this->session->userdata('beekeeper_id'))) {
                $beekeeper_id = array('beekeeper_id' => $beekeeper_data['id']);
                $this->session->set_userdata($beekeeper_id);} ?>
```

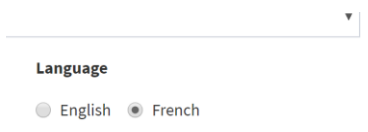
To access the session variable:

```
$beekeeper_id = $this->session->beekeeper_id;
```

Translation

Language and user

With the creation of users, you will indicate the language to be used in the system. The system can be available in different language. In this example, it's English and French.


Language

English [✓ French]

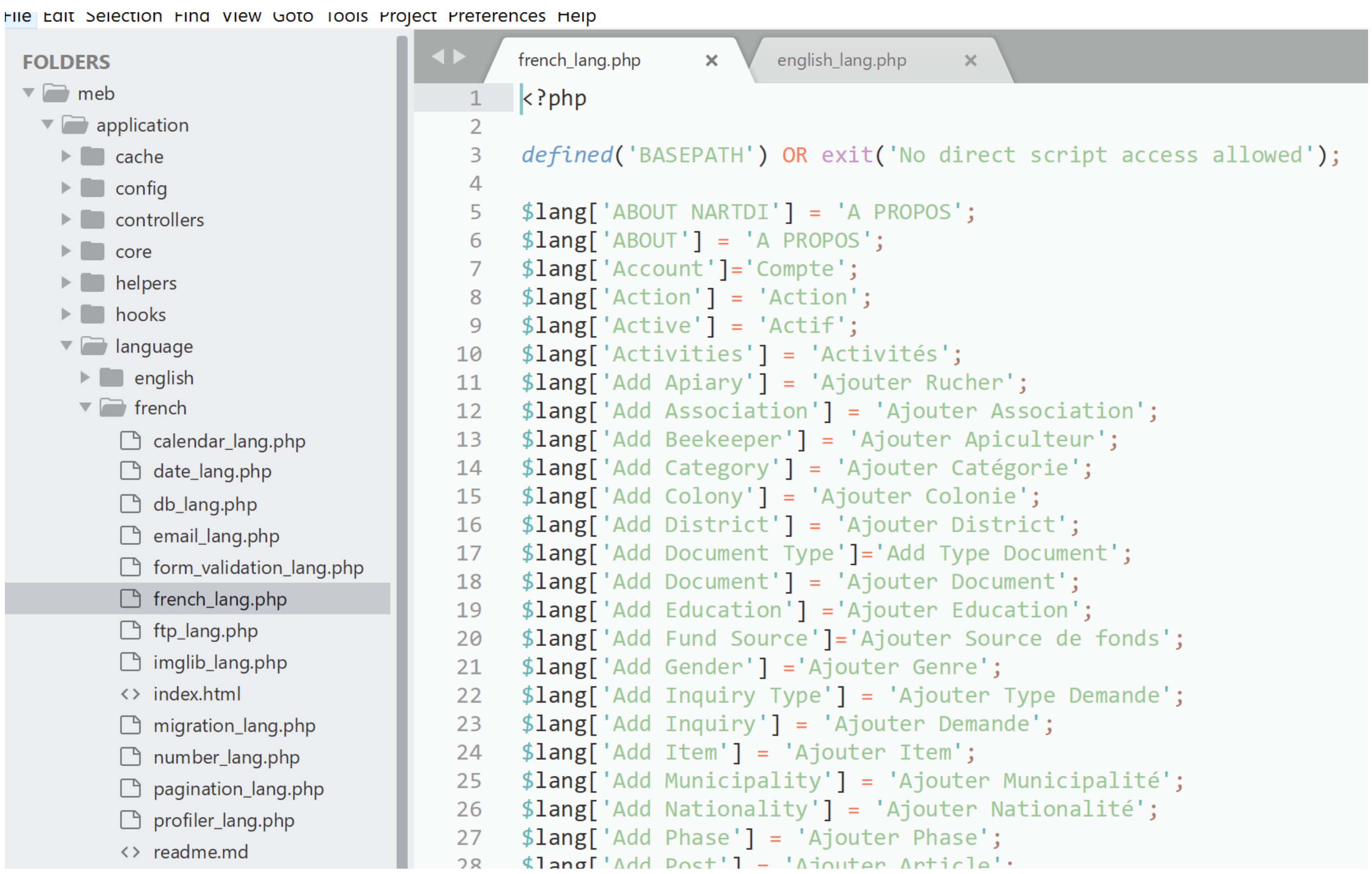
Language class

The language class is used for the translation of each titles in the system and for the messages from CodeIgniter.

In your CodeIgniter system folder, you will find a language sub-directory containing a set of language files for the English idiom. The files in this directory (system/language/english/) define the regular messages, error messages, and other generally output terms or expressions, for the different parts of the CodeIgniter framework.

You can create or incorporate your own language files, as needed, in order to provide application-specific error and other messages, or to provide translations of the core messages into other languages. These translations or additional messages would go inside your application/language/ directory, with separate sub-directories for each idiom (for instance 'french).

When CodeIgniter loads language files, it will load the one in system/language/ first and will then look for an override in your application/language/ directory.

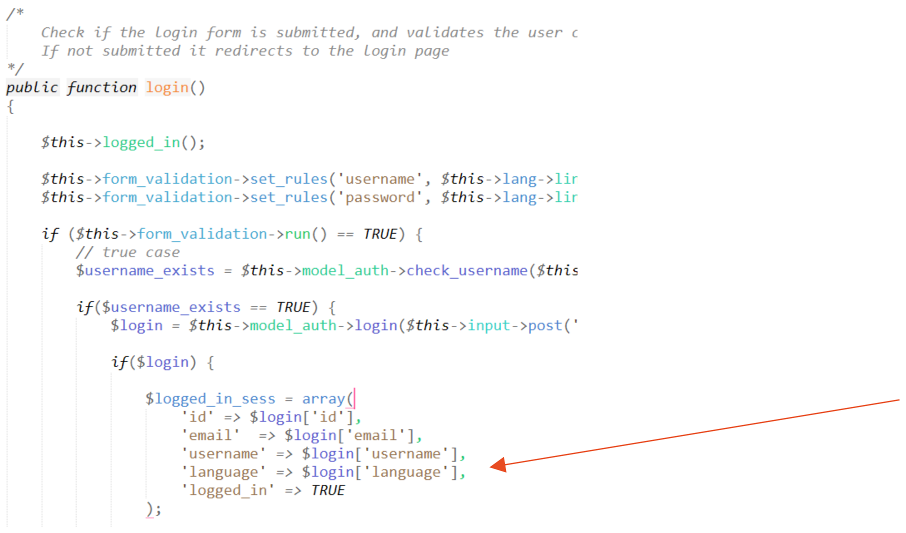


```php
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

$lang['ABOUT NARTDI'] = 'A PROPOS';
$lang['ABOUT'] = 'A PROPOS';
$lang['Account']='Compte';
$lang['Action'] = 'Action';
$lang['Active'] = 'Actif';
$lang['Activities'] = 'Activités';
$lang['Add Apiary'] = 'Ajouter Rucher';
$lang['Add Association'] = 'Ajouter Association';
$lang['Add Beekeeper'] = 'Ajouter Apiculteur';
$lang['Add Category'] = 'Ajouter Catégorie';
$lang['Add Colony'] = 'Ajouter Colonie';
$lang['Add District'] = 'Ajouter District';
$lang['Add Document Type']='Add Type Document';
$lang['Add Document'] = 'Ajouter Document';
$lang['Add Education'] ='Ajouter Education';
$lang['Add Fund Source']='Ajouter Source de fonds';
$lang['Add Gender'] ='Ajouter Genre';
$lang['Add Inquiry Type'] = 'Ajouter Type Demande';
$lang['Add Inquiry'] = 'Ajouter Demande';
$lang['Add Item'] = 'Ajouter Item';
$lang['Add Municipality'] = 'Ajouter Municipalité';
$lang['Add Nationality'] = 'Ajouter Nationalité';
$lang['Add Phase'] = 'Ajouter Phase';
```

In the English part of the language, you will find all the titles and messages that are used in the forms. You need to have the customize title in both languages: English and French. The language class will call the line and replace the field with the appropriate translation depending on the user preference.

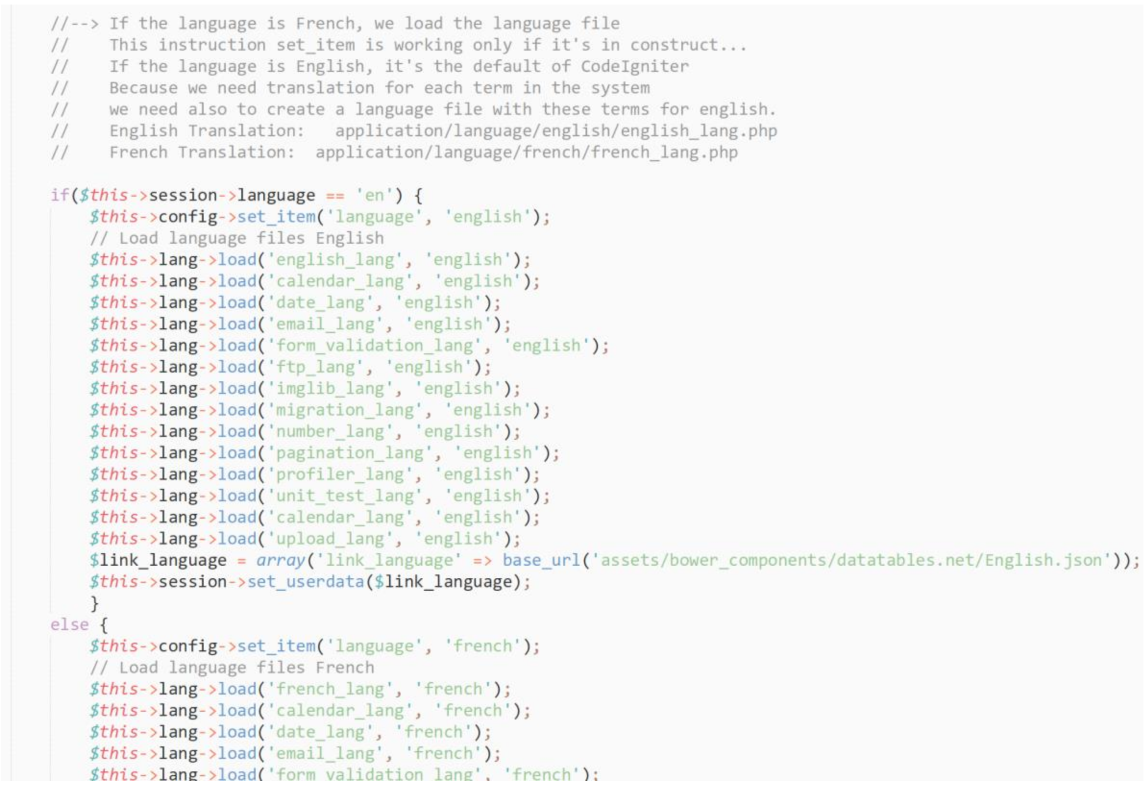
Sequence for translation

First, the user is identified and the language preference is kept in the session variable language. The form **/application/controller/Auth.php** will create the session variable language.



```
/*
    Check if the login form is submitted, and validates the user c
    If not submitted it redirects to the login page
*/
public function login()
{

    $this->logged_in();

    $this->form_validation->set_rules('username', $this->lang->lir
    $this->form_validation->set_rules('password', $this->lang->lir

    if ($this->form_validation->run() == TRUE) {
        // true case
        $username_exists = $this->model_auth->check_username($this

        if($username_exists == TRUE) {
            $login = $this->model_auth->login($this->input->post('

            if($login) {

                $logged_in_sess = array(
                    'id' => $login['id'],
                    'email'  => $login['email'],
                    'username' => $login['username'],
                    'language' => $login['language'],
                    'logged_in' => TRUE
                );
```

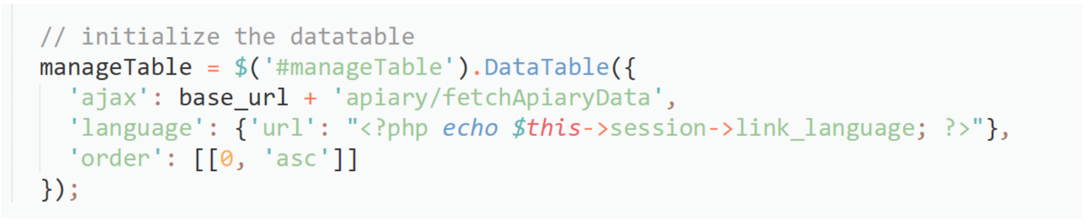
After **/application/core/My_controller.php** will load the module containing the translation in Mongolian.



```
//--> If the language is French, we load the language file
//    This instruction set_item is working only if it's in construct...
//    If the language is English, it's the default of CodeIgniter
//    Because we need translation for each term in the system
//    we need also to create a language file with these terms for english.
//    English Translation:   application/language/english/english_lang.php
//    French Translation:  application/language/french/french_lang.php

if($this->session->language == 'en') {
    $this->config->set_item('language', 'english');
    // Load language files English
    $this->lang->load('english_lang', 'english');
    $this->lang->load('calendar_lang', 'english');
    $this->lang->load('date_lang', 'english');
    $this->lang->load('email_lang', 'english');
    $this->lang->load('form_validation_lang', 'english');
    $this->lang->load('ftp_lang', 'english');
    $this->lang->load('imglib_lang', 'english');
    $this->lang->load('migration_lang', 'english');
    $this->lang->load('number_lang', 'english');
    $this->lang->load('pagination_lang', 'english');
    $this->lang->load('profiler_lang', 'english');
    $this->lang->load('unit_test_lang', 'english');
    $this->lang->load('calendar_lang', 'english');
    $this->lang->load('upload_lang', 'english');
    $link_language = array('link_language' => base_url('assets/bower_components/datatables.net/English.json'));
    $this->session->set_userdata($link_language);
    }
else {
    $this->config->set_item('language', 'french');
    // Load language files French
    $this->lang->load('french_lang', 'french');
    $this->lang->load('calendar_lang', 'french');
    $this->lang->load('date_lang', 'french');
    $this->lang->load('email_lang', 'french');
    $this->lang->load('form_validation_lang', 'french');
```

The dataTable used in the system for listing the information is not a CodeIgniter facility, so it's not translated in the language directory. It's a jQuery facility and the translation have been copied in **/assets/bower_components/datatables.net/English.json** or **French.json** depending on the language.

In Javascript, the parameter 'Language' will indicate which language will be used in the list tables.



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

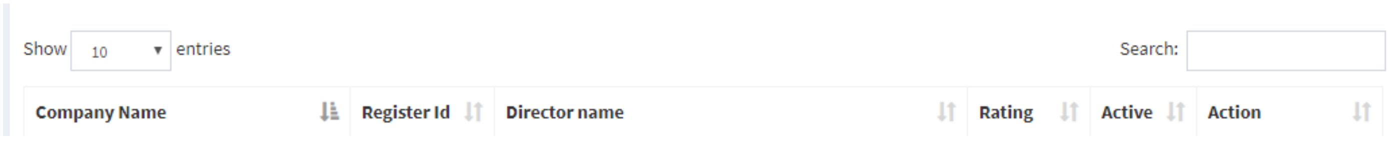
Jquery modal and Javascript

Order the table with javascript

For getting a different default order in the table with Javascript, you can use the parameter order. The first parameter will be the order of the field presented. In this case 0 is the name of the beekeeper, asc for ascending.

```
        manageTableDocument = $('#manageTableDocument').DataTable({
                'ajax': base_url+'beekeeper/fetchBeekeeperDocument/'+<?php echo $beekeeper_data['id']; ?>,
                'order': [[0, "asc"]]
```

The user will still be able to organize the order of the list differently by pressing the titles of the field.



Show 10 entries Search:

Company Name	Register Id	Director name	Rating	Active	Action

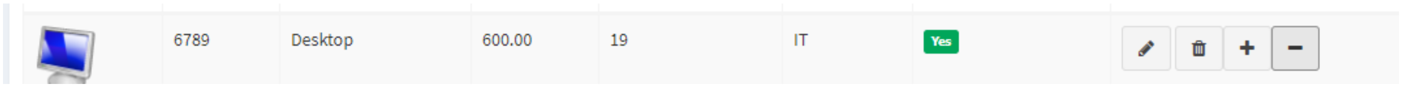
Passing parameter

We can pass many parameters to the modal. You just need to divide with a ,

See the example in Stock system. Index of item where I needed to indicate if the activity for the inventory would be + or – (in or out)

First in the controller where we generate the line for each item (item_id and type of activity):

```
if(in_array('updateItem', $this->permission)) {
            $buttons .= '<button type="button" class="btn btn-default" onclick="activity('.$value['id'].',2)" data-toggle="modal" data-target="#activityModal"><i class="fa fa-minus" title="O U T  of Inventory"></i></button>';       }
```



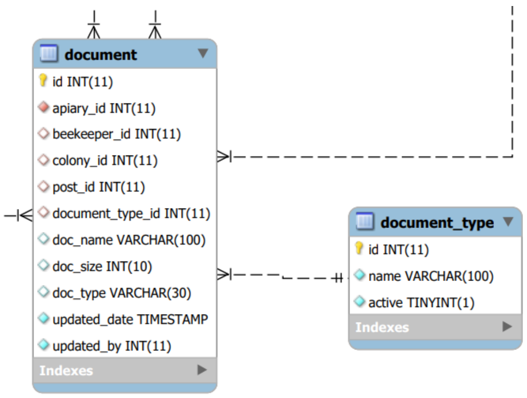
	6789	Desktop	600.00	19	IT	Yes	

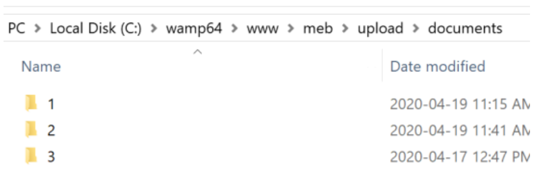
And in the index – modal, we just read the parametr in the function:

```
// Activity inventory
function activity(item_id,type_activity)
{
```

Upload of documents

You have the possibility to upload documents in Beekeeper, Apiary and Colony. You will find the method in the controller Beekeeper.php, the model model_beekeeper.php and in the edit.php of each beekeeper/apiary/colony. For the identification of documents, a document type table is in relation with the document table. Each document will have an entry in the table document but will also be uploaded in the appropriate directory created using the beekeeper id. **upload/documents/name_of_directory**




PC › Local Disk (C:) › wamp64 › www › meb › upload › documents

Name	Date modified
1	2020-04-19 11:15 AM
2	2020-04-19 11:41 AM
3	2020-04-17 12:47 PM

We use codeigniter class **Directory** and the library **upload**.

Creation of the directory of beekeeper

The directory is created when we add the beekeeper. We will take the beekeeper.id to identify the directory in the table beekeeper. All the documents uploaded from the beekeeper, apiary or colony forms will be in this repertory.

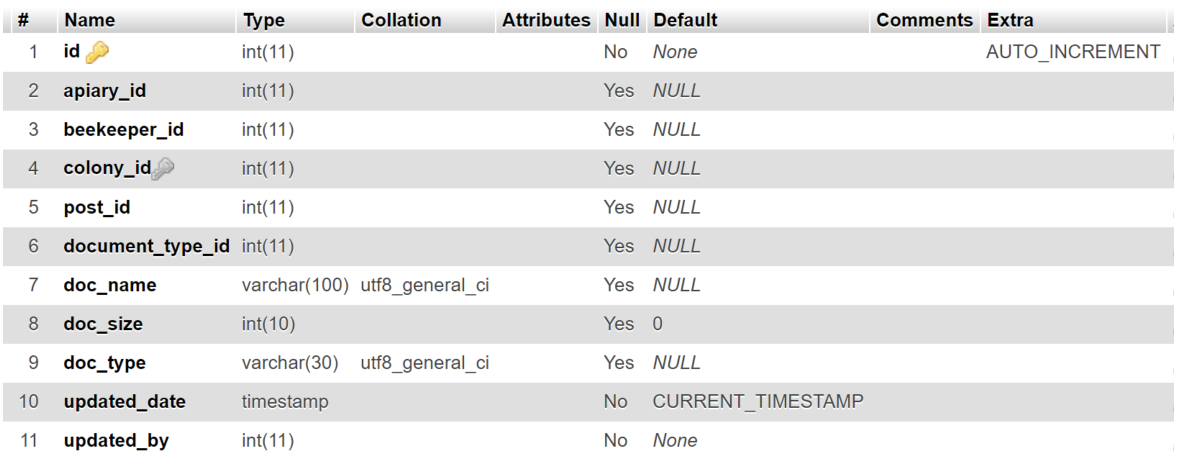
'directory' => $this->input->post('beekeeper.id'),

After the insert in the table, we proceed to the creation of the directory in upload/documents/

```
$beekeeper_id = $create;
//---> Create the directory for deposit of documents-->
$path = "./upload/documents/".$beekeeper_id;
$data = array(
        'directory' => $beekeeper_id,
```

Tables document and document type

2 tables are used for the management of documents. At the same moment we upload the document in the directory, an entry is created in table document. You can see that the name of the document will be renamed if there is some spaces in the name. Spaces are hard to manage when we work with upload and directory. The beekeeper_id will bring to the directory.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	int(11)			No	*None*		AUTO_INCREMENT
2	**apiary_id**	int(11)			Yes	*NULL*		
3	**beekeeper_id**	int(11)			Yes	*NULL*		
4	**colony_id**	int(11)			Yes	*NULL*		
5	**post_id**	int(11)			Yes	*NULL*		
6	**document_type_id**	int(11)			Yes	*NULL*		
7	**doc_name**	varchar(100)	utf8_general_ci		Yes	*NULL*		
8	**doc_size**	int(10)			Yes	0		
9	**doc_type**	varchar(30)	utf8_general_ci		Yes	*NULL*		
10	**updated_date**	timestamp			No	CURRENT_TIMESTAMP		
11	**updated_by**	int(11)			No	*None*		

Upload of documents

The functions in Beekeeper, Apiary and Colony are:

- fetchBeekeeperDocument generate the list of documents.
- UploadDocument for uploading the documents
- removeDocument for delete of documents

The directory will be indicated in a session variable in edit.php

```
<!-- Creation of a session to keep the directory for the manipulation
of upload of documents -->

<?php $this->session->unset_userdata('directory');?>
<?php if(empty($this->session->userdata('directory'))) {
    $directory = array('directory' => '/upload/documents/'.$beekeeper_data['directory'].'/');
    $this->session->set_userdata($directory);
    } ?>
```

Upload will first config the path, the types and maximum size for the uploading of documents. This is part of the library upload. Once all the information is completed in the configuration of the library upload, we proceed to the creation of the entry in the table document.

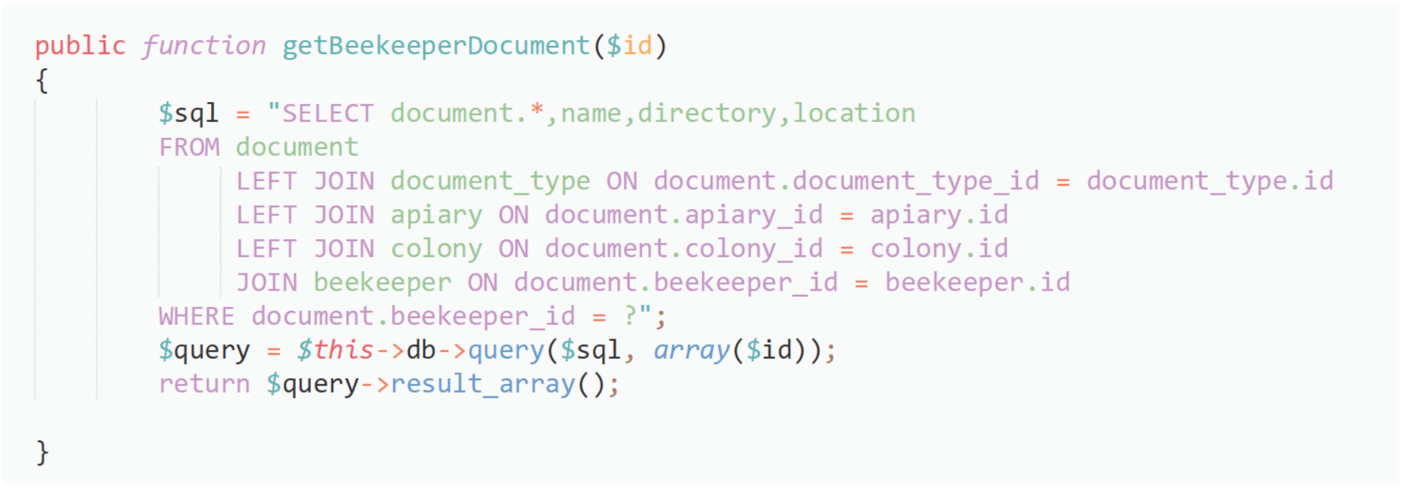
```
$directory = $this->session->directory;
$config['upload_path'] = './'.$directory;
$config['allowed_types'] = 'gif|jpg|png|pdf|xls|xlsx|docx|doc|pptx';
$config['max_size'] = '4000';

$this->load->library('upload', $config);

if ( ! $this->upload->do_upload('beekeeper_document')) {
    $msg_error = $this->lang->line('This type of document is not allowed or the document is too large.');
    $this->session->set_flashdata('warning', $msg_error);
    redirect('beekeeper/update/'.$this->session->beekeeper_id."?tab=document", 'refresh');
    }
else
    {
    //---> Create the document in the table document

    $doc_link = $directory.$this->upload->data('file_name');

    $data = array(
        'beekeeper_id' => $this->session->beekeeper_id,
        'doc_size' => $this->upload->data('file_size'),
        'doc_type' => $this->upload->data('file_type'),
        'doc_name' => $this->upload->data('file_name'),
        'document_type_id' => $this->input->post('document_type'),
        'updated_by' => $this->session->user_id,
    );

    $create = $this->model_beekeeper->createDocument($data);
```

When the document is successfully created in the table document, we can upload the document in the directory.

