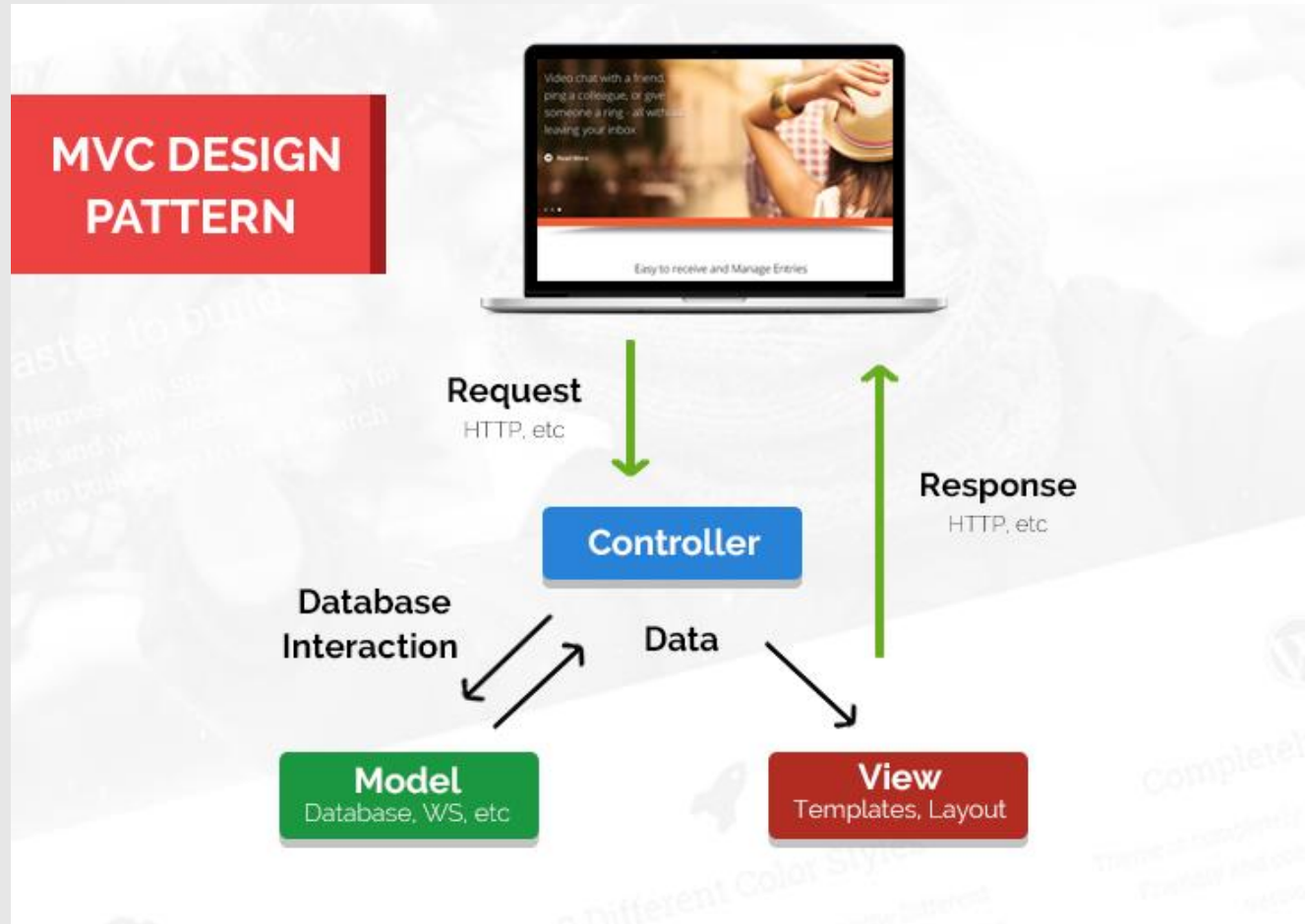
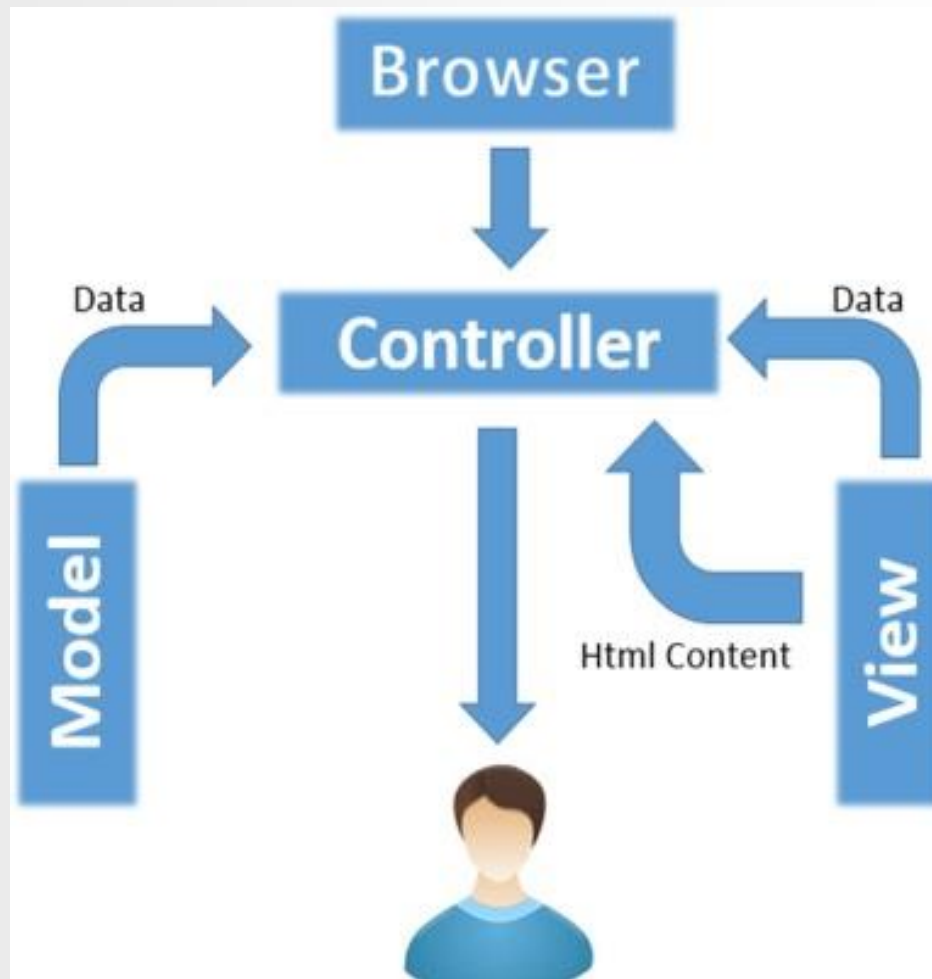


