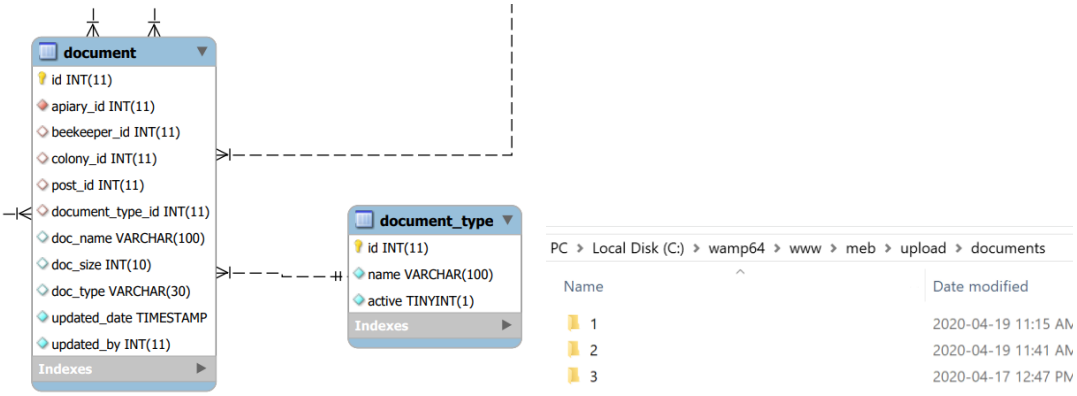
Upload of documents

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**



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

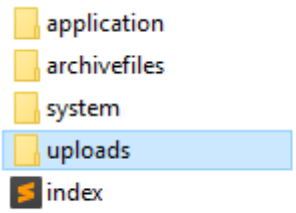
When we delete a document we must take care of deleting the entry in the table document.

```
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>
```