```
if($create == true) {
    //--->  Upload the document
    $data = array('upload_data' => $this->upload->data());
    redirect('beekeeper/update/'.$this->session->beekeeper_id."?tab=document", 'refresh');
```

View of documents

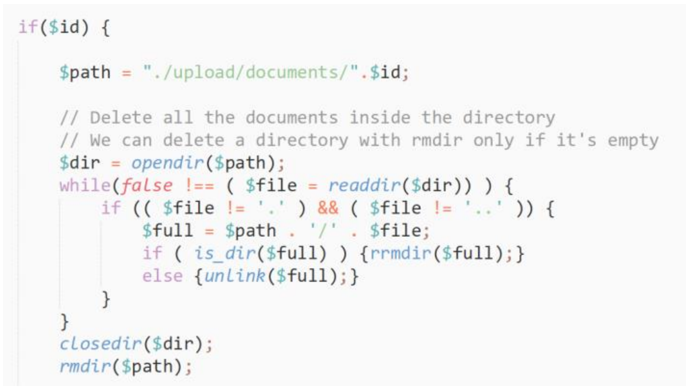
The view of documents is generated in the function fetchBeekeeperDocument who call the model_beekeeper to execute the function getBeekeeperDocument.



```
public function getBeekeeperDocument($id)
{
        $sql = "SELECT document.*,name,directory,location
        FROM document
             LEFT JOIN document_type ON document.document_type_id = document_type.id
             LEFT JOIN apiary ON document.apiary_id = apiary.id
             LEFT JOIN colony ON document.colony_id = colony.id
             JOIN beekeeper ON document.beekeeper_id = beekeeper.id
        WHERE document.beekeeper_id = ?";
        $query = $this->db->query($sql, array($id));
        return $query->result_array();

}
```

Delete a document

When we delete a document we must take care of deleting the entry in the table document.

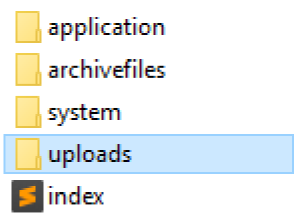


```
if($id) {

    $path = "./upload/documents/".$id;

    // Delete all the documents inside the directory
    // We can delete a directory with rmdir only if it's empty
    $dir = opendir($path);
    while(false !== ( $file = readdir($dir)) ) {
        if (( $file != '.' ) && ( $file != '..' )) {
            $full = $path . '/' . $file;
            if ( is_dir($full) ) {rrmdir($full);}
            else {unlink($full);}
        }
    }
    closedir($dir);
    rmdir($path);
```

Download and Zip file

1. Store files

Created a uploads directory at project root.



Store files in uploads directory and created another sub-directory documents for storing files.

I am using uploads directory in zip file creation.

Create archivefiles directory at project root to save the created zip files.

2. Configuration

Default controller   Open application/config/routes.php and edit default_controller value to Zip.

$route['default_controller'] = 'Zip';

3. Controller

Create a Zip.php file in application/controllers/ directory.

Create 3 methods –

- __construct – Load url helper and zip library.
- index – Load index_view view.
- createzip – This method is called on <form > submit.

Add files –

On the first button click, I am adding specified files for the compress.

For this, I have assign file path in $filepath1 and $filepath2.

To add file use $this->zip->read_file();. This method takes file-path as a parameter.

$this->zip->read_file([file-path]);

Pass the $filepath1 and $filepath2 in the method.

For downloading file on the user system call $this->zip->download() method. This method takes file-name as a parameter.

$this->zip->download([file-name]);

Add directory files and sub-directory –

On the second button click, I am adding whole directory files and sub-directory for compress.

Specify a directory path in the $path.

Use $this->zip->read_dir() to add directory files.

$this->zip->read_dir([directory-path]);

To save a zip file on a server call $this->zip->archive() method and pass the path with file-name where you want to store.

$this->zip->archive([file-path]);

Execute $this->zip->download() method for downloading the file to the user system.

Completed Code

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Zip extends CI_Controller {

 public function __construct(){

   parent::__construct();
   $this->load->helper('url');

   // Load zip library
   $this->load->library('zip');

 }

 public function index(){
   // Load view
   $this->load->view('index_view');
 }

 // Create zip
 public function createzip(){

   // Read file from path
   if($this->input->post('but_createzip1') != NULL){

     // File path
     $filepath1 = FCPATH.'/uploads/image1.jpg';
     $filepath2 = FCPATH.'/uploads/document/users.csv';

     // Add file
     $this->zip->read_file($filepath1);
     $this->zip->read_file($filepath2);

     // Download
     $filename = "backup.zip";
     $this->zip->download($filename);

   }

   // Read files from directory
   if($this->input->post('but_createzip2') != NULL){
     // File name
     $filename = "backup.zip";
     // Directory path (uploads directory stored in project root)
     $path = 'uploads';
```

```
      // Add directory to zip
      $this->zip->read_dir($path);

      // Save the zip file to archivefiles directory
      $this->zip->archive(FCPATH.'/archivefiles/'.$filename);

      // Download
      $this->zip->download($filename);
    }

    // Load view
    $this->load->view('index_view');
  }

}
```

4. View

Create a index_view.php file application/views/ directory.

Create a <form > and set action='<?= base_url() ?>index.php/zip/createzip‘.

Create two submit buttons.

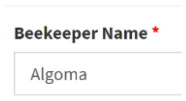
1. One for creating a zip file from the specified path.
2. and another for creating zip file from a directory.

Completed Code

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
?>
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Create and Download Zip file in CodeIgniter</title>
  </head>
  <body>

    <form method='post' action='<?= base_url() ?>index.php/posts/createzip/'>
      <input type="submit" name="but_createzip1" value='Add file from path and download zip'>
      <input type="submit" name="but_createzip2" value='Add directory files and sub-directory, save archive and
download zip'>
    </form>

  </body>
</html>
```

Validation

Validation is done using the codeigniter library **Form Validation**


Beekeeper Name *
Algoma

Let's take the example of Beekeeper. The form views/beekeeper/edit.php is where the user complete the information about the Beekeeper. The required field are indicated with a red asterisk in the system. It means that a validation will be done to make sure that the field is filled.

Library Form Validation

It's in the controller beekeeper.php that the validation will be handle in the create and update functions.

```php
$this->form_validation->set_rules('beekeeper_name', $this->lang->line('Beekeeper Name'), 'trim|required');
$this->form_validation->set_rules('category', $this->lang->line('Category'), 'trim|required');
$this->form_validation->set_rules('association', $this->lang->line('Association'), 'trim|required');
$this->form_validation->set_rules('nationality[]', $this->lang->line('Nationality'), 'trim|required');
$this->form_validation->set_rules('gender', $this->lang->line('Gender'), 'trim|required');
$this->form_validation->set_rules('address', $this->lang->line('Address'), 'trim|required');
$this->form_validation->set_rules('region', $this->lang->line('Region'), 'trim|required');
$this->form_validation->set_rules('province', $this->lang->line('Province'), 'trim|required');
$this->form_validation->set_rules('municipality', $this->lang->line('Municipality'), 'trim|required');
$this->form_validation->set_rules('district', $this->lang->line('District'), 'trim|required');
$this->form_validation->set_rules('birth_date', $this->lang->line('Birth Date'), 'trim|required|valid_date');
$this->form_validation->set_rules('education', $this->lang->line('Highest Educational Attainment'), 'trim|required');
$this->form_validation->set_rules('fund_source[]', $this->lang->line('Fund Source'), 'trim|required');
$this->form_validation->set_error_delimiters('<p class="alert alert-warning">','</p>');

if ($this->form_validation->run() == TRUE) {
    // True case, we create the new beekeeper
```

If the validation is completed, the update or create will be done. If not, the error message generated by the form_validation class will be sent to the view who will show the error message after the form have been submitted to the server.

```php
<?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; ?>
```

Creating our own validation with callback method

The validation system supports callbacks to your own validation methods. This allows you to extend the validation class to meet your needs. For example, if you need to run a specific function or a database query to validate some information, you can create a callback method that does that. Let's create an example of this for the validation of dates. We need to verify the range of dates (date from and date to) and the rules of the validation greater_than works only for numbers, not for date.

I used the rules of form_validation to indicate that it's required but for the date_end, I created a function to check the date.

```php
$this->form_validation->set_rules('date_issued', $this->lang->line('Date issued'), 'trim|required');

$this->form_validation->set_rules('date_end', $this->lang->line('Date end'), 'trim|required|callback_date_check');
```

```
public function date_check()
{
    if ($this->input->post('date_end') < $this->input->post('date_issued') )
        {$this->form_validation->set_message('date_check', $this->lang->line('The date End should be greater
            or equal to the date Issued.'));
        return FALSE;
        }
    else
        {return TRUE;}
}
```

The message should be translated in both language.

Check Integrity

More specific validation will be done for example, when deleting settings table that might be used in the system. In all the remove function of these tables, you will find this validation. In this example, the District can't be delete if the function checkIntegrity indicate that some rows are using the entry.

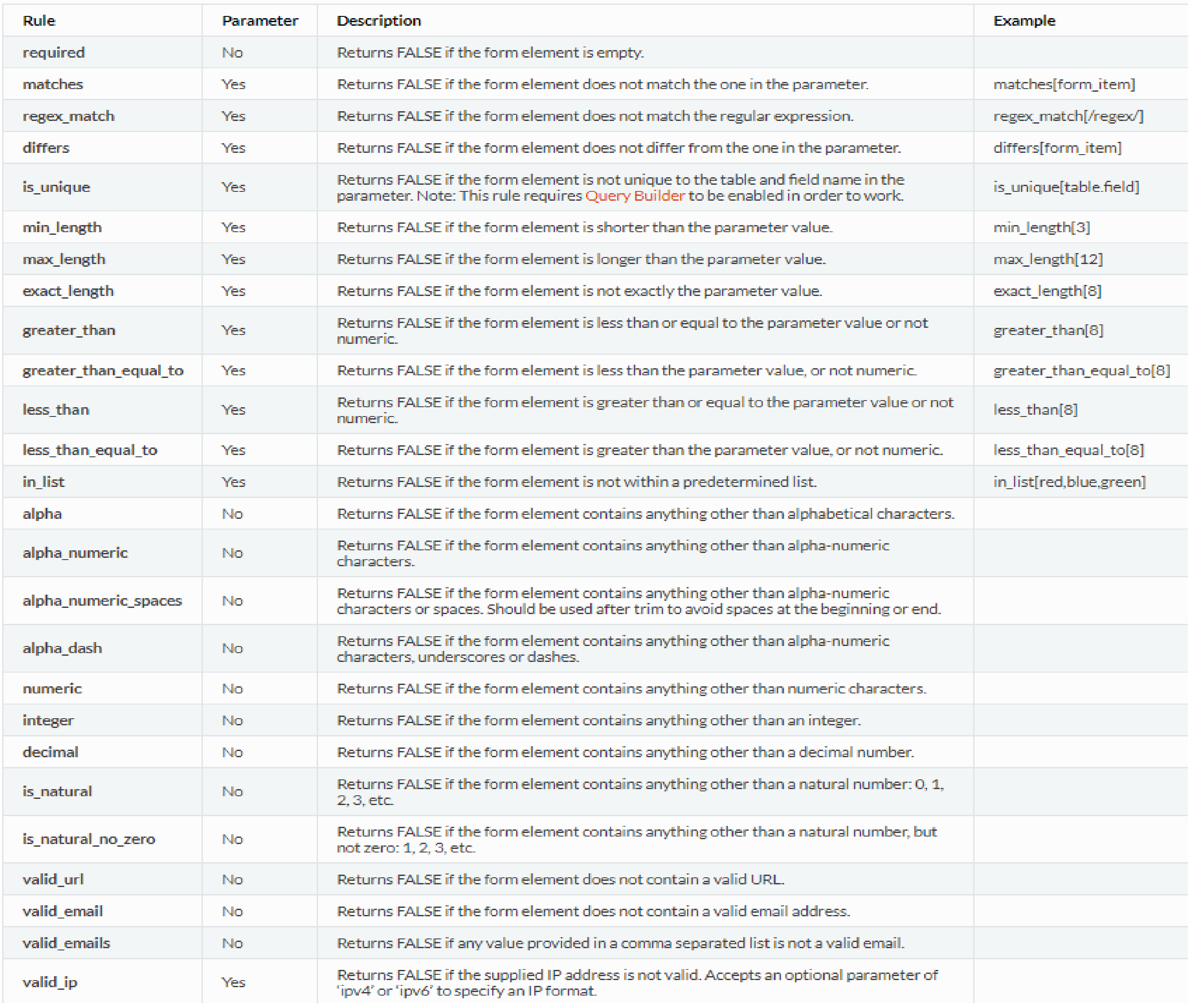
```
if($district_id) {
    //---> Validate if the information is used in beekeeper/apiary/association table
    $total_rows = $this->model_district->checkIntegrity($district_id);
    //---> If no beekeeper/apiary/association have this information, we can delete
    if ($total_rows == 0) {
        $delete = $this->model_district->remove($district_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');}
        }
```

In this case, District can be found in more than one tables.

```
//---> Validate if the district is used in table Beekeeper, Association or Apiary
public function checkIntegrity($id)
{

    $num_rows = 0;

    $sql = "SELECT * FROM beekeeper WHERE district_id = ?";
    $query = $this->db->query($sql, array($id));
    $num_rows = $num_rows + $query->num_rows();

    $sql = "SELECT * FROM association WHERE district_id = ?";
    $query = $this->db->query($sql, array($id));
    $num_rows = $num_rows + $query->num_rows();

    $sql = "SELECT * FROM apiary WHERE district_id = ?";
    $query = $this->db->query($sql, array($id));
    $num_rows = $num_rows + $query->num_rows();

    return $num_rows;

}
```

Form validation parameter

Here you have the information about the rules of the library form_validation



Rule	Parameter	Description	Example
required	No	Returns FALSE if the form element is empty.	
matches	Yes	Returns FALSE if the form element does not match the one in the parameter.	matches[form_item]
regex_match	Yes	Returns FALSE if the form element does not match the regular expression.	regex_match[/regex/]
differs	Yes	Returns FALSE if the form element does not differ from the one in the parameter.	differs[form_item]
is_unique	Yes	Returns FALSE if the form element is not unique to the table and field name in the parameter. Note: This rule requires Query Builder to be enabled in order to work.	is_unique[table.field]
min_length	Yes	Returns FALSE if the form element is shorter than the parameter value.	min_length[3]
max_length	Yes	Returns FALSE if the form element is longer than the parameter value.	max_length[12]
exact_length	Yes	Returns FALSE if the form element is not exactly the parameter value.	exact_length[8]
greater_than	Yes	Returns FALSE if the form element is less than or equal to the parameter value or not numeric.	greater_than[8]
greater_than_equal_to	Yes	Returns FALSE if the form element is less than the parameter value, or not numeric.	greater_than_equal_to[8]
less_than	Yes	Returns FALSE if the form element is greater than or equal to the parameter value or not numeric.	less_than[8]
less_than_equal_to	Yes	Returns FALSE if the form element is greater than the parameter value, or not numeric.	less_than_equal_to[8]
in_list	Yes	Returns FALSE if the form element is not within a predetermined list.	in_list[red,blue,green]
alpha	No	Returns FALSE if the form element contains anything other than alphabetical characters.	
alpha_numeric	No	Returns FALSE if the form element contains anything other than alpha-numeric characters.	
alpha_numeric_spaces	No	Returns FALSE if the form element contains anything other than alpha-numeric characters or spaces. Should be used after trim to avoid spaces at the beginning or end.	
alpha_dash	No	Returns FALSE if the form element contains anything other than alpha-numeric characters, underscores or dashes.	
numeric	No	Returns FALSE if the form element contains anything other than numeric characters.	
integer	No	Returns FALSE if the form element contains anything other than an integer.	
decimal	No	Returns FALSE if the form element contains anything other than a decimal number.	
is_natural	No	Returns FALSE if the form element contains anything other than a natural number: 0, 1, 2, 3, etc.	
is_natural_no_zero	No	Returns FALSE if the form element contains anything other than a natural number, but not zero: 1, 2, 3, etc.	
valid_url	No	Returns FALSE if the form element does not contain a valid URL.	
valid_email	No	Returns FALSE if the form element does not contain a valid email address.	
valid_emails	No	Returns FALSE if any value provided in a comma separated list is not a valid email.	
valid_ip	Yes	Returns FALSE if the supplied IP address is not valid. Accepts an optional parameter of 'ipv4' or 'ipv6' to specify an IP format.	

Reports

TCPDF and CodeIgniter

Report are constructed using a combination of TCPDF and CodeIgniter. You can have information at www.tcpdf.org

The Report form handle the criteria's:

- Controller: Report.php main form and Report01.php Report02.php …. For each report
- Model: Model_report.php
- View: report/index.php

Each report have his own form in the Controller of CodeIgniter for a more simple maintenance. They are called directly in report/index.php using the html OBJECT data. This will allow the report to open inside CodeIgniter framework.

```
<!------------------------------------------------  P R I N T   R E P O R T S  ------------------------------------------ -->

    <?php if($this->session->printREP01 == 'yes') : ?>
      <object data="<?php echo  base_url("report01/REP01"); ?>" width="100%" height="1000px" type="application/pdf"> </object>
    <?php endif; ?>
```

The Controller Report.php will control the Report View and create the drop-down list. If a report have been asked, it create some temporary variable session for the parameters. These variables will be invoked in the Controller of each program.

Example: Report01.php

It's in the controller Report01.php that the report will be composed with TCPDF and the Table class of CodeIgniter. Depending on the complexity of the report, it might be composed only with TCPDF. A call to the model report will be done to get the data and fill the information in the column of the report.

Add a new criteria in a report

Example of adding category as a new criteria. To add a field in a report you must:

1. Enable the criteria in JQuery of report/index.php.

```
//--> Enable the parameters depending on the report chosen

switch($("#report :selected").val()) {
  case 'REP01':
        $("#province").prop( 'disabled', false );
        $("#municipality").prop( 'disabled', false );
        $("#category").prop( 'disabled', false );
        $("#generate").prop( 'disabled', false );
        break;
```

2. Create a temporary session variable for the criteria chosen in the controller Report.php

```
$this->session->set_flashdata('printdoc', 'no');

if($this->input->post('report') == 'REP01') {
    $this->session->set_flashdata('printREP01', 'yes');
    $this->session->set_flashdata('printdoc', 'yes');
    $this->session->set_flashdata('province', $this->input->post('province'));
    $this->session->set_flashdata('municipality', $this->input->post('municipality'));
    $this->session->set_flashdata('category', $this->input->post('category'));
}
```

3. Add the criteria in model_report.php

```
//--> Criteria Category
$category = $this->session->category;
if ($category == 'all') {
    $category_from = 1;
    $category_to = 999;
}
else {
    $category_from = $category;
    $category_to = $category;        }
```

Add a field in a report

1. To add a new field in a report, make sure that this field is available in the SELECT of the model_report.php
2. Add the field in the heading and treatment of the record set of the report and adjust the format of each column considering this new field.

```
$this->table->set_heading('<th width="10%" height="28"><strong>'.$this->lang->line('Register Id').'</strong></th>',
                          '<th width="25%" height="28"><strong>'.$this->lang->line('Beekeeper Name').'</strong></th>',
                          '<th width="19%" height="28"><strong>'.$this->lang->line('Address').'</strong></th>',
                          '<th width="10%" height="28"><strong>'.$this->lang->line('Province').'</strong></th>',
                          '<th width="13%" height="28"><strong>'.$this->lang->line('Municipality').'</strong></th>',
                          '<th width="10%" height="28"><strong>'.$this->lang->line('Association').'</strong></th>',
                          '<th width="13%" height="28"><strong>'.$this->lang->line('Category').'</strong></th>');

// Call to the database
$REP01 = $this->model_report->getREP01();

if ($REP01 == null) {
    // If not data found, we indicate in the report first line
    $this->table->add_row($this->lang->line('No data found'));
}
else {
    foreach ($REP01 as $rs):
        $cell1 = array('data' => $rs->beekeeper_register_id, 'width' => '10%');
        $cell2 = array('data' => $rs->beekeeper_name, 'width' => '25%');
        $cell3 = array('data' => $rs->address, 'width' => '19%');
        $cell4 = array('data' => $rs->province_name, 'width' => '10%');
        $cell5 = array('data' => $rs->municipality_name, 'width' => '13%');
        $cell6 = array('data' => $rs->association_name, 'width' => '10%');
        $cell7 = array('data' => $rs->category_name, 'width' => '13%');
        $this->table->add_row($cell1, $cell2, $cell3, $cell4, $cell5, $cell6, $cell7);
    endforeach;
}
```

Creation of new report in the system

The exercice is to create a report for Production on a step by step basic.

Step 1 - Report template needed

We need a list of all the production for an interval of dates. The report should be available with the criteria:

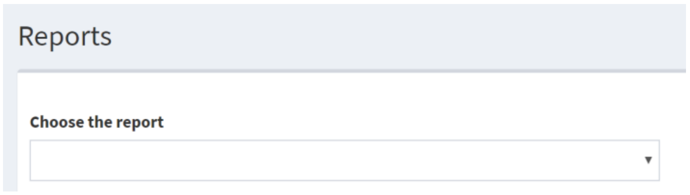
- Date From / Date To
- All Municipalities or one particular municipality

The information to print will be:

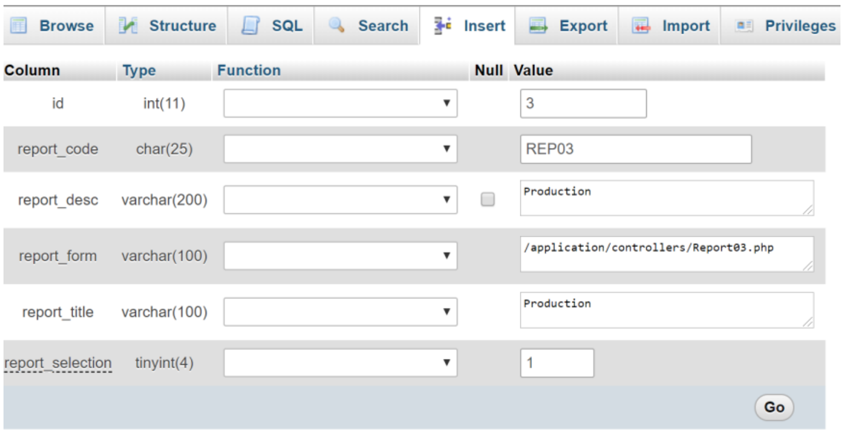
- Register ID
- Name of the beekeeper
- Product
- Total colony
- Total Production
- Production Date

Step 2 – Create the new report in the report list

This is the drop-down list of the report in the Report form.



Goto phpMyAdmin – Database name – table report



Column	Type	Function	Null	Value
id	int(11)			3
report_code	char(25)			REP03
report_desc	varchar(200)		☐	Production
report_form	varchar(100)			/application/controllers/Report03.php
report_title	varchar(100)			Production
report_selection	tinyint(4)			1

Step 3 – Integrate this new report in the controller Report.php and view /report/index.php

The new report should be integrated in the process of the controller/Report.php

This is where we send as a session variable (temporary flashdata) the information to the view/report/index.php

```
else if($this->input->post('report') == 'REP03') {
    $this->session->set_flashdata('printREP03', 'yes');
    $this->session->set_flashdata('printdoc', 'yes');
    $this->session->set_flashdata('municipality', $this->input->post('municipality'));
    $this->session->set_flashdata('date_from', $this->input->post('date_from'));
    $this->session->set_flashdata('date_to', $this->input->post('date_to'));
}
else if($this->input->post('report') == 'REP04') {
```

By default, all the criteria of the report form are disabled. It's when the user choose the report that we will enable the appropriate criteria. You must enable the parameters Municipality and Date-from Date-to for Report 03 in the javascript part of /views/report/index.php

```
case 'REP03':
        $("#date_from").prop( 'disabled', false );
        $("#date_to").prop( 'disabled', false );
        $("#municipality").prop( 'disabled', false );
        $("#generate").prop( 'disabled', false );
        break;
```

And call the report after the criteria have been filled by the user.

```
<!-------------------------------------------------- P R I N T   R E P O R T S ----------------------------------------- -->

    <?php if($this->session->printREP01 == 'yes') : ?>
        <object data="<?php echo  base_url("report01/REP01"); ?>" width="100%" height="1000px" type="application/pdf"> </object>
    <?php endif; ?>

    <?php if($this->session->printREP02 == 'yes') : ?>
        <object data="<?php echo  base_url("report02/REP02"); ?>" width="100%" height="1000px" type="application/pdf"> </object>
    <?php endif; ?>

    <?php if($this->session->printREP03 == 'yes') : ?>
        <object data="<?php echo  base_url("report03/REP03"); ?>" width="100%" height="1000px" type="application/pdf"> </object>
    <?php endif; ?>
```

Step 4 – Create the SQL query for report 03 in /models/model_report.php

```
public function getREP03()
{
    //--> Criteria Municipality
    $municipality = $this->session->municipality;
    if ($municipality == 'all') {
        $municipality_from = 0;
        $municipality_to = 999;
    }
    else {
        $municipality_from = $municipality;
        $municipality_to = $municipality;
    }

    $date_from = $this->session->date_from;
    $date_to = $this->session->date_to;
    if ($date_from == null) {$date_from = "'1900-01-01'";} else {$date_from = "'".$date_from."'";}
    if ($date_to == null) {$date_to = "'2500-01-01'";} else {$date_to = "'".$date_to."'";}

    $sql = "SELECT production.*,beekeeper_name,beekeeper_register_id,total_colony,
                   product.name AS 'product_name'
    FROM production
         JOIN colony ON production.colony_id = colony.id
         JOIN apiary ON colony.apiary_id = apiary.id
         JOIN beekeeper ON apiary.beekeeper_id = beekeeper.id
         JOIN product ON production.product_id = product.id
    WHERE apiary.municipality_id BETWEEN $municipality_from AND $municipality_to
        AND production_date BETWEEN $date_from AND $date_to
    ORDER BY beekeeper_name";

    $query = $this->db->query($sql, array());

    if ($query->num_rows() > 0) {return $query->result();}

    return NULL;

}
```

Step 5 – Create the report in /application/controllers/Report03.php

We will copy from a report that already exists like Report01.php The register ID and Beekeeper name is already there.

- Change everywhere Report01 for Report03
- Change everywhere REP01 for REP03
- Modify the size of the character because we have lots of information and we need small characters….
  - $pdf->SetFont('dejavusans', '', 8);
- Modify the title of the report: It will be Production taken from translation file. Make sure that the title is in the dictionnary of the language unless the title won't appear.   We add the year chosen by the user in the title.
  - $title_report = $this->lang->line('Production');
- Verify if Production exists in the translation english and mongolian
  - /application/language/english/english_lang.php
  - /application/language/english/mongolian_lang.php
- Change the heading for the appropriate title.

```
// Create a session variable to use the title in the header of tcpdf (library tcpdf / Pdf.php)
$this->session->set_flashdata('report_code', 'REP03');
```

- The result will be for the header:

```
$this->table->set_heading('<th width="15%" height="28"><strong>'.$this->lang->line('Register Id').'</strong></th>',
                          '<th width="20%" height="28"><strong>'.$this->lang->line('Beekeeper Name').'</strong></th>',
                          '<th width="10%" height="28"><strong>'.$this->lang->line('Product').'</strong></th>',
                          '<th width="15%" height="28" align="right"><strong>'.$this->lang->line('Total Colonies').'</strong></th>',
                          '<th width="20%" height="28" align="right"><strong>'.$this->lang->line('Total Production').'</strong></th>',
                          '<th width="20%" height="28" align="right"><strong>'.$this->lang->line('Production Date').'</strong></th>'
                          );
```

We want to align the header right when it's an amount. The percentage given of the total space taken on the report for each column must be evaluated while testing the printing.

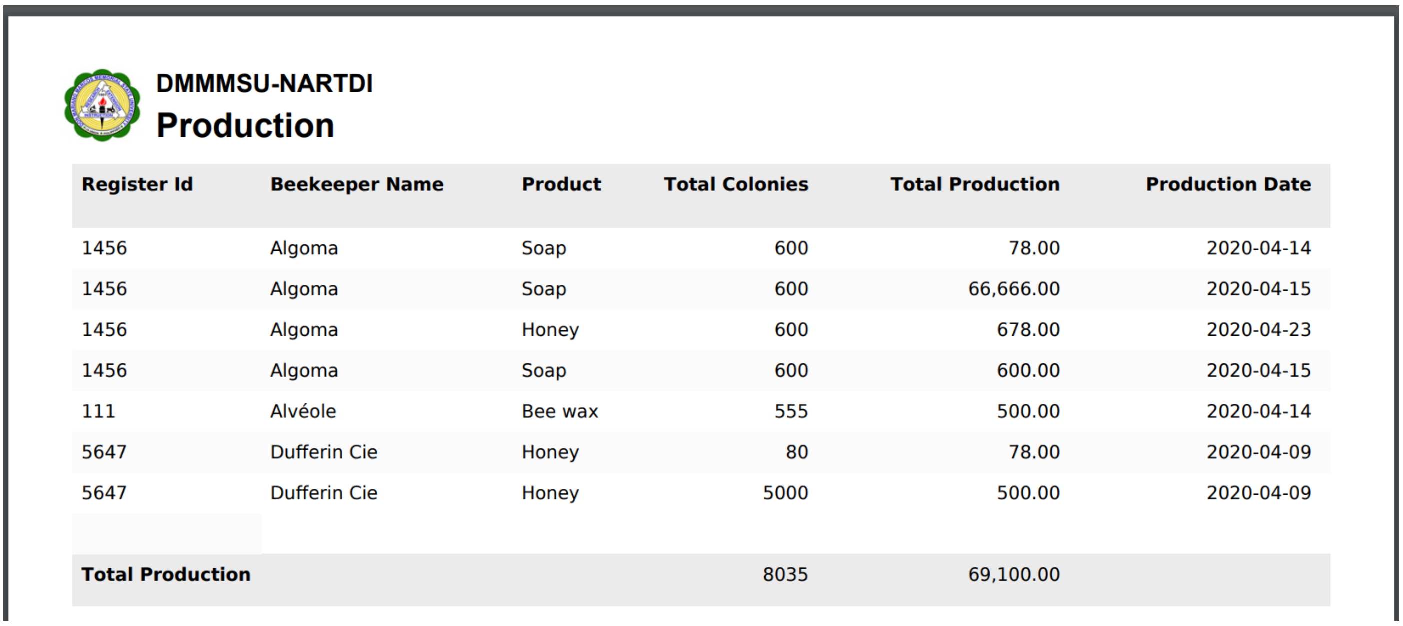
The call to the database will bring the record set and the percentage for each column must be the same

```
// Call to the database
$REP03 = $this->model_report->getREP03();

$grand_total_colony = 0;
$grand_total_production = 0;


if ($REP03 == null) {
    // If not data found, we indicate in the report first line
    $this->table->add_row($this->lang->line('No data found'));
}
else {
    foreach ($REP03 as $rs):
        $grand_total_colony =  $grand_total_colony + $rs->total_colony;
        $grand_total_production =  $grand_total_production + $rs->total_production;

        $cell1 = array('data' => $rs->beekeeper_register_id, 'width' => '15%');
        $cell2 = array('data' => $rs->beekeeper_name, 'width' => '20%');
        $cell3 = array('data' => $rs->product_name, 'width' => '10%');
        $cell4 = array('data' => $rs->total_colony, 'width' => '15%', 'align' => 'right');
        $cell5 = array('data' => number_format($rs->total_production,2), 'width' => '20%', 'align' => 'right');
        $cell6 = array('data' => $rs->production_date, 'width' => '20%', 'align' => 'right');
        $this->table->add_row($cell1, $cell2, $cell3, $cell4, $cell5, $cell6);
    endforeach;
}

$blank_line = array('data' => '');
$this->table->add_row($blank_line);

//Print the total

$cell1 = array('data' => '<strong>'.$this->lang->line('Total Production').'</strong>', 'width' => '45%', 'bgcolor'=> 'rgb(235,235,235
    , 'height'=>'23', 'valign' => 'center');
$cell2 = array('data' => $grand_total_colony, 'width' => '15%', 'align' => 'right', 'bgcolor'=> 'rgb(235,235,235)', 'height'=>'23');
$cell3 = array('data' => number_format($grand_total_production,2), 'width' => '20%', 'align' => 'right', 'bgcolor'=> 'rgb(235,235,235
    , 'height'=>'23');
$cell4 = array('data' => '', 'width' => '20%', 'align' => 'left', 'bgcolor'=> 'rgb(235,235,235)', 'height'=>'23');
$this->table->add_row($cell1, $cell2, $cell3, $cell4);
```

Then we call the OUTPUT to receive the PDF report in the report section of the system.

```
// Generate the table in html format using the table class of codeigniter
$html = $this->table->generate();

// Add a page and change the orientation/size
$pdf->AddPage('P','LETTER');

// Output the HTML content
$pdf->writeHTML($html, true, false, true, false, '');

// Reset pointer to the last page
$pdf->lastPage();

// Close and output PDF document
// (I - Inline, D - Download, F - File)
$pdf->Output('Report03.pdf', 'I');
```

Print a total for the report



DMMMSU-NARTDI

Production

Register Id	Beekeeper Name	Product	Total Colonies	Total Production	Production Date
1456	Algoma	Soap	600	78.00	2020-04-14
1456	Algoma	Soap	600	66,666.00	2020-04-15
1456	Algoma	Honey	600	678.00	2020-04-23
1456	Algoma	Soap	600	600.00	2020-04-15
111	Alvéole	Bee wax	555	500.00	2020-04-14
5647	Dufferin Cie	Honey	80	78.00	2020-04-09
5647	Dufferin Cie	Honey	5000	500.00	2020-04-09
Total Production			8035	69,100.00	

In controller/report03.php

I created a variable to add the total of each column.

You must assign 0 to the variable before beginning to cumulate the amount in each variable.

```
$grand_total_colony = 0;
$grand_total_production = 0;


if ($REP03 == null) {
    // If not data found, we indicate in the report first line
    $this->table->add_row($this->lang->line('No data found'));
}
else {
    foreach ($REP03 as $rs):
        $grand_total_colony =  $grand_total_colony + $rs->total_colony;
        $grand_total_production =  $grand_total_production + $rs->total_production;
```

After the variable is created, you can begin to add the content of each row. After all the lines have been created, you must print the total.

Main command TCPDF

`AddPage( $orientation = '', $format = '', $keepmargins = false, $tocpage = false )`

Adds a new page to the document. If a page is already present, the Footer() method is called first to output the footer (if enabled). Then the page is added, the current position set to the top-left corner according to the left and top margins (or top-right if in RTL mode), and Header() is called to display the header (if enabled). The origin of the coordinate system is at the top-left corner (or top-right for RTL) and increasing ordinates go downwards.

`endPage( $tocpage = false )`

Terminate the current page

`Ln( $h = '', $cell = false )`

Performs a line break. The current abscissa goes back to the left margin and the ordinate increases by the amount passed in parameter.

`GetX( )`

Returns the relative X value of current position. The value is relative to the left border for LTR languages and to the right border for RTL languages.

`GetAbsX( )`

Returns the absolute X value of current position.

`GetY( )`

Returns the ordinate of the current position.

`SetX( $x, $rtloff = false )`

Defines the abscissa of the current position. If the passed value is negative, it is relative to the right of the page (or left if language is RTL).

`SetY( $y, $resetx = true, $rtloff = false )`

Moves the current abscissa back to the left margin and sets the ordinate. If the passed value is negative, it is relative to the bottom of the page.

`SetXY( $x, $y, $rtloff = false )`

Defines the abscissa and ordinate of the current position. If the passed values are negative, they are relative respectively to the right and bottom of the page.

`SetAbsX( $x )`

Set the absolute X coordinate of the current pointer.

`SetAbsY( $y )`

Set the absolute Y coordinate of the current pointer.

`SetAbsXY( $x, $y )`

Set the absolute X and Y coordinates of the current pointer.

`SetMargins( $left, $top, $right = -1, $keepmargins = false )`

Defines the left, top and right margins.

`SetLeftMargin( $margin )`

Defines the left margin. The method can be called before creating the first page. If the current abscissa gets out of page, it is brought back to the margin.

`SetTopMargin( $margin )`

Defines the top margin. The method can be called before creating the first page.

`SetRightMargin( $margin )`

Defines the right margin. The method can be called before creating the first page.

`SetCellPadding( $pad )`

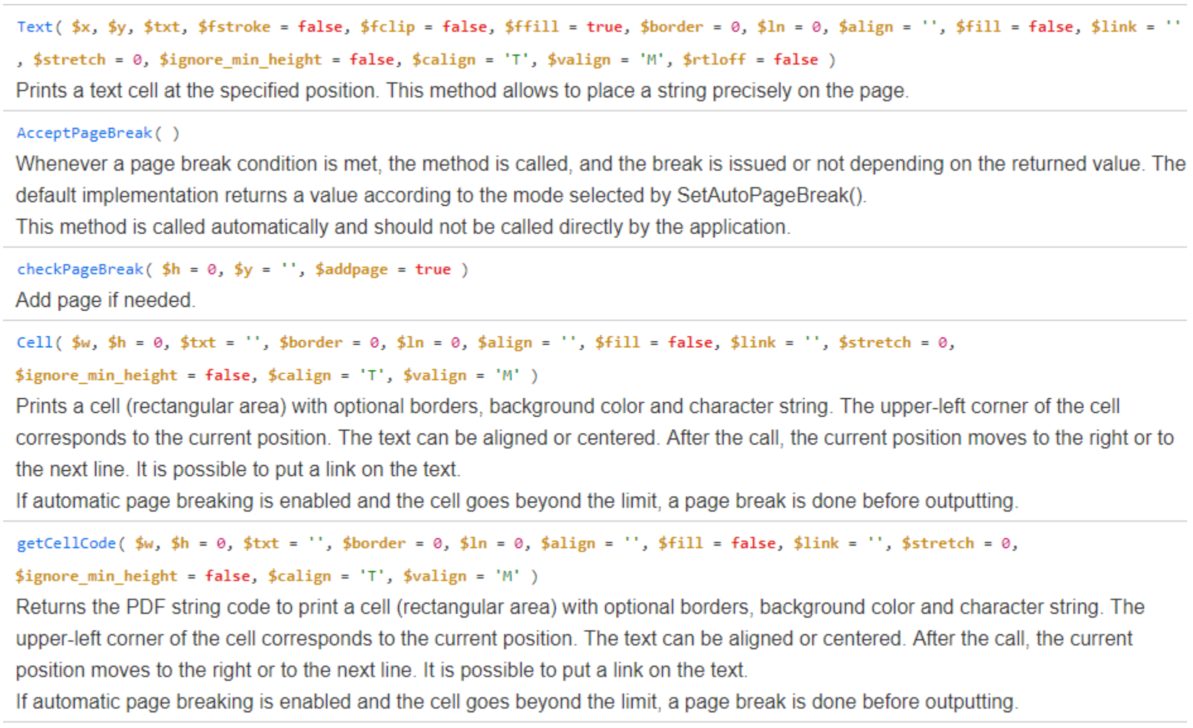
Set the same internal Cell padding for top, right, bottom, left-

`setCellPaddings( $left = '', $top = '', $right = '', $bottom = '' )`

Set the internal Cell paddings.

`getCellPaddings( )`

Get the internal Cell padding array.



`Text( $x, $y, $txt, $fstroke = false, $fclip = false, $ffill = true, $border = 0, $ln = 0, $align = '', $fill = false, $link = '' , $stretch = 0, $ignore_min_height = false, $calign = 'T', $valign = 'M', $rtloff = false )`

Prints a text cell at the specified position. This method allows to place a string precisely on the page.

`AcceptPageBreak( )`

Whenever a page break condition is met, the method is called, and the break is issued or not depending on the returned value. The default implementation returns a value according to the mode selected by SetAutoPageBreak().
This method is called automatically and should not be called directly by the application.

`checkPageBreak( $h = 0, $y = '', $addpage = true )`

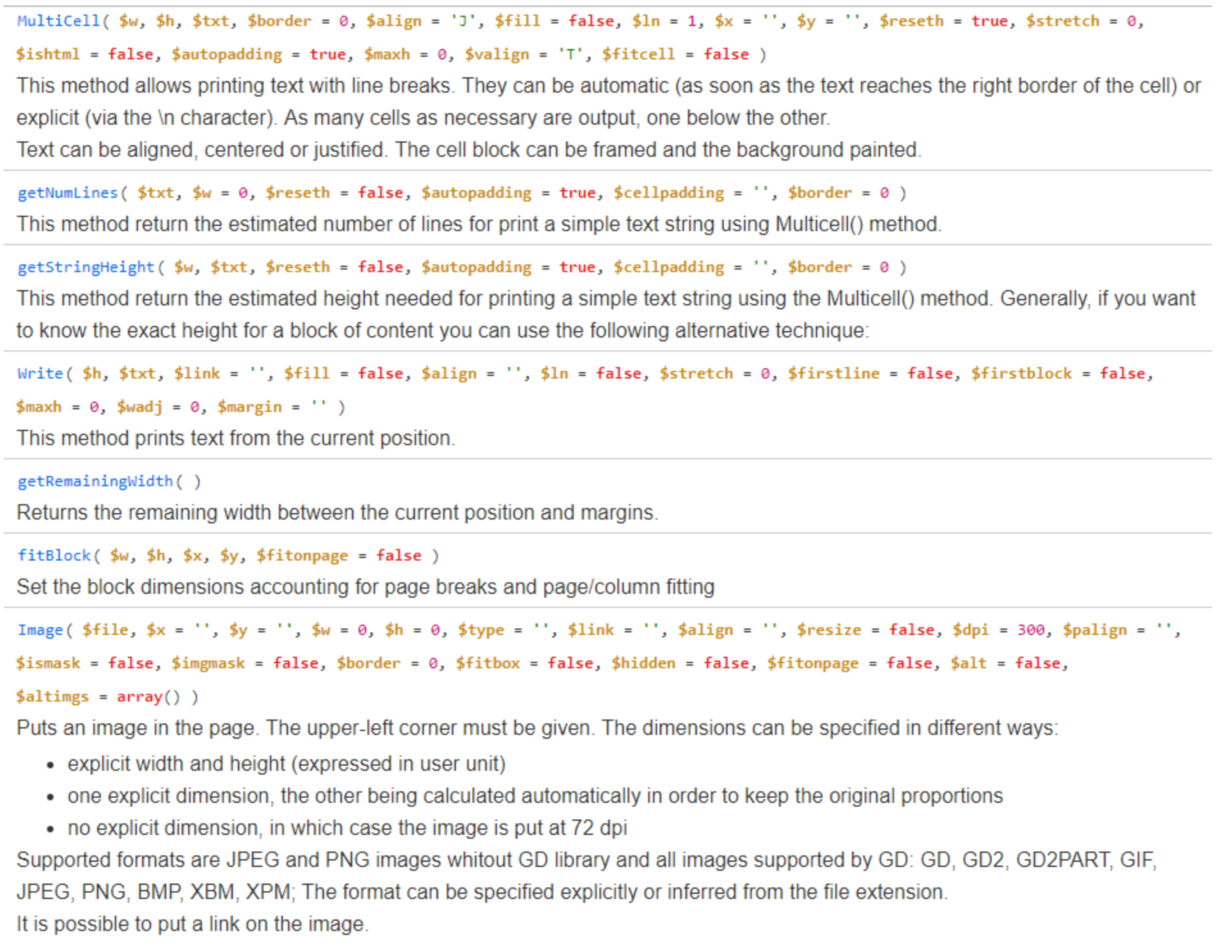
Add page if needed.

`Cell( $w, $h = 0, $txt = '', $border = 0, $ln = 0, $align = '', $fill = false, $link = '', $stretch = 0, $ignore_min_height = false, $calign = 'T', $valign = 'M' )`

Prints a cell (rectangular area) with optional borders, background color and character string. The upper-left corner of the cell corresponds to the current position. The text can be aligned or centered. After the call, the current position moves to the right or to the next line. It is possible to put a link on the text.
If automatic page breaking is enabled and the cell goes beyond the limit, a page break is done before outputting.

`getCellCode( $w, $h = 0, $txt = '', $border = 0, $ln = 0, $align = '', $fill = false, $link = '', $stretch = 0, $ignore_min_height = false, $calign = 'T', $valign = 'M' )`

Returns the PDF string code to print a cell (rectangular area) with optional borders, background color and character string. The upper-left corner of the cell corresponds to the current position. The text can be aligned or centered. After the call, the current position moves to the right or to the next line. It is possible to put a link on the text.
If automatic page breaking is enabled and the cell goes beyond the limit, a page break is done before outputting.



`MultiCell( $w, $h, $txt, $border = 0, $align = 'J', $fill = false, $ln = 1, $x = '', $y = '', $reseth = true, $stretch = 0, $ishtml = false, $autopadding = true, $maxh = 0, $valign = 'T', $fitcell = false )`

This method allows printing text with line breaks. They can be automatic (as soon as the text reaches the right border of the cell) or explicit (via the \n character). As many cells as necessary are output, one below the other.
Text can be aligned, centered or justified. The cell block can be framed and the background painted.

`getNumLines( $txt, $w = 0, $reseth = false, $autopadding = true, $cellpadding = '', $border = 0 )`

This method return the estimated number of lines for print a simple text string using Multicell() method.

`getStringHeight( $w, $txt, $reseth = false, $autopadding = true, $cellpadding = '', $border = 0 )`

This method return the estimated height needed for printing a simple text string using the Multicell() method. Generally, if you want to know the exact height for a block of content you can use the following alternative technique:

`Write( $h, $txt, $link = '', $fill = false, $align = '', $ln = false, $stretch = 0, $firstline = false, $firstblock = false, $maxh = 0, $wadj = 0, $margin = '' )`

This method prints text from the current position.

`getRemainingWidth( )`

Returns the remaining width between the current position and margins.

`fitBlock( $w, $h, $x, $y, $fitonpage = false )`

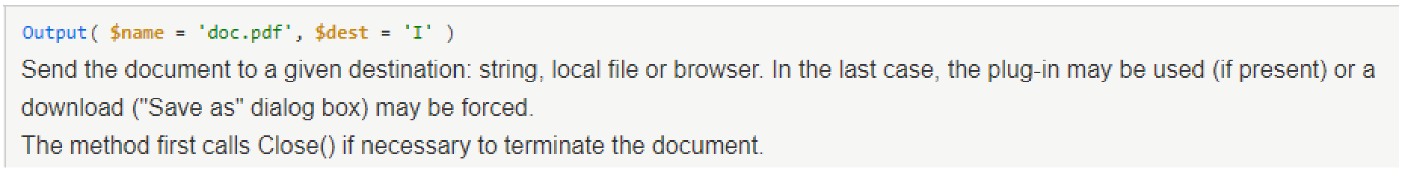
Set the block dimensions accounting for page breaks and page/column fitting

`Image( $file, $x = '', $y = '', $w = 0, $h = 0, $type = '', $link = '', $align = '', $resize = false, $dpi = 300, $palign = '', $ismask = false, $imgmask = false, $border = 0, $fitbox = false, $hidden = false, $fitonpage = false, $alt = false, $altimgs = array() )`

Puts an image in the page. The upper-left corner must be given. The dimensions can be specified in different ways:

- explicit width and height (expressed in user unit)
- one explicit dimension, the other being calculated automatically in order to keep the original proportions
- no explicit dimension, in which case the image is put at 72 dpi

Supported formats are JPEG and PNG images whitout GD library and all images supported by GD: GD, GD2, GD2PART, GIF, JPEG, PNG, BMP, XBM, XPM; The format can be specified explicitly or inferred from the file extension.
It is possible to put a link on the image.



`Output( $name = 'doc.pdf', $dest = 'I' )`

Send the document to a given destination: string, local file or browser. In the last case, the plug-in may be used (if present) or a download ("Save as" dialog box) may be forced.
The method first calls Close() if necessary to terminate the document.

Tools for finding errors

Print_r in PHP

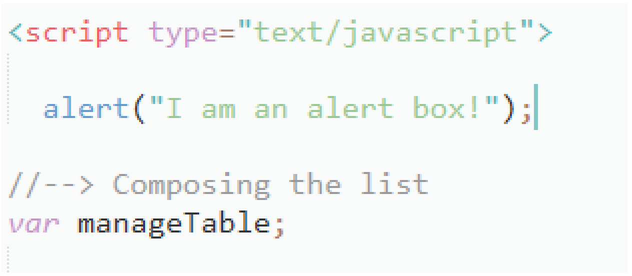
Print_r ($variable);
Die();

It will show the content of the variable but in CodeIgniter, you might need to kill the process after with Die() command. Normally it will be seen before the view but it's not visible all the time. CodeIgniter is using lots of different layers of information and sometimes this process will be overpassed.

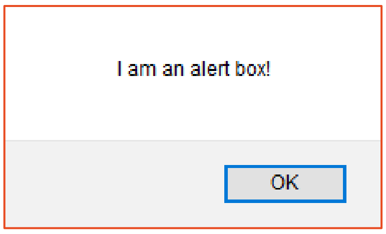
Alert() in Javascript

When trying to debug Javascript errors, you might use Alert box to see what's happening. If there is a syntax error in the javascript statement, it's not easy to find because all the script will not work.

The alert box *might* be used inside the javascript script to see where the problem is…



```
<script type="text/javascript">

  alert("I am an alert box!");

//--> Composing the list
var manageTable;
```

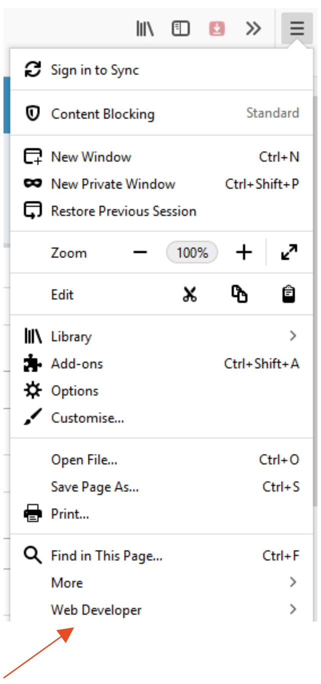


To see a parameter, it's a + and the parameter:

alert ('type activity'+type_activity);

Developer tools in browser

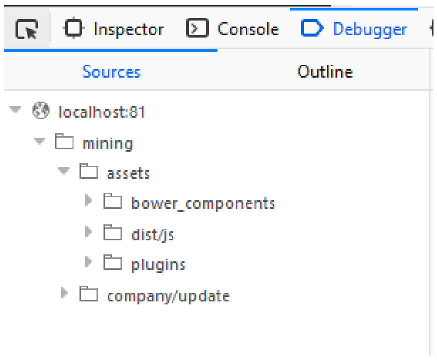
Chrome	**Firefox**	**Edge**
Open the browser. From the menu, select "More tools". From tools, choose "Developer tools". Finally, select Console.	Open the browser. From the menu, select "Web Developer". Finally, select "Web Console".	Open the browser. From the menu, select "Developer Tools". Finally, select "Console".



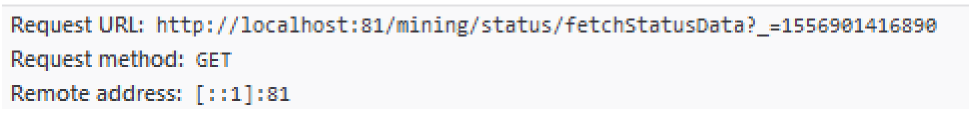
Developer tools in the browser are useful to see where the process is giving error. This is an example of Firefox Web Developer tool.

Inspector, console, debugger are all the tools that you might explore when you are debugging your system.

Here, using the debugger, you can see what are the sources used.



In the Network part, you will be able to find the function called …



```
Request URL: http://localhost:81/mining/status/fetchStatusData?_=1556901416890
Request method: GET
Remote address: [::1]:81
```

PHP Basics

PHP Comparison Operators

The PHP comparison operators are used to compare two values (number or string):

Operator	Name	Example	Result
==	Equal	$x == $y	Returns true if $x is equal to $y
===	Identical	$x === $y	Returns true if $x is equal to $y, and they are of the same type
!=	Not equal	$x != $y	Returns true if $x is not equal to $y
<>	Not equal	$x <> $y	Returns true if $x is not equal to $y
!==	Not identical	$x !== $y	Returns true if $x is not equal to $y, or they are not of the same type
>	Greater than	$x > $y	Returns true if $x is greater than $y
<	Less than	$x < $y	Returns true if $x is less than $y
>=	Greater than or equal to	$x >= $y	Returns true if $x is greater than or equal to $y
<=	Less than or equal to	$x <= $y	Returns true if $x is less than or equal to $y
<=>	Spaceship	$x <=> $y	Returns an integer less than, equal to, or greater than zero, depending on if $x is less than, equal to, or greater than $y. Introduced in PHP 7.

PHP Logical Operators

The PHP logical operators are used to combine conditional statements.

Operator	Name	Example	Result
and	And	$x and $y	True if both $x and $y are true
or	Or	$x or $y	True if either $x or $y is true
xor	Xor	$x xor $y	True if either $x or $y is true, but not both
&&	And	$x && $y	True if both $x and $y are true
||	Or	$x || $y	True if either $x or $y is true
!	Not	!$x	True if $x is not true

PHP Increment / Decrement Operators

The PHP increment operators are used to increment a variable's value.

The PHP decrement operators are used to decrement a variable's value.

Operator	Name	Description
++$x	Pre-increment	Increments $x by one, then returns $x
$x++	Post-increment	Returns $x, then increments $x by one
--$x	Pre-decrement	Decrements $x by one, then returns $x
$x--	Post-decrement	Returns $x, then decrements $x by one

If .. elseif ... else

```
if (condition) {
    code to be executed if this condition is true;
} elseif (condition) {
    code to be executed if first condition is false and this condition is true;
} else {
    code to be executed if all conditions are false;
}
```

Switch

```
switch (n) {
    case label1:
        code to be executed if n=label1;
        break;
    case label2:
        code to be executed if n=label2;
        break;
    case label3:
        code to be executed if n=label3;
        break;
    ...
    default:
        code to be executed if n is different from all labels;
}
```

Do ... while

```
do {
    code to be executed;
} while (condition is true);
```

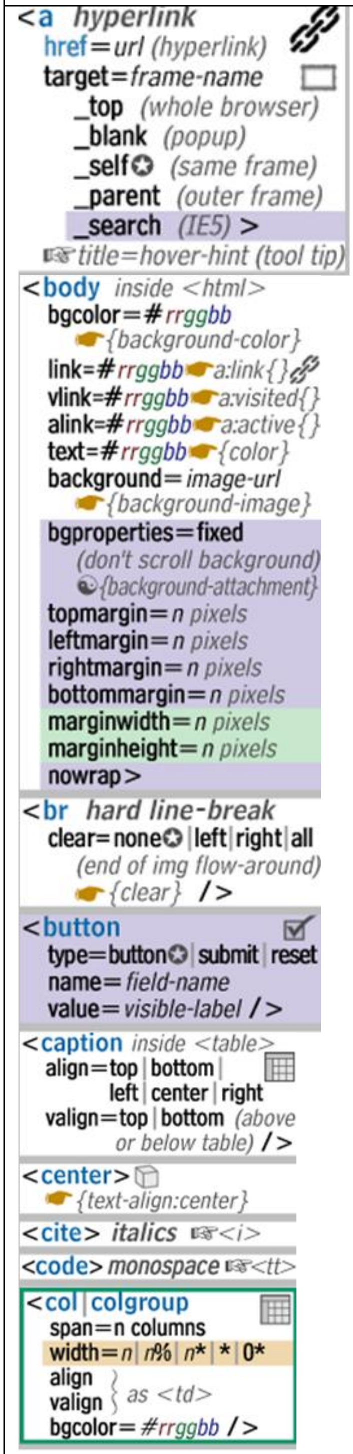
For and Foreach loop

```
for (init counter; test counter; increment counter) {
    code to be executed;
}
```

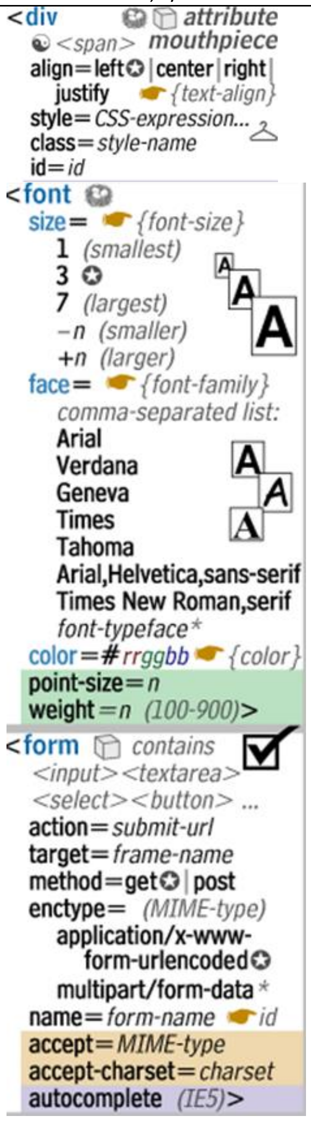
```
foreach ($array as $value) {
    code to be executed;
}
```

HTML Basics

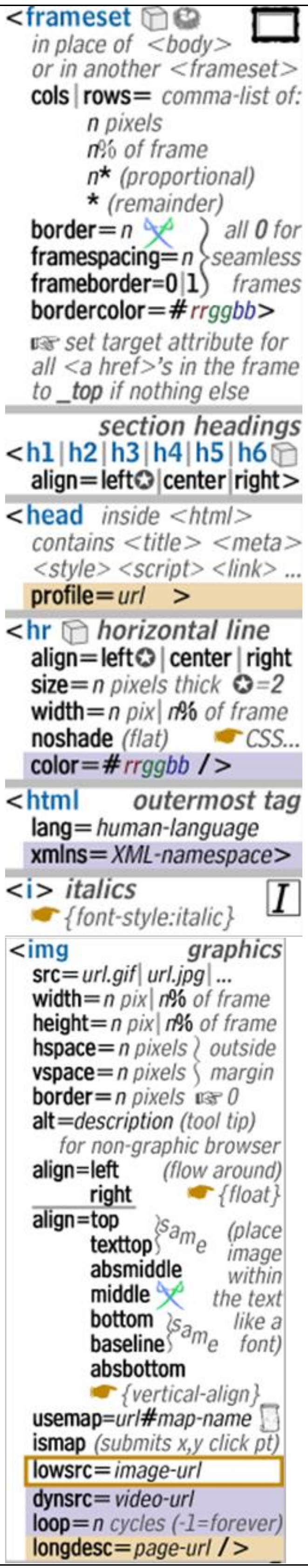
Name or ID
Name will identify the field in POST or GET. ID must be unique and is normally used in Javascript.
	To have a **bouton transparent**
	><input type="submit" value="Rechercher" style="color: transparent; background-color: transparent; border-color: transparent; cursor: default;" />
		Characters
		Espace = &nbsp;
		À = &Agrave;
		à = &agrave;
		â = &acirc;
			É = &Eacute;
			è = &egrave;
			é = &eacute;
			ê = &ecirc;



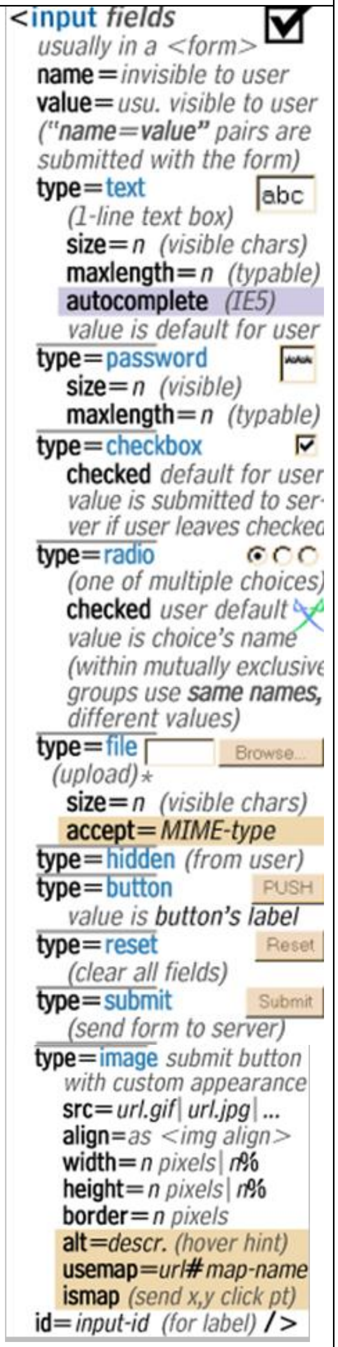
```
<a  hyperlink
  href=url (hyperlink)
  target=frame-name
    _top   (whole browser)
    _blank  (popup)
    _self  (same frame)
    _parent  (outer frame)
    _search  (IE5) >
  title=hover-hint (tool tip)
<body  inside <html>
  bgcolor=#rrggbb
    {background-color}
  link=#rrggbb a:link{}
  vlink=#rrggbb a:visited{}
  alink=#rrggbb a:active{}
  text=#rrggbb {color}
  background=image-url
    {background-image}
  bgproperties=fixed
    (don't scroll background)
    {background-attachment}
  topmargin=n pixels
  leftmargin=n pixels
  rightmargin=n pixels
  bottommargin=n pixels
  marginwidth=n pixels
  marginheight=n pixels
  nowrap>
<br  hard line-break
  clear=none|left|right|all
    (end of img flow-around)
    {clear}  />
<button
  type=button|submit|reset
  name=field-name
  value=visible-label />
<caption inside <table>
  align=top|bottom|
        left|center|right
  valign=top|bottom (above
        or below table) />
<center>
  {text-align:center}
<cite>  italics  <i>
<code> monospace <tt>
<col|colgroup
  span=n columns
  width=n|n%|n*|*|0*
  align  }
  valign } as <td>
  bgcolor=#rrggbb />
```



```
<div              attribute
  <span>  mouthpiece
  align=left|center|right|
    justify    {text-align}
  style=CSS-expression...
  class=style-name
  id=id
<font
  size=  {font-size}
    1  (smallest)
    3
    7  (largest)
    -n  (smaller)
    +n  (larger)
  face=  {font-family}
    comma-separated list:
    Arial
    Verdana
    Geneva
    Times
    Tahoma
    Arial,Helvetica,sans-serif
    Times New Roman,serif
    font-typeface*
  color=#rrggbb {color}
  point-size=n
  weight=n  (100-900)>
<form   contains
  <input><textarea>
  <select><button> ...
  action=submit-url
  target=frame-name
  method=get|post
  enctype=  (MIME-type)
    application/x-www-
      form-urlencoded
    multipart/form-data*
  name=form-name   id
  accept=MIME-type
  accept-charset=charset
  autocomplete  (IE5)>
```



```
<frameset
  in place of  <body>
  or in another <frameset>
  cols|rows= comma-list of:
    n pixels
    n% of frame
    n* (proportional)
    * (remainder)
  border=n          } all 0 for
  framespacing=n    } seamless
  frameborder=0|1   } frames
  bordercolor=#rrggbb>
  set target attribute for
  all <a href>'s in the frame
  to _top if nothing else
             section headings
<h1|h2|h3|h4|h5|h6
  align=left|center|right>
<head  inside <html>
  contains <title> <meta>
  <style> <script> <link> ...
  profile=url   >
<hr  horizontal line
  align=left|center|right
  size=n pixels thick  =2
  width=n pix| n% of frame
  noshade (flat)    CSS...
  color=#rrggbb />
<html        outermost tag
  lang=human-language
  xmlns=XML-namespace>
<i> italics
  {font-style:italic}
<img               graphics
  src=url.gif|url.jpg|...
  width=n pix|n% of frame
  height=n pix|n% of frame
  hspace=n pixels } outside
  vspace=n pixels } margin
  border=n pixels  0
  alt=description (tool tip)
    for non-graphic browser
  align=left     (flow around)
        right        {float}
  align=top      }same   (place
        texttop  }       image
        absmiddle        within
        middle         the text
        bottom   }same   like a
        baseline }       font)
        absbottom
          {vertical-align}
  usemap=url#map-name
  ismap (submits x,y click pt)
  lowsrc=image-url
  dynsrc=video-url
  loop=n cycles (-1=forever)
  longdesc=page-url />
```



```
<input fields
  usually in a <form>
  name=invisible to user
  value=usu. visible to user
  ("name=value" pairs are
  submitted with the form)
  type=text
    (1-line text box)
    size=n  (visible chars)
    maxlength=n  (typable)
    autocomplete  (IE5)
    value is default for user
  type=password
    size=n  (visible)
    maxlength=n  (typable)
  type=checkbox
    checked default for user
    value is submitted to ser-
    ver if user leaves checked
  type=radio
    (one of multiple choices)
    checked user default
    value is choice's name
    (within mutually exclusive
    groups use same names,
    different values)
  type=file
   (upload)*
    size=n  (visible chars)
    accept=MIME-type
  type=hidden (from user)
  type=button
    value is button's label
  type=reset
    (clear all fields)
  type=submit
    (send form to server)
  type=image submit button
    with custom appearance
    src=url.gif|url.jpg|...
    align=as <img align>
    width=n pixels|n%
    height=n pixels|n%
    border=n pixels
    alt=descr. (hover hint)
    usemap=url#map-name
    ismap (send x,y click pt)
  id=input-id  (for label) />
```

<li numbered or bulleted paragraph
inside <ol> or <ul>
type=1|a|i|A|I
type=disc|circle|square
(as in <ol> and <ul>)
value=*n (alter sequence)*>

<link file relation
inside <head>
rel=stylesheet) *by far the*
type=text/css } *most com-*
href=*url.css*) *mon link*
rel=shortcut icon *(IE5)*
rel=*(other relationships)*
contents|next|prev|...
rev=*(reverse relation)*
type=*MIME-type*
media=screen|print|all✪
target=*frame-name*
charset=*character-set* />

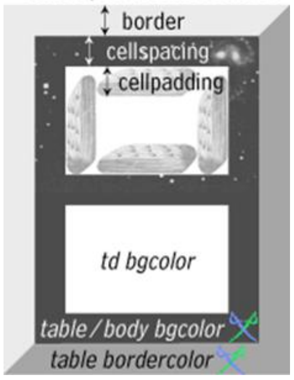
<optgroup> *visual grouping of <option>'s*

<option *multiple choice*
inside <select>
value=*choice-name*
selected *(user's default)* />

<p paragraph
align=left✪|center|right|
justify> ☛{text-align}

<param *inside <applet> <embed> <object>*
valuetype=data|ref|object
name=*name*
value=*value* />

Table parts and colors



<table
contains <tr>
<colgroup><col><caption>
<thead> <tfoot> <tbody>
border=*n pixels* ✪=0
(around outside of table)
{border-width}
cellspacing=*n pix* ✪=2 to 4
(around & between cells)
cellpadding=*n pixels* ✪=1
(extra room inside cells)
{padding}
bgcolor=#*rrggbb*
☛*{background-color}*
background=*url.gif*|*url.jpg*
{background-image}
bordercolor=#*rrggbb*
bordercolorlight=#*rrggbb*
bordercolordark=#*rrggbb*
width=*n pix*|*n% of frame*
height=*n pix*|*n% of frame*
hspace=*n pixels*) *outside*
vspace=*n pixels*) *margin*
align=left|center|right
☛*{float}*
cols=*n virtual columns*
frame=*(around table)*
void|box|border✪|
above|below|lhs|
rhs|hsides|vsides
rules=*(between cells)*
none|all✪|groups|
cols|rows
summary=*summary*>

<select multiple choice drop-down
inside <form>
contains <option>'s
name=*field-name*
size=*n rows when open*
multiple *simult. answers*
select-one ✪
select-multiple>

<td table cell
inside <tr>
align=left✪|center|right|
justify ☛*{text-align}*
align=char) *e.g. align*
char=.|,|... } *by decimal*
charoff=*n*) *points*
valign=top|middle✪|
bottom|baseline
{vertical-align}
bgcolor=#*rrggbb*
☛*{background-color}*
background=*url.gif*|*url.jpg*
{background-image}
bordercolor=#*rrggbb*
colspan|rowspan=
1✪
2...*n (wide col, tall row)*
0 *(remainder of group)*
nowrap *(all on one line)*
☛*{white-space:nowrap}*
width=*n pix*|*n% of frame*
☛*{width}*
height=*n pix*|*n% of frame*
☛*{height}*
headers=*id, id, ...*
scope=row|col|...
abbr=*brief* *<img alt>*
axis=*categories*>

<textarea multi-line text entry
inside <form>
(content is user's default)
cols=*visual-width* *(no limit*
rows=*visual-lines* *to data)*
wrap= ☞ *wrap=soft*
soft✪ *visually wraps, submits data unwrapped*
hard *visually wraps, submits with hard returns*
off✪ *window scrolls rather than wrap text*
readonly
name=*field-name*>

<th> header cell ☞*<td>*
same as <td> (with bold)

<title>! *inside <head>*
(page title if bookmarked)

<tr table row
inside <table>
contains <td><th>
align)
valign } *as <td>*
width=*n pix*|*n% of frame*
height=*n pix*|*n% of frame*
bgcolor=#*rrggbb*
bordercolor=#*rrggbb*>

<tt> monospace
<pre>
☛*{font-family:monospace}*

<u> underline
☛*{text-decoration: underline}*

<ul bulleted list
contains <li>'s *<ol>*
type=disc✪ *1st level* ●
circle✪ *2nd level* ○
square✪ *3rd, 4th...* ■
compact>

Event Attributes
(values usually JavaScript)
onload
onunload
onresize
onscroll
onclick
ondblclick
onkeydown
onkeyup
onkeypress
onmouseover
onmousemove
onmousedown
onmouseup
onmouseout
onchange
onerror
onfocus
onblur
onsubmit
onreset
ondragstart)
ondrag } *drag source events*
ondragend)
ondragenter)
ondragover } *drag target events*
ondragleave
ondrop)
oncopy
oncut
onpaste

<script JavaScript...
language= ☛*type*
javascript✪|jscript
javascript 1.2
vbscript|vbs
xml *(IE5)*
type=*(MIME type)*
text/javascript
text/vbscript
text/xml *(IE5)*
src=*url.js* *(script in a file as opposed to the content)*
for=*object-id*
event=*eventname*
defer *(faster load)*
charset=*charset*>

Browns

Color Name	RGB CODE	HEX #	Sample
Rosy Brown	188-143-143	bc8f8f	🟫
Indian Red	205-92-92	cd5c5c	🟥
Saddle Brown	139-69-19	8b4513	🟫
Sienna	160-82-45	a0522d	🟫
Peru	205-133-63	cd853f	🟧
Burlywood	222-184-135	deb887	🟫
Beige	245-245-220	f5f5dc	⬜
Wheat	245-222-179	f5deb3	🟨
Sandy Brown	244-164-96	f4a460	🟧
Tan	210-180-140	d2b48c	🟫
Chocolate	210-105-30	d2691e	🟧
Firebrick	178-34-34	b22222	🟥
Brown	165-42-42	a52a2a	🟥

Oranges

Color Name	RGB CODE	HEX #	Sample
Dark Salmon	233-150-122	e9967a	🟧
Salmon	250-128-114	fa8072	🟥
Light Salmon	255-160-122	ffa07a	🟧
Orange	255-165-0	ffa500	🟧
Dark Orange	255-140-0	ff8c00	🟧
Coral	255-127-80	ff7f50	🟧
Light Coral	240-128-128	f08080	🟥
Tomato	255-99-71	ff6347	🟥
Orange Red	255-69-0	ff4500	🟥
Red	255-0-0	ff0000	🟥

Whites/Pastels

Color Name	RGB CODE	HEX #	Sample
Snow	255-250-250	fffafa	⬜
Snow 2	238-233-233	eee9e9	⬜
Snow 3	205-201-201	cdc9c9	⬜
Snow 4	139-137-137	8b8989	⬛
Ghost White	248-248-255	f8f8ff	⬜
White Smoke	245-245-245	f5f5f5	⬜
Gainsboro	220-220-220	dcdcdc	⬜
Floral White	255-250-240	fffaf0	⬜
Old Lace	253-245-230	fdf5e6	⬜
Linen	240-240-230	faf0e6	⬜
Antique White	250-235-215	faebd7	⬜
Antique White 2	238-223-204	eedfcc	⬜
Antique White 3	205-192-176	cdc0b0	🟫
Antique White 4	139-131-120	8b8378	🟫

Grays

Color Name	RGB CODE	HEX #	Sample
Black	0-0-0	000000	⬛
Dark Slate Gray	49-79-79	2f4f4f	⬛
Dim Gray	105-105-105	696969	⬛
Slate Gray	112-138-144	708090	⬛
Light Slate Gray	119-136-153	778899	⬛
Gray	190-190-190	bebebe	⬜
Light Gray	211-211-211	d3d3d3	⬜

Blues

Color Name	RGB CODE	HEX #	Sample
Midnight Blue	25-25-112	191970	🟦
Navy	0-0-128	000080	🟦
Cornflower Blue	100-149-237	6495ed	🟦
Dark Slate Blue	72-61-139	483d8b	🟪
Slate Blue	106-90-205	6a5acd	🟪
Medium Slate Blue	123-104-238	7b68ee	🟪
Light Slate Blue	132-112-255	8470ff	🟪
Medium Blue	0-0-205	0000cd	🟦
Royal Blue	65-105-225	4169e1	🟦
Blue	0-0-255	0000ff	🟦
Dodger Blue	30-144-255	1e90ff	🟦
Deep Sky Blue	0-191-255	00bfff	🟦
Sky Blue	135-206-250	87ceeb	🟦
Light Sky Blue	135-206-250	87cefa	🟦
Steel Blue	70-130-180	4682b4	🟦
Light Steel Blue	176-196-222	b0c4de	🟦
Light Blue	173-216-230	add8e6	🟦
Powder Blue	176-224-230	b0e0e6	🟦
Pale Turquoise	175-238-238	afeeee	🟦
Dark Turquoise	0-206-209	00ced1	🟦
Medium Turquoise	72-209-204	48d1cc	🟦
Turquoise	64-224-208	40e0d0	🟦
Cyan	0-255-255	00ffff	🟦
Light Cyan	224-255-255	e0ffff	⬜
Cadet Blue	95-158-160	5f9ea0	🟦

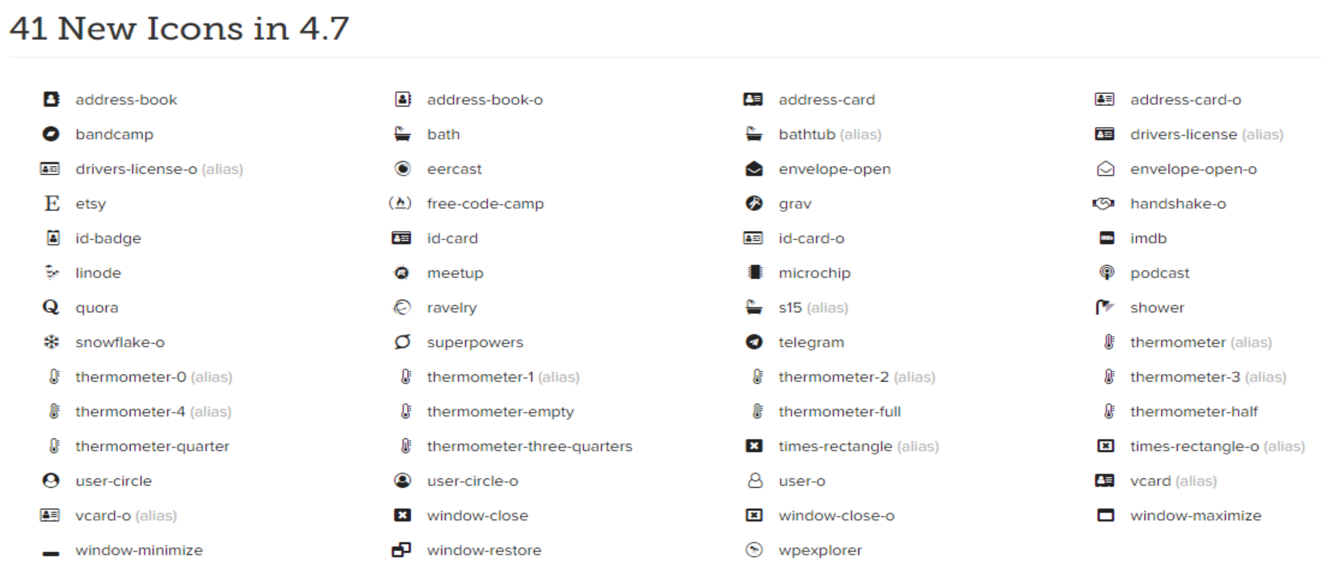
Greens

Color Name	RGB CODE	HEX #	Sample
Medium Aquamarine	102-205-170	66cdaa	🟩
Aquamarine	127-255-212	7fffd4	🟩
Dark Green	0-100-0	006400	🟩
Dark Olive Green	85-107-47	556b2f	🟩
Dark Sea Green	143-188-143	8fbc8f	🟩
Sea Green	46-139-87	2e8b57	🟩
Medium Sea Green	60-179-113	3cb371	🟩
Light Sea Green	32-178-170	20b2aa	🟩
Pale Green	152-251-152	98fb98	🟩
Spring Green	0-255-127	00ff7f	🟩
Lawn Green	124-252-0	7cfc00	🟩
Chartreuse	127-255-0	7fff00	🟩
Medium Spring Green	0-250-154	00fa9a	🟩
Green Yellow	173-255-47	adff2f	🟩
Lime Green	50-205-50	32cd32	🟩
Yellow Green	154-205-50	9acd32	🟩
Forest Green	34-139-34	228b22	🟩
Olive Drab	107-142-35	6b8e23	🟩
Dark Khaki	189-183-107	bdb76b	🟨
Khaki	240-230-140	f0e68c	🟨

Yellow

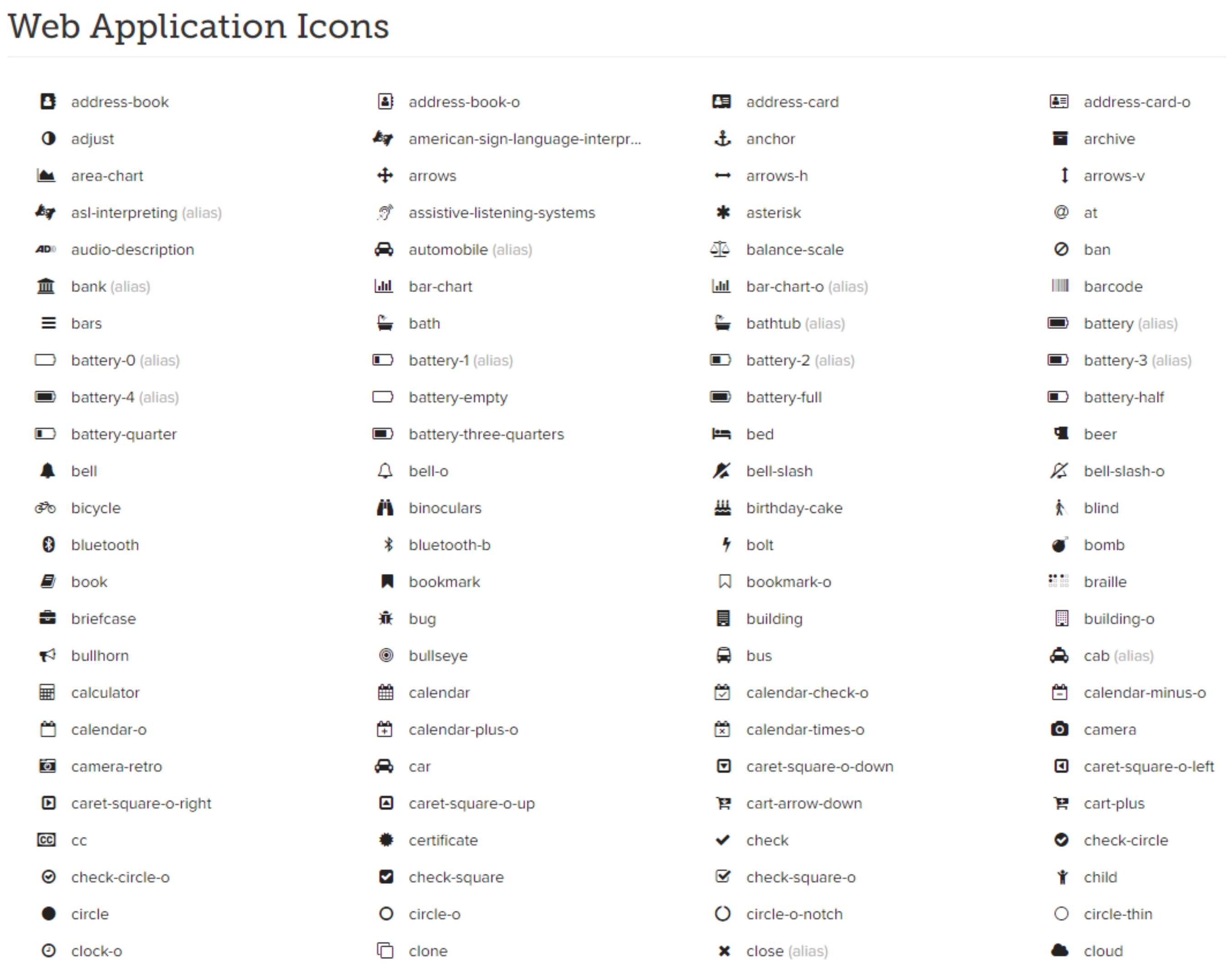
Color Name	RGB CODE	HEX #	Sample
Pale Goldenrod	238-232-170	eee8aa	🟨
Light Goldenrod Yellow	250-250-210	fafad2	🟨
Light Yellow	255-255-224	ffffe0	🟨
Yellow	255-255-0	ffff00	🟨
Gold	255-215-0	ffd700	🟨
Light Goldenrod	238-221-130	eedd82	🟨
Goldenrod	218-165-32	daa520	🟨
Dark Goldenrod	184-134-11	b8860b	🟨

Font-Awesome (FA)



41 New Icons in 4.7

address-book	address-book-o	address-card	address-card-o
bandcamp	bath	bathtub (alias)	drivers-license (alias)
drivers-license-o (alias)	eercast	envelope-open	envelope-open-o
etsy	free-code-camp	grav	handshake-o
id-badge	id-card	id-card-o	imdb
linode	meetup	microchip	podcast
quora	ravelry	s15 (alias)	shower
snowflake-o	superpowers	telegram	thermometer (alias)
thermometer-0 (alias)	thermometer-1 (alias)	thermometer-2 (alias)	thermometer-3 (alias)
thermometer-4 (alias)	thermometer-empty	thermometer-full	thermometer-half
thermometer-quarter	thermometer-three-quarters	times-rectangle (alias)	times-rectangle-o (alias)
user-circle	user-circle-o	user-o	vcard (alias)
vcard-o (alias)	window-close	window-close-o	window-maximize
window-minimize	window-restore	wpexplorer	



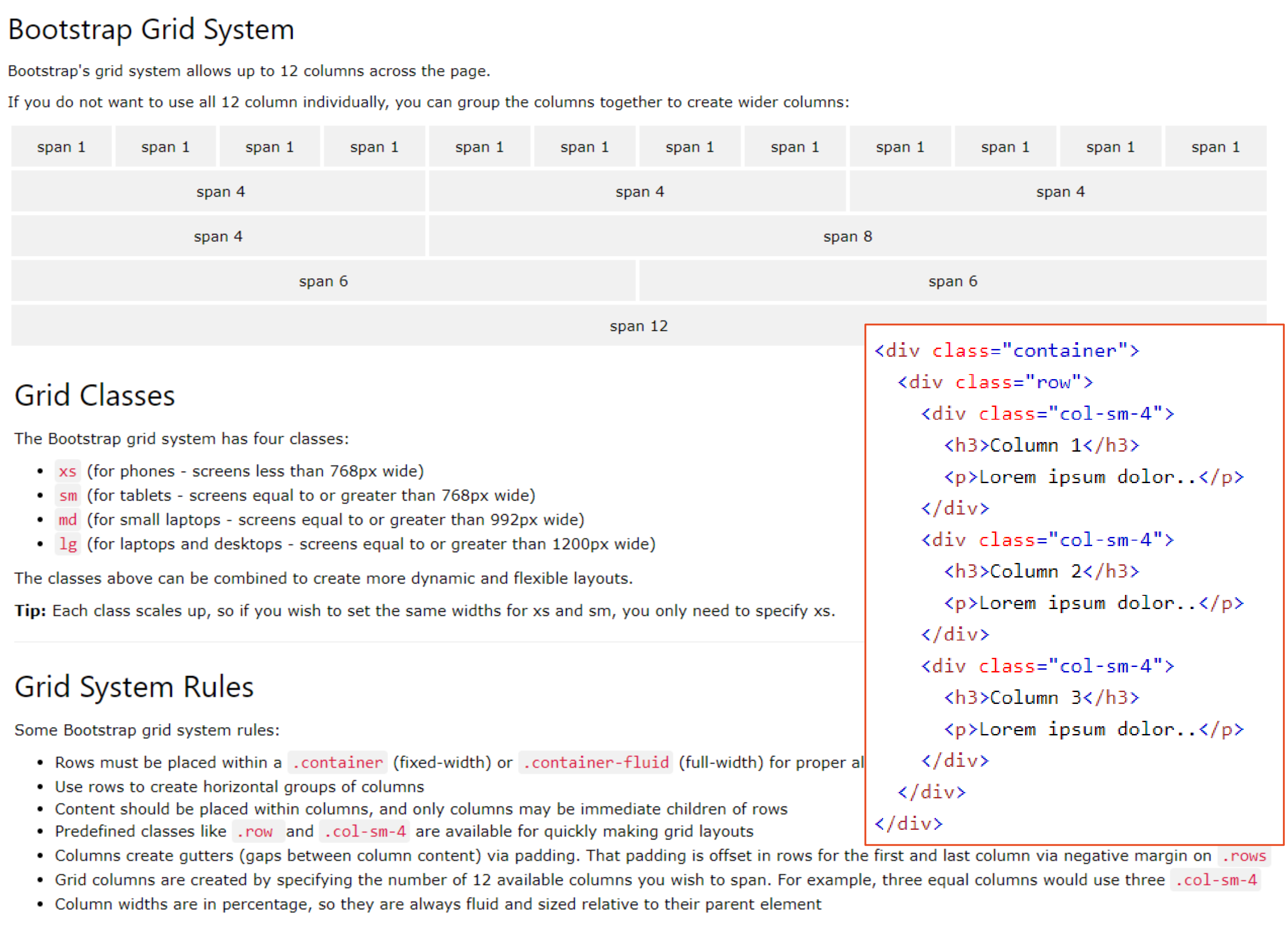
Web Application Icons

address-book	address-book-o	address-card	address-card-o
adjust	american-sign-language-interpr...	anchor	archive
area-chart	arrows	arrows-h	arrows-v
asl-interpreting (alias)	assistive-listening-systems	asterisk	at
audio-description	automobile (alias)	balance-scale	ban
bank (alias)	bar-chart	bar-chart-o (alias)	barcode
bars	bath	bathtub (alias)	battery (alias)
battery-0 (alias)	battery-1 (alias)	battery-2 (alias)	battery-3 (alias)
battery-4 (alias)	battery-empty	battery-full	battery-half
battery-quarter	battery-three-quarters	bed	beer
bell	bell-o	bell-slash	bell-slash-o
bicycle	binoculars	birthday-cake	blind
bluetooth	bluetooth-b	bolt	bomb
book	bookmark	bookmark-o	braille
briefcase	bug	building	building-o
bullhorn	bullseye	bus	cab (alias)
calculator	calendar	calendar-check-o	calendar-minus-o
calendar-o	calendar-plus-o	calendar-times-o	camera
camera-retro	car	caret-square-o-down	caret-square-o-left
caret-square-o-right	caret-square-o-up	cart-arrow-down	cart-plus
cc	certificate	check	check-circle
check-circle-o	check-square	check-square-o	child
circle	circle-o	circle-o-notch	circle-thin
clock-o	clone	close (alias)	cloud

id-card-o	image (alias)	inbox	industry
info	info-circle	institution (alias)	key
keyboard-o	language	laptop	leaf
legal (alias)	lemon-o	level-down	level-up
life-bouy (alias)	life-buoy (alias)	life-ring	life-saver (alias)
lightbulb-o	line-chart	location-arrow	lock
low-vision	magic	magnet	mail-forward (alias)
mail-reply (alias)	mail-reply-all (alias)	male	map
map-marker	map-o	map-pin	map-signs
meh-o	microchip	microphone	microphone-slash
minus	minus-circle	minus-square	minus-square-o
mobile	mobile-phone (alias)	money	moon-o
mortar-board (alias)	motorcycle	mouse-pointer	music
navicon (alias)	newspaper-o	object-group	object-ungroup
paint-brush	paper-plane	paper-plane-o	paw
pencil	pencil-square	pencil-square-o	percent
phone	phone-square	photo (alias)	picture-o
pie-chart	plane	plug	plus
plus-circle	plus-square	plus-square-o	podcast
power-off	print	puzzle-piece	qrcode
question	question-circle	question-circle-o	quote-left
quote-right	random	recycle	refresh
registered	remove (alias)	reorder (alias)	reply
reply-all	retweet	road	rocket
rss	rss-square	s15 (alias)	search
search-minus	search-plus	send (alias)	send-o (alias)
server	share	share-alt	share-alt-square
share-square	share-square-o	shield	ship
shopping-bag	shopping-basket	shopping-cart	shower
sign-in	sign-language	sign-out	signal
signing (alias)	sitemap	sliders	smile-o
snowflake-o	soccer-ball-o (alias)	sort	sort-alpha-asc
sort-alpha-desc	sort-amount-asc	sort-amount-desc	sort-asc
sort-desc	sort-down (alias)	sort-numeric-asc	sort-numeric-desc
sort-up (alias)	space-shuttle	spinner	spoon
square	square-o	star	star-half
star-half-empty (alias)	star-half-full (alias)	star-half-o	star-o
sticky-note	sticky-note-o	street-view	suitcase
sun-o	support (alias)	tablet	tachometer
tag	tags	tasks	taxi
television	terminal	thermometer (alias)	thermometer-0 (alias)
thermometer-1 (alias)	thermometer-2 (alias)	thermometer-3 (alias)	thermometer-4 (alias)
thermometer-empty	thermometer-full	thermometer-half	thermometer-quarter
thermometer-three-quarters	thumb-tack	thumbs-down	thumbs-o-down
thumbs-o-up	thumbs-up	ticket	times
times-circle	times-circle-o	times-rectangle (alias)	times-rectangle-o (alias)
tint	toggle-down (alias)	toggle-left (alias)	toggle-off
toggle-on	toggle-right (alias)	toggle-up (alias)	trademark
trash	trash-o	tree	trophy
truck	tty	tv (alias)	umbrella
universal-access	university	unlock	unlock-alt
unsorted (alias)	upload	user	user-circle
user-circle-o	user-o	user-plus	user-secret
user-times	users	vcard (alias)	vcard-o (alias)
video-camera	volume-control-phone	volume-down	volume-off
volume-up	warning (alias)	wheelchair	wheelchair-alt
wifi	window-close	window-close-o	window-maximize

Bootstrap basics

Grid System



Bootstrap Grid System

Bootstrap's grid system allows up to 12 columns across the page.

If you do not want to use all 12 column individually, you can group the columns together to create wider columns:

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

Grid Classes

The Bootstrap grid system has four classes:

- `xs` (for phones - screens less than 768px wide)
- `sm` (for tablets - screens equal to or greater than 768px wide)
- `md` (for small laptops - screens equal to or greater than 992px wide)
- `lg` (for laptops and desktops - screens equal to or greater than 1200px wide)

The classes above can be combined to create more dynamic and flexible layouts.

Tip: Each class scales up, so if you wish to set the same widths for xs and sm, you only need to specify xs.

Grid System Rules

Some Bootstrap grid system rules:

- Rows must be placed within a `.container` (fixed-width) or `.container-fluid` (full-width) for proper al
- Use rows to create horizontal groups of columns
- Content should be placed within columns, and only columns may be immediate children of rows
- Predefined classes like `.row` and `.col-sm-4` are available for quickly making grid layouts
- Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and last column via negative margin on `.rows`
- Grid columns are created by specifying the number of 12 available columns you wish to span. For example, three equal columns would use three `.col-sm-4`
- Column widths are in percentage, so they are always fluid and sized relative to their parent element

```
<div class="container">
  <div class="row">
    <div class="col-sm-4">
      <h3>Column 1</h3>
      <p>Lorem ipsum dolor..</p>
    </div>
    <div class="col-sm-4">
      <h3>Column 2</h3>
      <p>Lorem ipsum dolor..</p>
    </div>
    <div class="col-sm-4">
      <h3>Column 3</h3>
      <p>Lorem ipsum dolor..</p>
    </div>
  </div>
</div>
```

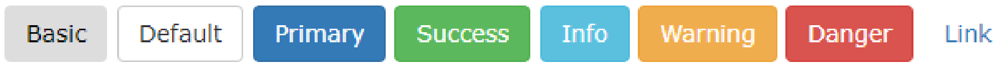
Alert

Alerts are created with the .alert class, followed by one of the four contextual classes .alert-success, .alert-info, .alert-warning or .alert-danger:

```
<div class="alert alert-success">
  <strong>Success!</strong> Indicates a successful or positive action.
</div>

<div class="alert alert-info">
  <strong>Info!</strong> Indicates a neutral informative change or action.
</div>

<div class="alert alert-warning">
  <strong>Warning!</strong> Indicates a warning that might need attention.
</div>

<div class="alert alert-danger">
  <strong>Danger!</strong> Indicates a dangerous or potentially negative action.
</div>
```

Button

Bootstrap provides different styles of buttons:



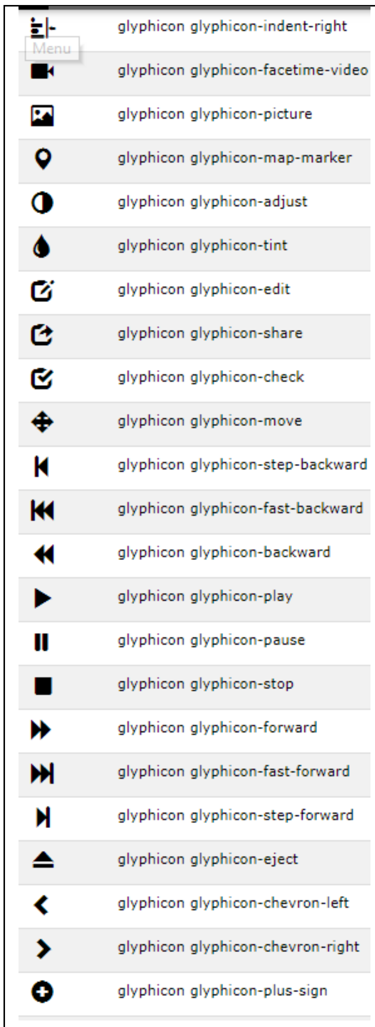
To achieve the button styles above, Bootstrap has the following classes:

- `.btn`
- `.btn-default`
- `.btn-primary`
- `.btn-success`
- `.btn-info`
- `.btn-warning`
- `.btn-danger`
- `.btn-link`

```
<button type="button" class="btn">Basic</button>
<button type="button" class="btn btn-default">Default</button>
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-link">Link</button>
```

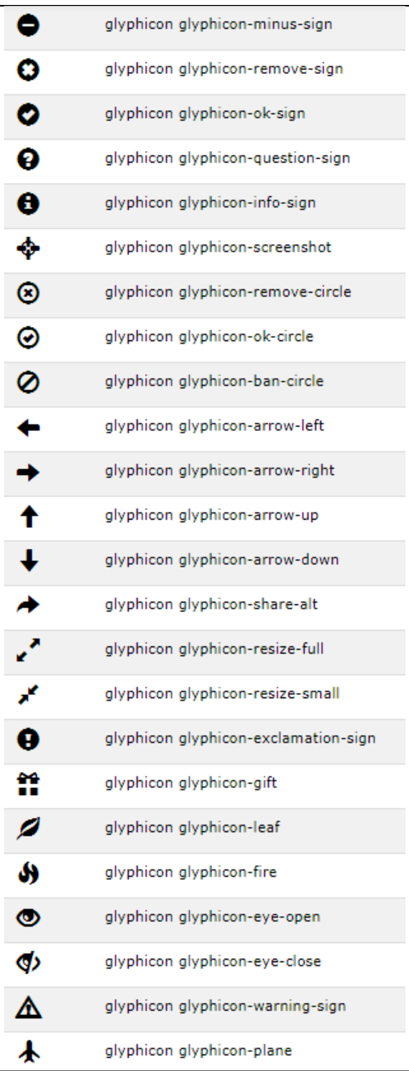
Glyphicon

Glyph	Description
✱	glyphicon glyphicon-asterisk
✚	glyphicon glyphicon-plus
▬	glyphicon glyphicon-minus
€	glyphicon glyphicon-euro
☁	glyphicon glyphicon-cloud
✉	glyphicon glyphicon-envelope
✏	glyphicon glyphicon-pencil
🍸	glyphicon glyphicon-glass
♫	glyphicon glyphicon-music
🔍	glyphicon glyphicon-search
♥	glyphicon glyphicon-heart
★	glyphicon glyphicon-star
☆	glyphicon glyphicon-star-empty
👤	glyphicon glyphicon-user
🎞	glyphicon glyphicon-film
▦	glyphicon glyphicon-th-large
⠿	glyphicon glyphicon-th
☰	glyphicon glyphicon-th-list
✔	glyphicon glyphicon-ok
✖	glyphicon glyphicon-remove
🔍	glyphicon glyphicon-zoom-in
🔍	glyphicon glyphicon-zoom-out
⏻	glyphicon glyphicon-off
📶	glyphicon glyphicon-signal
⚙	glyphicon glyphicon-cog
🗑	glyphicon glyphicon-trash
🏠	glyphicon glyphicon-home
📄	glyphicon glyphicon-file
🕒	glyphicon glyphicon-time
🛣	glyphicon glyphicon-road
⤓	glyphicon glyphicon-download-alt
⬇	glyphicon glyphicon-download
⬆	glyphicon glyphicon-upload
📥	glyphicon glyphicon-inbox
▶	glyphicon glyphicon-play-circle
↻	glyphicon glyphicon-repeat
🔄	glyphicon glyphicon-refresh
📋	glyphicon glyphicon-list-alt
🔒	glyphicon glyphicon-lock
⚑	glyphicon glyphicon-flag
🎧	glyphicon glyphicon-headphones
🔇	glyphicon glyphicon-volume-off
🔉	glyphicon glyphicon-volume-down
🔊	glyphicon glyphicon-volume-up
▩	glyphicon glyphicon-qrcode
║	glyphicon glyphicon-barcode
🏷	glyphicon glyphicon-tag
🏷	glyphicon glyphicon-tags
📕	glyphicon glyphicon-book
🔖	glyphicon glyphicon-bookmark
🖨	glyphicon glyphicon-print
📷	glyphicon glyphicon-camera
A	glyphicon glyphicon-font
B	glyphicon glyphicon-bold
I	glyphicon glyphicon-italic
↕T	glyphicon glyphicon-text-height
T	glyphicon glyphicon-text-width
≡	glyphicon glyphicon-align-left
≡	glyphicon glyphicon-align-center
≡	glyphicon glyphicon-align-right
≡	glyphicon glyphicon-align-justify
☰	glyphicon glyphicon-list
⇤	glyphicon glyphicon-indent-left

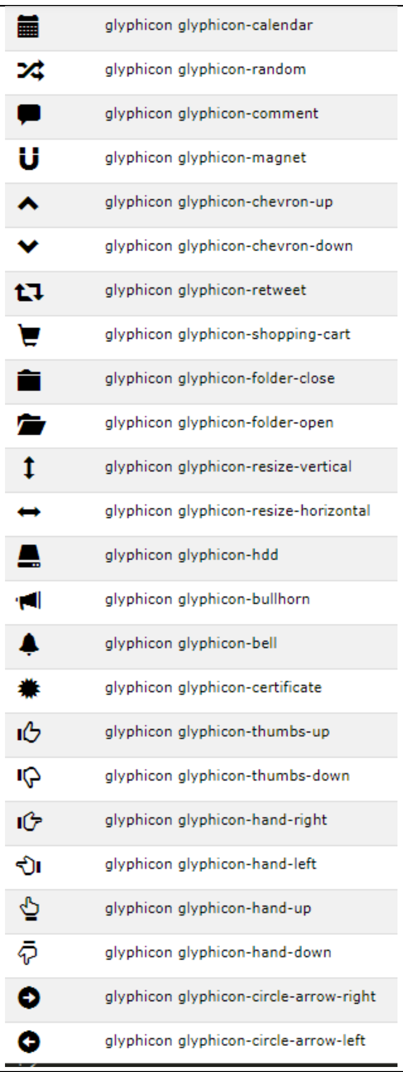


Menu

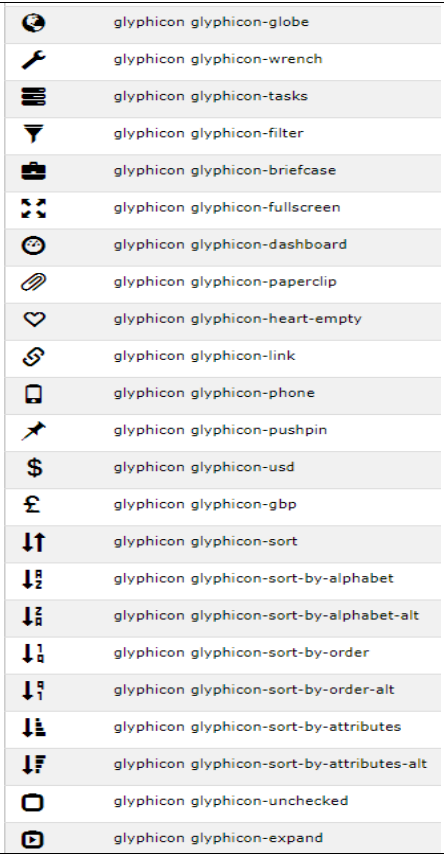
glyphicon glyphicon-indent-right
glyphicon glyphicon-facetime-video
glyphicon glyphicon-picture
glyphicon glyphicon-map-marker
glyphicon glyphicon-adjust
glyphicon glyphicon-tint
glyphicon glyphicon-edit
glyphicon glyphicon-share
glyphicon glyphicon-check
glyphicon glyphicon-move
glyphicon glyphicon-step-backward
glyphicon glyphicon-fast-backward
glyphicon glyphicon-backward
glyphicon glyphicon-play
glyphicon glyphicon-pause
glyphicon glyphicon-stop
glyphicon glyphicon-forward
glyphicon glyphicon-fast-forward
glyphicon glyphicon-step-forward
glyphicon glyphicon-eject
glyphicon glyphicon-chevron-left
glyphicon glyphicon-chevron-right
glyphicon glyphicon-plus-sign



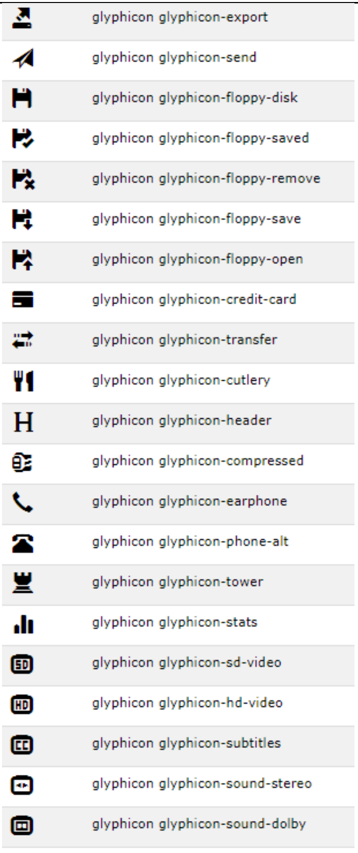
glyphicon glyphicon-minus-sign
glyphicon glyphicon-remove-sign
glyphicon glyphicon-ok-sign
glyphicon glyphicon-question-sign
glyphicon glyphicon-info-sign
glyphicon glyphicon-screenshot
glyphicon glyphicon-remove-circle
glyphicon glyphicon-ok-circle
glyphicon glyphicon-ban-circle
glyphicon glyphicon-arrow-left
glyphicon glyphicon-arrow-right
glyphicon glyphicon-arrow-up
glyphicon glyphicon-arrow-down
glyphicon glyphicon-share-alt
glyphicon glyphicon-resize-full
glyphicon glyphicon-resize-small
glyphicon glyphicon-exclamation-sign
glyphicon glyphicon-gift
glyphicon glyphicon-leaf
glyphicon glyphicon-fire
glyphicon glyphicon-eye-open
glyphicon glyphicon-eye-close
glyphicon glyphicon-warning-sign
glyphicon glyphicon-plane



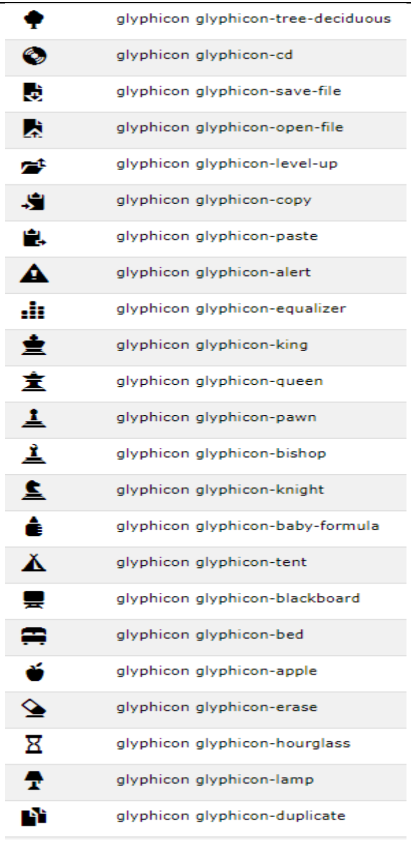
glyphicon glyphicon-calendar
glyphicon glyphicon-random
glyphicon glyphicon-comment
glyphicon glyphicon-magnet
glyphicon glyphicon-chevron-up
glyphicon glyphicon-chevron-down
glyphicon glyphicon-retweet
glyphicon glyphicon-shopping-cart
glyphicon glyphicon-folder-close
glyphicon glyphicon-folder-open
glyphicon glyphicon-resize-vertical
glyphicon glyphicon-resize-horizontal
glyphicon glyphicon-hdd
glyphicon glyphicon-bullhorn
glyphicon glyphicon-bell
glyphicon glyphicon-certificate
glyphicon glyphicon-thumbs-up
glyphicon glyphicon-thumbs-down
glyphicon glyphicon-hand-right
glyphicon glyphicon-hand-left
glyphicon glyphicon-hand-up
glyphicon glyphicon-hand-down
glyphicon glyphicon-circle-arrow-right
glyphicon glyphicon-circle-arrow-left



glyphicon glyphicon-globe
glyphicon glyphicon-wrench
glyphicon glyphicon-tasks
glyphicon glyphicon-filter
glyphicon glyphicon-briefcase
glyphicon glyphicon-fullscreen
glyphicon glyphicon-dashboard
glyphicon glyphicon-paperclip
glyphicon glyphicon-heart-empty
glyphicon glyphicon-link
glyphicon glyphicon-phone
glyphicon glyphicon-pushpin
glyphicon glyphicon-usd
glyphicon glyphicon-gbp
glyphicon glyphicon-sort
glyphicon glyphicon-sort-by-alphabet
glyphicon glyphicon-sort-by-alphabet-alt
glyphicon glyphicon-sort-by-order
glyphicon glyphicon-sort-by-order-alt
glyphicon glyphicon-sort-by-attributes
glyphicon glyphicon-sort-by-attributes-alt
glyphicon glyphicon-unchecked
glyphicon glyphicon-expand

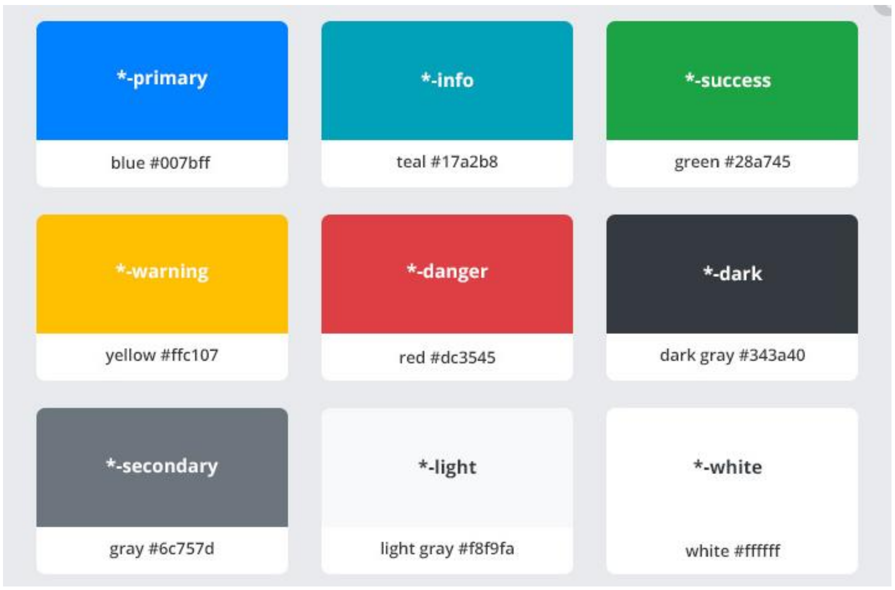


glyphicon glyphicon-export
glyphicon glyphicon-send
glyphicon glyphicon-floppy-disk
glyphicon glyphicon-floppy-saved
glyphicon glyphicon-floppy-remove
glyphicon glyphicon-floppy-save
glyphicon glyphicon-floppy-open
glyphicon glyphicon-credit-card
glyphicon glyphicon-transfer
glyphicon glyphicon-cutlery
glyphicon glyphicon-header
glyphicon glyphicon-compressed
glyphicon glyphicon-earphone
glyphicon glyphicon-phone-alt
glyphicon glyphicon-tower
glyphicon glyphicon-stats
glyphicon glyphicon-sd-video
glyphicon glyphicon-hd-video
glyphicon glyphicon-subtitles
glyphicon glyphicon-sound-stereo
glyphicon glyphicon-sound-dolby

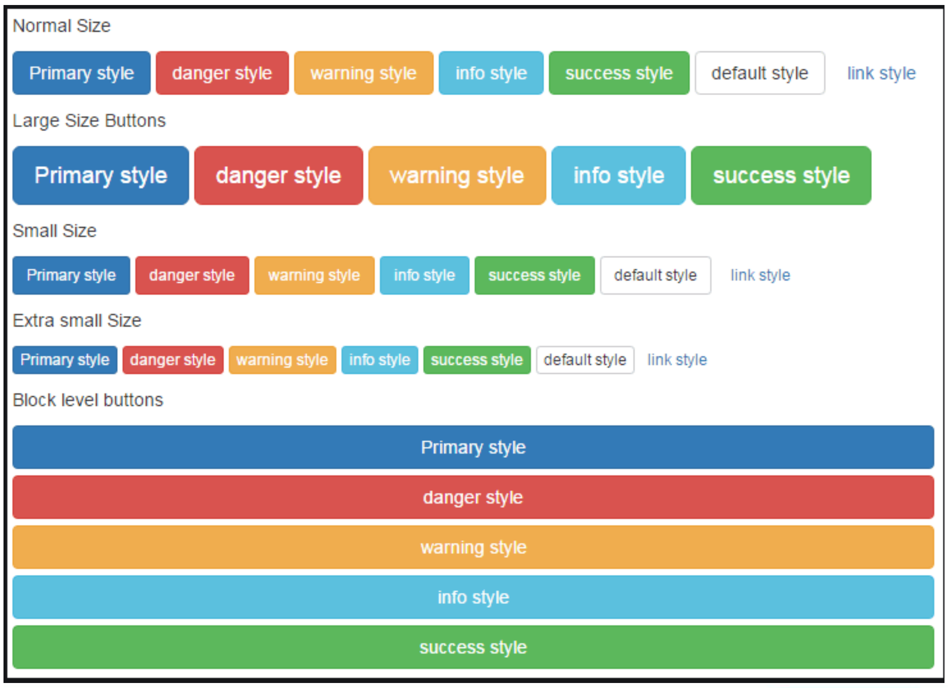


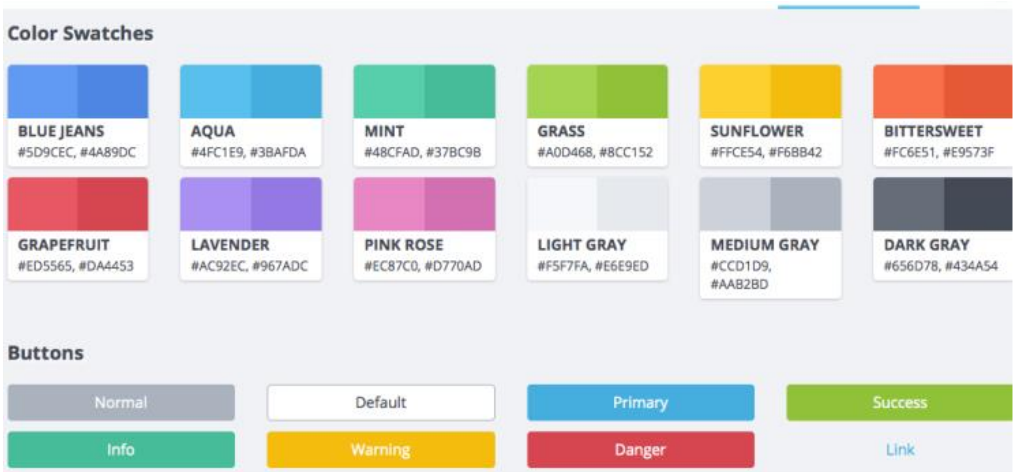
glyphicon glyphicon-tree-deciduous
glyphicon glyphicon-cd
glyphicon glyphicon-save-file
glyphicon glyphicon-open-file
glyphicon glyphicon-level-up
glyphicon glyphicon-copy
glyphicon glyphicon-paste
glyphicon glyphicon-alert
glyphicon glyphicon-equalizer
glyphicon glyphicon-king
glyphicon glyphicon-queen
glyphicon glyphicon-pawn
glyphicon glyphicon-bishop
glyphicon glyphicon-knight
glyphicon glyphicon-baby-formula
glyphicon glyphicon-tent
glyphicon glyphicon-blackboard
glyphicon glyphicon-bed
glyphicon glyphicon-apple
glyphicon glyphicon-erase
glyphicon glyphicon-hourglass
glyphicon glyphicon-lamp
glyphicon glyphicon-duplicate

Color



*-primary	*-info	*-success
blue #007bff	teal #17a2b8	green #28a745
*-warning	*-danger	*-dark
yellow #ffc107	red #dc3545	dark gray #343a40
*-secondary	*-light	*-white
gray #6c757d	light gray #f8f9fa	white #ffffff





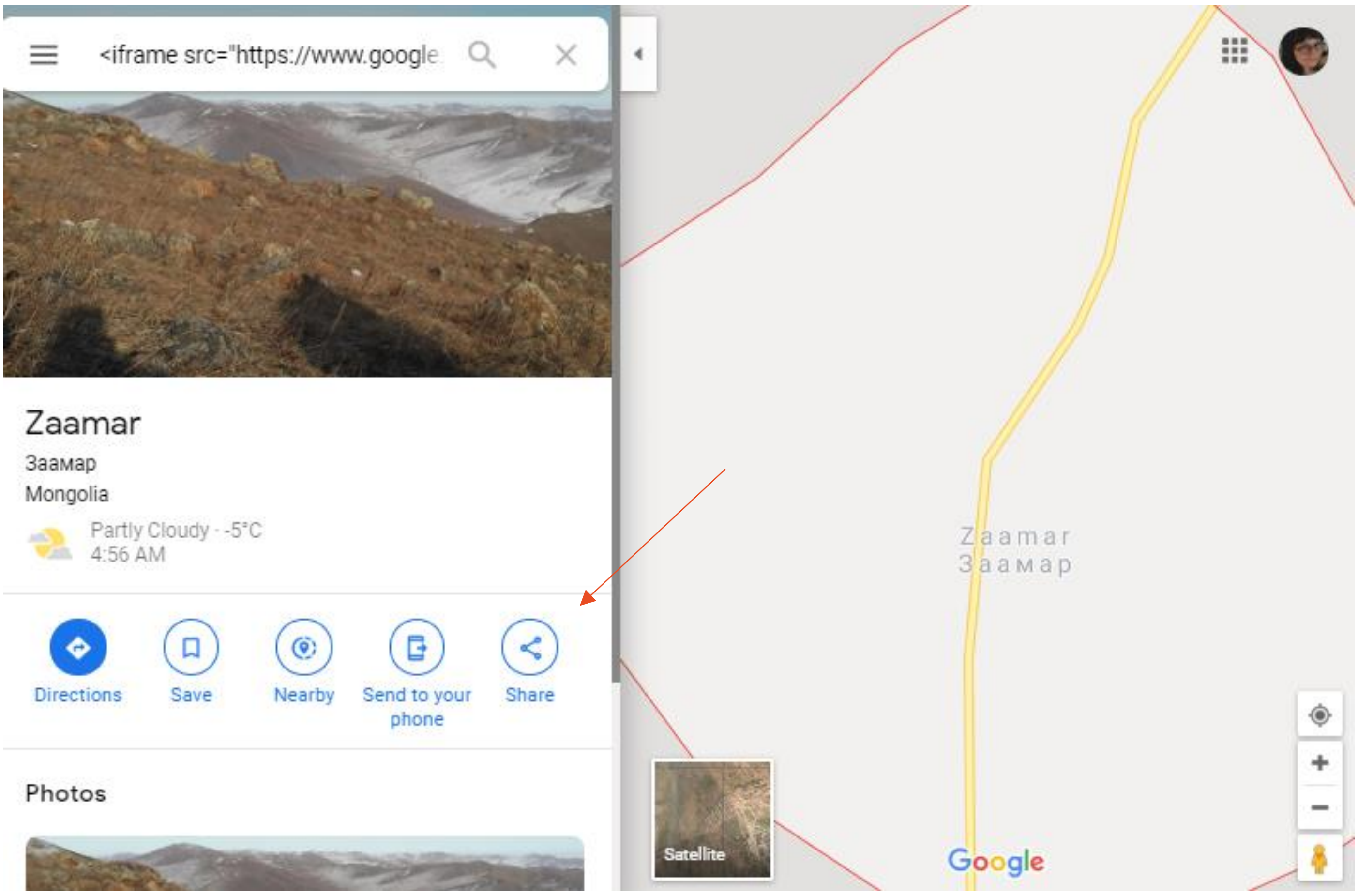
Color Swatches

BLUE JEANS #5D9CEC, #4A89DC	AQUA #4FC1E9, #3BAFDA	MINT #48CFAD, #37BC9B	GRASS #A0D468, #8CC152	SUNFLOWER #FFCE54, #F6BB42	BITTERSWEET #FC6E51, #E9573F
GRAPEFRUIT #ED5565, #DA4453	LAVENDER #AC92EC, #967ADC	PINK ROSE #EC87C0, #D770AD	LIGHT GRAY #F5F7FA, #E6E9ED	MEDIUM GRAY #CCD1D9, #AAB2BD	DARK GRAY #656D78, #434A54

Google Map

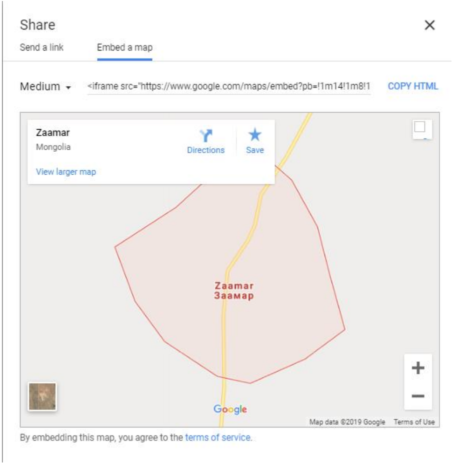
Embed a map

To get a free access to a google map you can embed a map in your site. The method is very simple. Go to Google map and find the site you wish to show in a map.



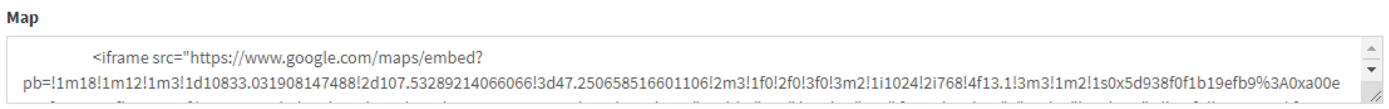
Press share and choose embed a map.

Copy the html part in the map field of the apiary and save.



<iframe src="https://www.google.com/maps/embed?pb=!1m14!1m8!1

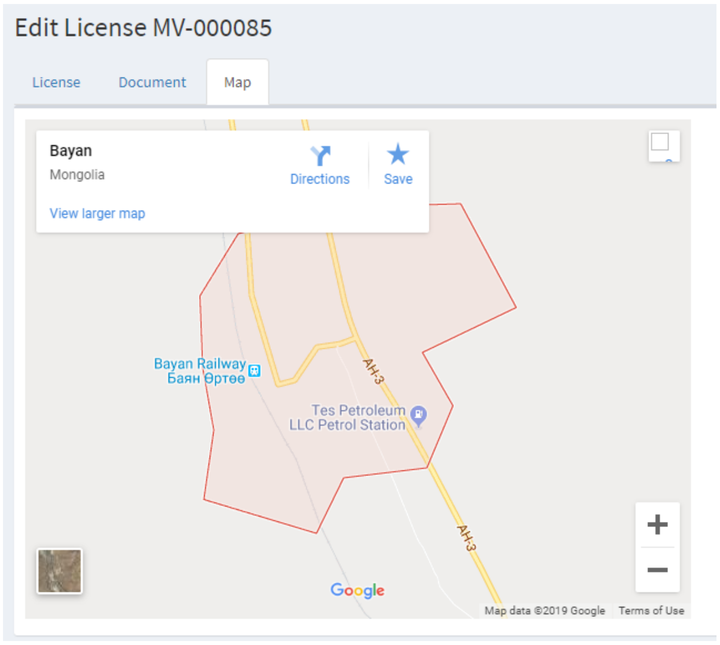
By embedding this map, you agree to the terms of service.



Map

```
<iframe src="https://www.google.com/maps/embed?
pb=!1m18!1m12!1m3!1d10833.031908147488!2d107.5328921406606!3d47.25065851660110!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x5d938f0f1b19efb9%3A0xa00e
```

In Apiary part, the tab MAP will present the google map of the site.



Google Map API with key

To use google map to show a site, you need to use an API for google map.

If you don't have an API key, you will get the error message: this page didn't load google maps correctly. It means you are not using an API key and will need to sign up for one then configure your site to use it. Websites that started using Google Maps on or after June 22, 2016 require an API key in order for maps to show.

Follow these steps to create and implement a Google Maps API Key. Google gives you a very large amount of free credits every month which makes their maps service virtually free. To date, none of our customers have ever reported needing to pay anything. Even so, you can set limits and alerts.

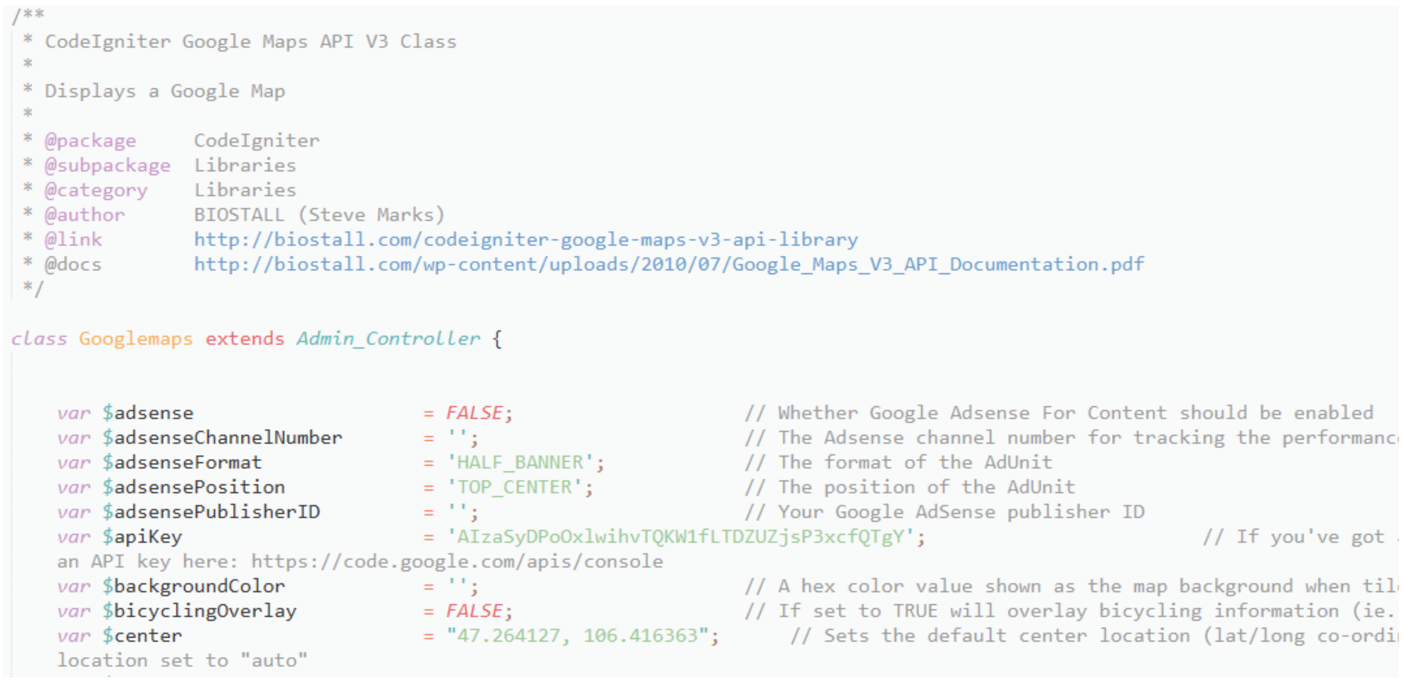
1. Go to the Google Maps Platform welcome page then click Get Started.
2. Choose Maps, Routes and Places (all three) then click Continue.
3. Log into your Google Account or create a new one, if necessary.
   You may need to repeat the previous steps after logging in.
4. Choose "Create a new project", enter a name, then click Next.
5. Set up billing for your new project then proceed to enable your APIs.
   Google gives you such a large amount of free credits **every month that to date none of our customers have reported needing to pay.**
6. Click the API Console link to restrict your key's use to your website only (important).
7. Under Application restrictions, choose "HTTP referrers (web sites)" then add the two entries below (replacing yourname.com with your own domain). Type the first entry then hit enter on your keyboard to add it. Repeat to add the second entry. Having both entries (with asterisks) will help ensure your maps work on any URL of your website. yourname.com/* *.yourname.com/*
8. Click Save then copy your key that is now shown on the screen.

Work in Codeigniter

For the call in codeigniter, you need first to copy the googlemap api v3. This seems to be a project that have been abandonned by the community but I was able to correct some errors. To get the map you should insert the API key inside the form.

Application/library/Googlemaps.php and Jsmin.php must be copied first. I had to correct the definition of the Class that needs to be extended now. This version must be for an old Codeigniter version. I also add the API key I created on my google account. The var center should be the province office. It will be the center location for this API.

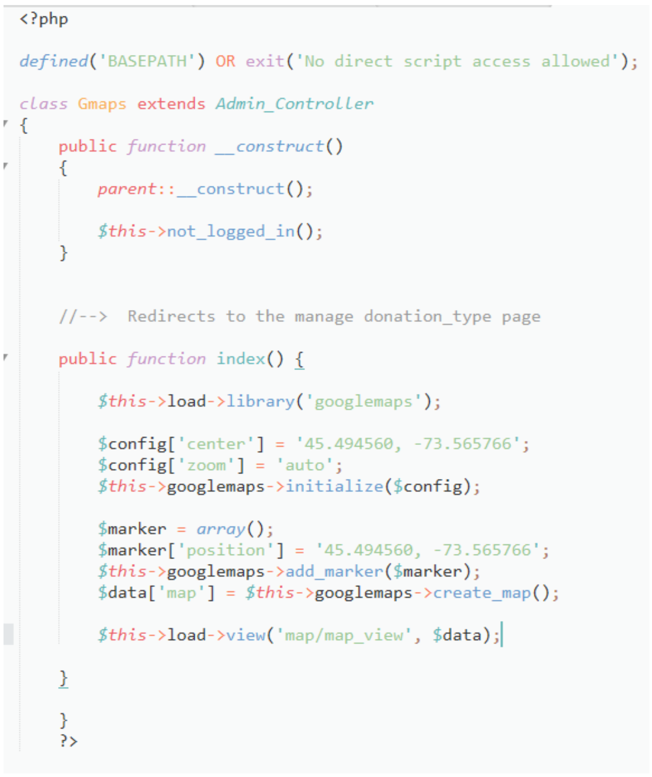
The site biostall.com have all the information about this method.



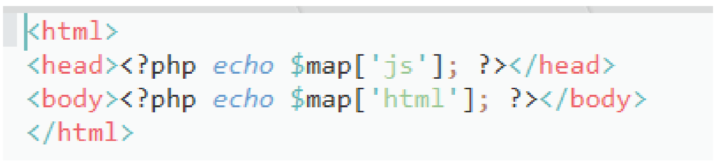
```
/**
 * CodeIgniter Google Maps API V3 Class
 *
 * Displays a Google Map
 *
 * @package     CodeIgniter
 * @subpackage  Libraries
 * @category    Libraries
 * @author      BIOSTALL (Steve Marks)
 * @link        http://biostall.com/codeigniter-google-maps-v3-api-library
 * @docs        http://biostall.com/wp-content/uploads/2010/07/Google_Maps_V3_API_Documentation.pdf
 */

class Googlemaps extends Admin_Controller {


    var $adsense                  = FALSE;                    // Whether Google Adsense For Content should be enabled
    var $adsenseChannelNumber     = '';                       // The Adsense channel number for tracking the performanc
    var $adsenseFormat            = 'HALF_BANNER';            // The format of the AdUnit
    var $adsensePosition          = 'TOP_CENTER';             // The position of the AdUnit
    var $adsensePublisherID       = '';                       // Your Google AdSense publisher ID
    var $apiKey                   = 'AIzaSyDPoOxlwihvTQKW1fLTDZUZjsP3xcfQTgY';                    // If you've got
    an API key here: https://code.google.com/apis/console
    var $backgroundColor          = '';                       // A hex color value shown as the map background when til
    var $bicyclingOverlay         = FALSE;                    // If set to TRUE will overlay bicycling information (ie.
    var $center                   = "47.264127, 106.416363";     // Sets the default center location (lat/long co-ordi
    location set to "auto"
```

The controller Gmaps.php and view map_view.php are a basic for more work … You can see the result with

Localhost:81/xxxxxx/index.php/gmaps



```
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Gmaps extends Admin_Controller
{
    public function __construct()
    {
        parent::__construct();

        $this->not_logged_in();
    }


    //-->  Redirects to the manage donation_type page

    public function index() {

        $this->load->library('googlemaps');

        $config['center'] = '45.494560, -73.565766';
        $config['zoom'] = 'auto';
        $this->googlemaps->initialize($config);

        $marker = array();
        $marker['position'] = '45.494560, -73.565766';
        $this->googlemaps->add_marker($marker);
        $data['map'] = $this->googlemaps->create_map();

        $this->load->view('map/map_view', $data);

    }

    }
    ?>
```



```
<html>
<head><?php echo $map['js']; ?></head>
<body><?php echo $map['html']; ?></body>
</html>
```

Some tricks

Remove trailing spaces in the code

For the trailing whitespaces, I installed an extension for removing whitespaces. You can check this extension: https://marketplace.visualstudio.com/items?itemName=shardulm94.trailing-spaces
It automatically removes whitespaces on your code.

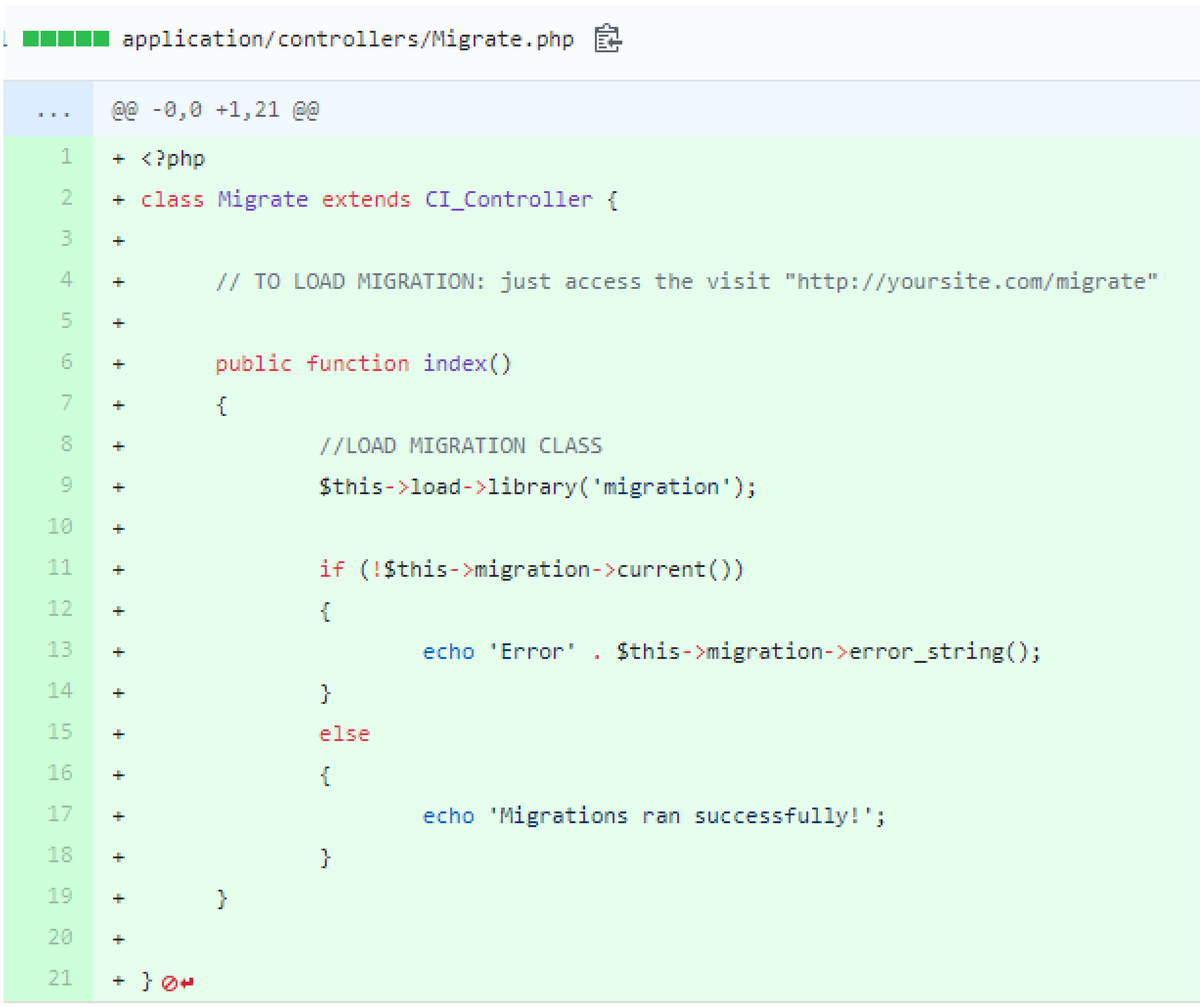
Migration process

We can use Migration facilities in codeigniter to upgrade the database and keep the versioning.

First in /application/config/migration.php

```
$config['migration_enabled'] = TRUE;

$config['migration_type'] = 'sequential';

$config['migration_version'] = 1;
```

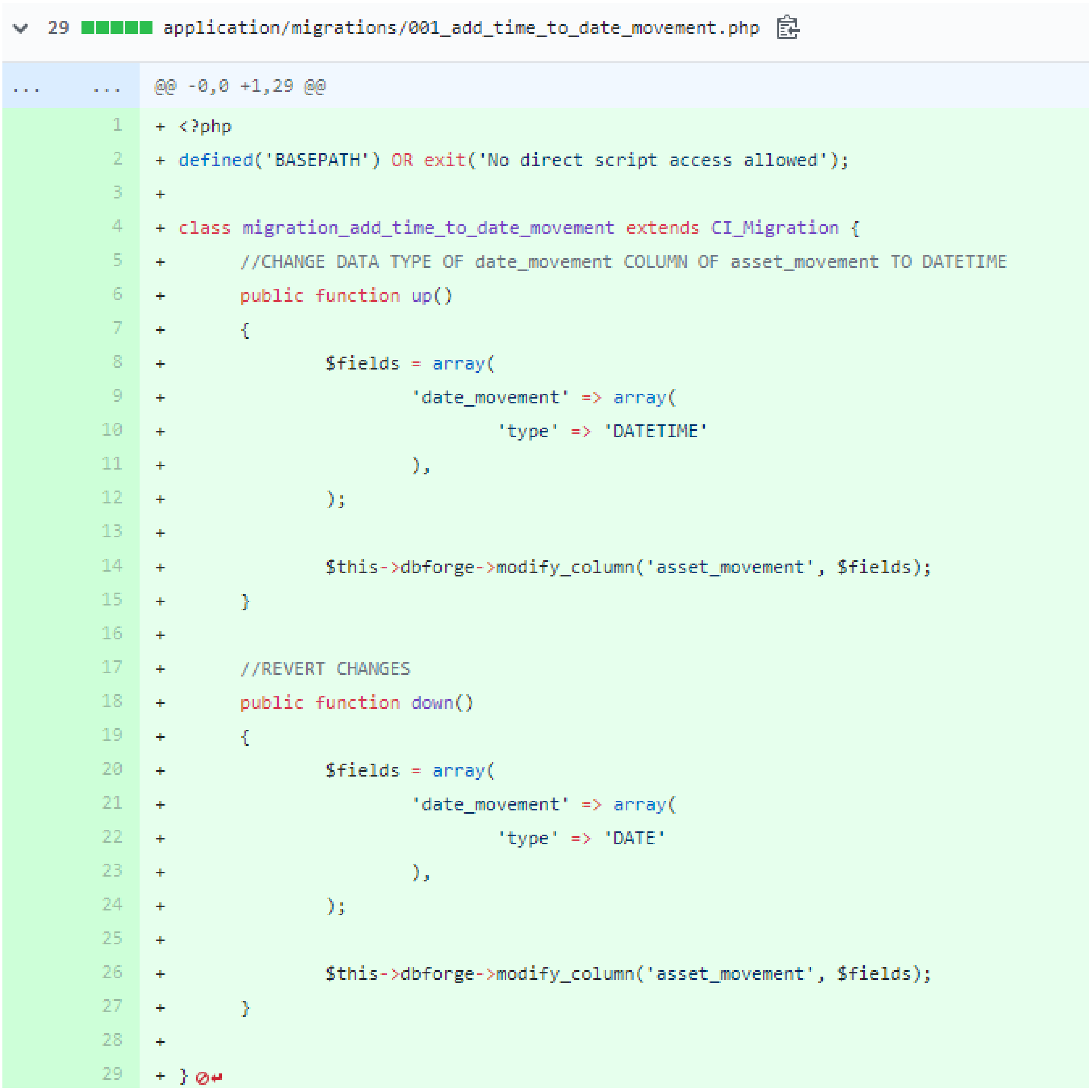
Create a Migration class



```
application/controllers/Migrate.php
@@ -0,0 +1,21 @@
+ <?php
+ class Migrate extends CI_Controller {
+
+       // TO LOAD MIGRATION: just access the visit "http://yoursite.com/migrate"
+
+       public function index()
+       {
+               //LOAD MIGRATION CLASS
+               $this->load->library('migration');
+
+               if (!$this->migration->current())
+               {
+                       echo 'Error' . $this->migration->error_string();
+               }
+               else
+               {
+                       echo 'Migrations ran successfully!';
+               }
+       }
+
+ }
```

This will serve to ensure that the migration is done properly.

Create the module that will update the database. In this example, we change the definition of a field from DATE to DATETIME.



```
29 application/migrations/001_add_time_to_date_movement.php
@@ -0,0 +1,29 @@
+ <?php
+ defined('BASEPATH') OR exit('No direct script access allowed');
+
+ class migration_add_time_to_date_movement extends CI_Migration {
+       //CHANGE DATA TYPE OF date_movement COLUMN OF asset_movement TO DATETIME
+       public function up()
+       {
+               $fields = array(
+                       'date_movement' => array(
+                               'type' => 'DATETIME'
+                       ),
+               );
+
+               $this->dbforge->modify_column('asset_movement', $fields);
+       }
+
+       //REVERT CHANGES
+       public function down()
+       {
+               $fields = array(
+                       'date_movement' => array(
+                               'type' => 'DATE'
+                       ),
+               );
+
+               $this->dbforge->modify_column('asset_movement', $fields);
+       }
+
+ }
```

For the migration, to apply the migration just go to the http://{base_url}/migrate. CodeIgniter applies the current migration version on the migrations table on the database. To rollback, just set the version on the migrations table. In our database, I used incremental migration versioning.
https://codeigniter.com/user_guide/libraries/migration.html

Tab control

So we have this GET parameters that can be seen on the url and can be used by PHP.

I returned this line in the ajax callback on submit function.

window.location.href = "<?php echo base_url('asset/update/'.$asset_data['asset']['id']) ?>" + "?tab=movement";

…

and I added a GET parameter tab with a value of movement.

So in CodeIgniter we have this Input Class that gets the value of these GET parameters:

In __construct() function of the Asset Controller, I added this line:

```
public function __construct()
{
parent::__construct();
$this->not_logged_in();
$this->data['page_title'] = 'Asset';
$this->data['active_tab'] = $this->input->get('tab') ?? 'asset';

}
```

?? is called a **Null Coalescing Operator** (It is a new feature in PHP7) that functions like **isset** method. The Null coalescing operator returns its first operand if it exists and is not NULL; otherwise it returns its second operand.

https://www.tutorialspoint.com/php7/php7_coalescing_operator.htm

Next, I changed the tabs:

```
<section class="content">
      <ul class="nav nav-tabs">
        <li class="<?php echo (($active_tab === 'asset') ? 'active' : '')
?>"><a data-toggle="tab" href="#asset">Asset</a></li>
        <li class="<?php echo (($active_tab === 'movement') ? 'active' : '')
?>"><a data-toggle="tab" href="#movement">Movement</a></li>
        <li class="<?php echo (($active_tab === 'activity') ? 'active' : '')
?>"><a data-toggle="tab" href="#activity">Activity</a></li>
        <li class="<?php echo (($active_tab === 'document') ? 'active' : '')
?>"><a data-toggle="tab" href="#document">Document</a></li>
      </ul>

...
```

And the tab panes:

```
<div id="asset" class="tab-pane fade <?php echo (($active_tab === 'asset') ?
'in active' : '') ?>">

...

<div id="movement" class="tab-pane fade <?php echo (($active_tab ===
'movement') ? 'in active' : '') ?>"">

...
```

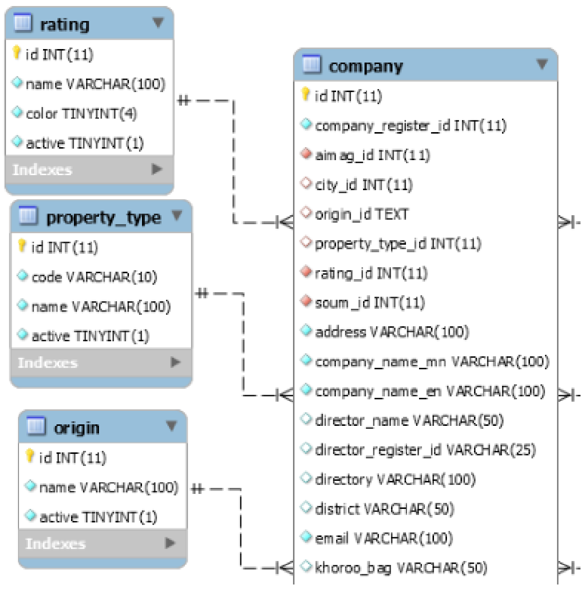
```
<div id="activity" class="tab-pane fade <?php echo (($active_tab ===
'activity') ? 'in active' : '') ?>"">
...

<div id="document" class="tab-pane fade <?php echo (($active_tab ===
'document') ? 'in active' : '') ?>"">
...
```

So here I used Shorthand If-Else. https://dzone.com/articles/php-shorthand-if-else-examples

So now we can return to tabs where we came from. You can add `+ "?tab=activity"` on the activity part, etc.

Mutiple choice for a drop-down list (json)

Here is a method to avoid the creation of sub-form to keep more than one choice for a drop-down list. In this example, it is possible to have more than one origin for the beekeeper. You can see the difference between the foreign key for RATING, which have only one data, so the key will have the same format of the field rating.id int(11).



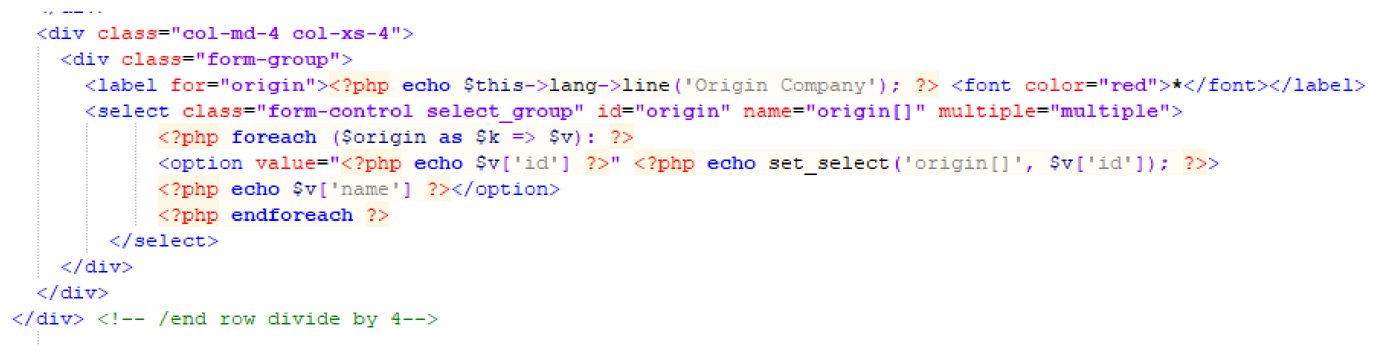
For keeping more than one information in a field, we need to create a chain. In the case of Origin, the foreing key will be with format TEXT.

Every origin presented in the field origin.id will looks like this:

["1","11"]

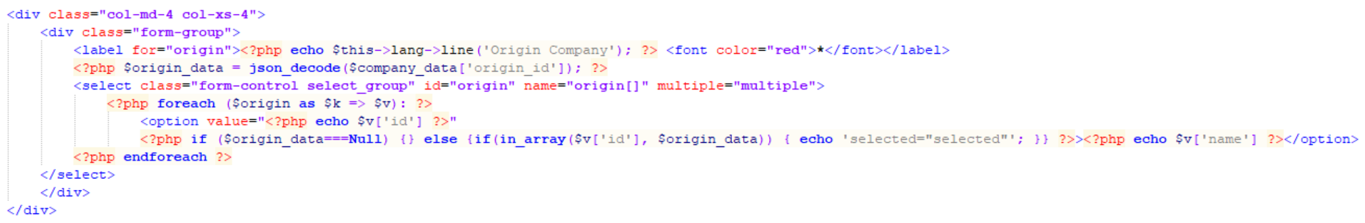
The method to create the chain and extract the information will be the following:

For the **view** create, the name will have [] and a criteria multiple.



```
  <div class="col-md-4 col-xs-4">
    <div class="form-group">
      <label for="origin"><?php echo $this->lang->line('Origin Company'); ?> <font color="red">*</font></label>
      <select class="form-control select_group" id="origin" name="origin[]" multiple="multiple">
            <?php foreach ($origin as $k => $v): ?>
            <option value="<?php echo $v['id'] ?>" <?php echo set_select('origin[]', $v['id']); ?>>
            <?php echo $v['name'] ?></option>
            <?php endforeach ?>
        </select>
    </div>
  </div>
</div> <!-- /end row divide by 4-->
```

For the update, same treatment except that we need to fill the data from the database. For this purpose, we will need to json_decode the information:



```
<div class="col-md-4 col-xs-4">
    <div class="form-group">
        <label for="origin"><?php echo $this->lang->line('Origin Company'); ?> <font color="red">*</font></label>
        <?php $origin_data = json_decode($company_data['origin_id']); ?>
        <select class="form-control select_group" id="origin" name="origin[]" multiple="multiple">
            <?php foreach ($origin as $k => $v): ?>
                <option value="<?php echo $v['id'] ?>"
                <?php if ($origin_data===Null) {} else {if(in_array($v['id'], $origin_data)) { echo 'selected="selected"'; }} ?>><?php echo $v['name'] ?></option>
        <?php endforeach ?>
    </select>
    </div>
</div>
```

In the **controller,** you need to encode the data before inserting or updating in the database.

'origin_id' => json_encode($this->input->post('origin')),

The select and where clause will be different:

Here we have a field where more than one District is possible:

$district_to_find = '"'.$district.'"';

$sql = "SELECT apiary   FROM apiary   WHERE apiary.district_id LIKE '%$district_to_find%'

In Object Oriented method:

```
// select with the wildcard %.  It is possible to have more
// than one district in apiary table.   In this case, the information
// will appear between bracket ex:["1"].  The search will be
// SELECT * FROM apiary WHERE district_id LIKE '%["1"]%'
$this->db->select('*');
$this->db->from('apiary');
$this->db->like('province_id', $id, 'both');
$query = $this->db->get();
$num_rows = $query->num_rows();
```

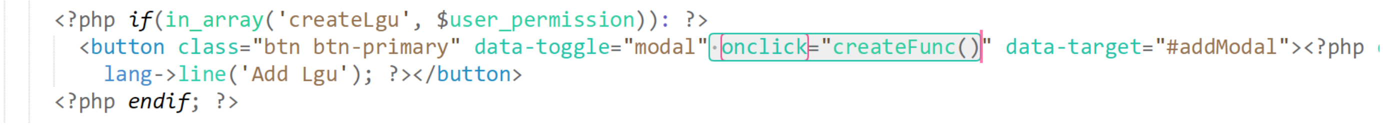
For printing the list of district in a single field, we need to decode and create a chain with all the names

```
foreach ($REP02 as $rs):
        $district_apiary = json_decode($rs->district_id);
        $district_to_print = '';
        // Get the content of each district for the apiary
        if (!$district_apiary == null) {
        foreach($district_apiary as $key=>$value){
                $district_data = $this->model_district->getDistrictData($district_apiary[$key]);
               $district_to_print = $district_to_print.' '.$district_data['name'];}
}
```

Function for clearing the form with ajax

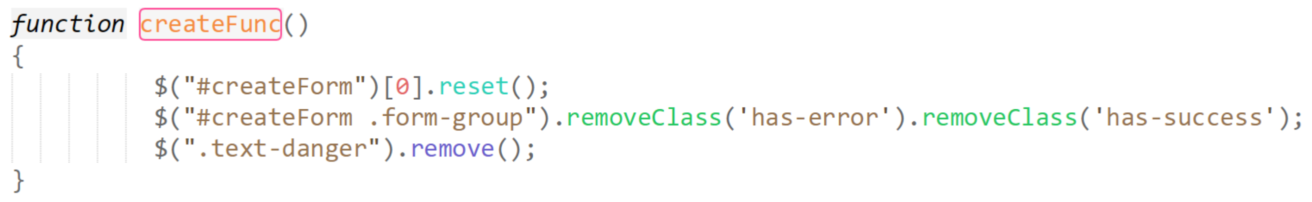
Here is the method for clearing the form after an error have been detected.

In the create button, add the onclick function. This will make sure that the form is cleared in any situation; if there is an error or not.



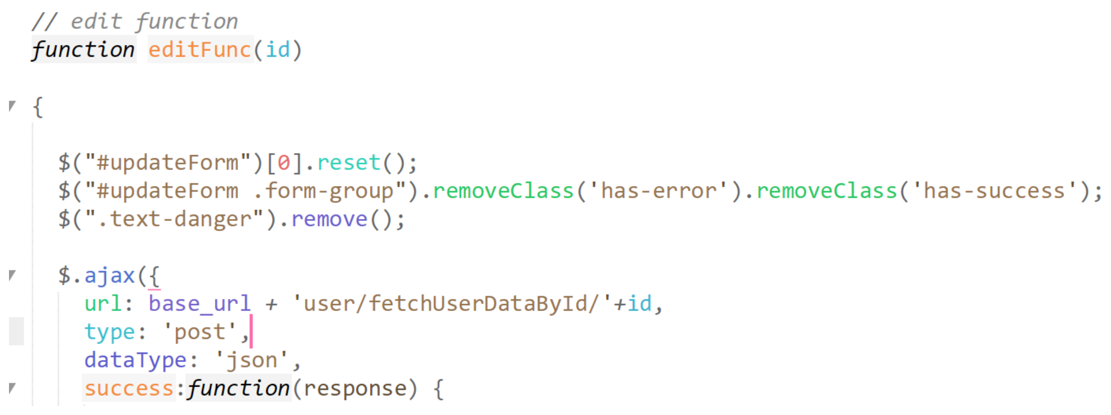
```
<?php if(in_array('createLgu', $user_permission)): ?>
  <button class="btn btn-primary" data-toggle="modal" onclick="createFunc()" data-target="#addModal"><?php
    lang->line('Add Lgu'); ?></button>
<?php endif; ?>
```

The function should be inserted in Javascript, (just before edit function)



```
function createFunc()
{
        $("#createForm")[0].reset();
        $("#createForm .form-group").removeClass('has-error').removeClass('has-success');
        $(".text-danger").remove();
}
```

The same lines should be inserted in the edit function



```
// edit function
function editFunc(id)

{

    $("#updateForm")[0].reset();
    $("#updateForm .form-group").removeClass('has-error').removeClass('has-success');
    $(".text-danger").remove();

    $.ajax({
      url: base_url + 'user/fetchUserDataById/'+id,
      type: 'post',
      dataType: 'json',
      success:function(response) {
```

Bootstrap Tooltip

Tooltips are not CSS-only plugins, and must therefore be initialized with jQuery: select the specified element and call the tooltip() method.

You must fist install tooltip in the javascript <script>

```
<script>
$(document).ready(function(){
  $('[data-toggle="tooltip"]').tooltip();
});
</script>
```

You can specify here the tooltip for a specific field. In this example, it will work for every field.

You can use a clickable link for the tooltip, if it’s interesting to go to a specific page.

<a href="#" data-toggle="tooltip" title="Hooray!">Hover over me</a>

Or use it under an image as a way to help the user understand what’s inside the form.

<img width="25" height="25" data-toggle="tooltip" data-placement="bottom" title="Some information about the client."

src="<?php echo base_url('assets/images/question.png'); ?>" />

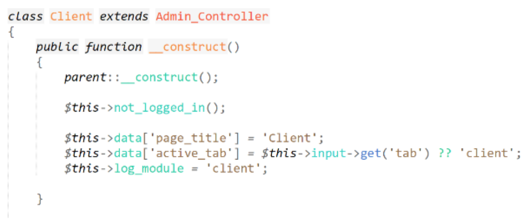
Installation of a LOG

First you need to create a log table. Here is the example:

```
CREATE TABLE IF NOT EXISTS `log` (
  `id` int(9) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `timestamp` datetime NOT NULL,
  `module` varchar(100) NOT NULL,
  `action` varchar(100) NOT NULL,
  `subject_id` int(11) NOT NULL,
  `attributes` longtext NOT NULL,
  PRIMARY KEY (`id`)
```

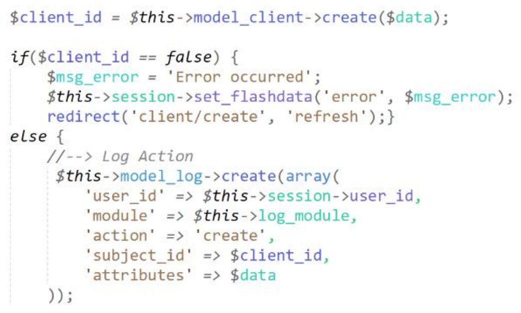
The table log will be filled inside the system, where it is needed to keep a trace.
It will be mainly used in the controller of the table we want to keep the log.

First indicate the table in the construction of the controller. It will be used to create an entry in the log.



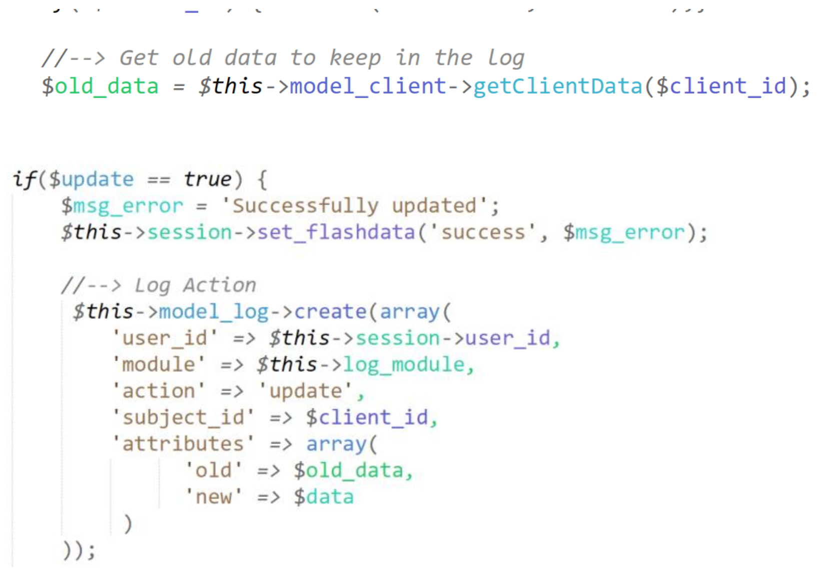
```
class Client extends Admin_Controller
{
    public function __construct()
    {
        parent::__construct();

        $this->not_logged_in();

        $this->data['page_title'] = 'Client';
        $this->data['active_tab'] = $this->input->get('tab') ?? 'client';
        $this->log_module = 'client';

    }
```

For CREATE, you will insert in the log after the succesful insert in the table client. In the attributes, copy the whole set of data that were used to insert in the table.



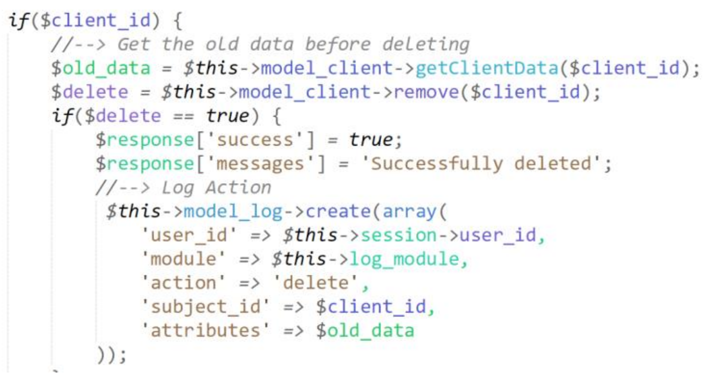
```
$client_id = $this->model_client->create($data);

if($client_id == false) {
    $msg_error = 'Error occurred';
    $this->session->set_flashdata('error', $msg_error);
    redirect('client/create', 'refresh');}
else {
    //--> Log Action
     $this->model_log->create(array(
        'user_id' => $this->session->user_id,
        'module' => $this->log_module,
        'action' => 'create',
        'subject_id' => $client_id,
        'attributes' => $data
    ));
```

For UPDATE, you keep the old-data and the new-data so that you can eventually retrace the information.



```
    //--> Get old data to keep in the log
    $old_data = $this->model_client->getClientData($client_id);


if($update == true) {
    $msg_error = 'Successfully updated';
    $this->session->set_flashdata('success', $msg_error);

    //--> Log Action
     $this->model_log->create(array(
        'user_id' => $this->session->user_id,
        'module' => $this->log_module,
        'action' => 'update',
        'subject_id' => $client_id,
        'attributes' => array(
            'old' => $old_data,
            'new' => $data
        )
    ));
```

For DELETE, keep also the old data set.



```
if($client_id) {
    //--> Get the old data before deleting
    $old_data = $this->model_client->getClientData($client_id);
    $delete = $this->model_client->remove($client_id);
    if($delete == true) {
        $response['success'] = true;
        $response['messages'] = 'Successfully deleted';
        //--> Log Action
         $this->model_log->create(array(
            'user_id' => $this->session->user_id,
            'module' => $this->log_module,
            'action' => 'delete',
            'subject_id' => $client_id,
            'attributes' => $old_data
        ));
```

The LOG can be installed in any tables you wish to keep a log.