# Framework Codeigniter phần 1

# Mô hình MVC

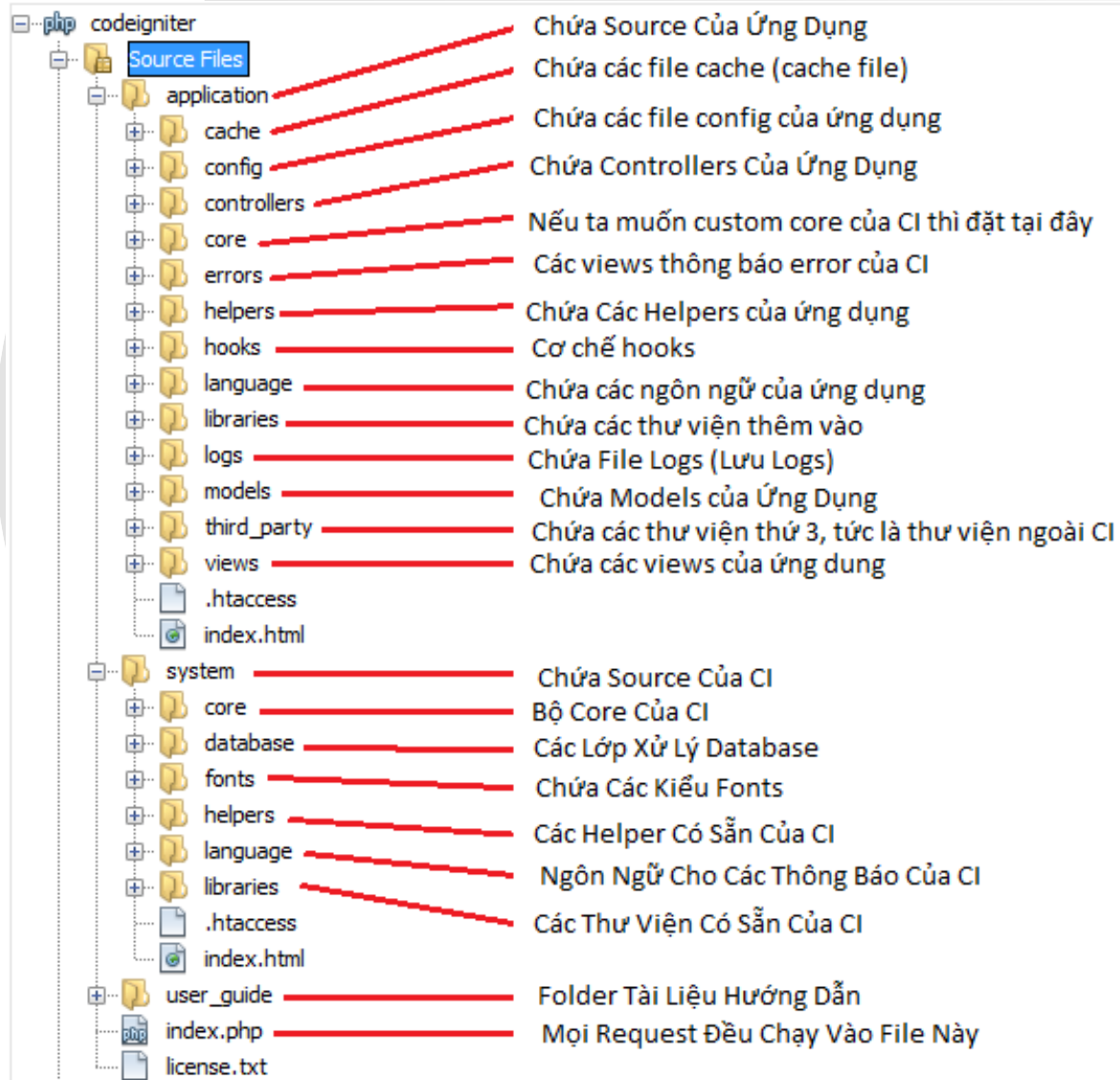


# Mô hình MVC



- ❖ Frameworks đơn giản và dễ học nhất mà ai cũng biết đó là Codeigniter, đây là một framework được xây dựng từ mô hình MVC có ưu điểm là chạy nhanh, dễ học cho người mới tìm hiểu MVC Framework. Codeigniter Framework còn viết tắt là CI Framework.
- ❖ Download Và Cài Đặt
  - Trước tiên bạn vào <https://github.com/bcit-ci/CodeIgniter/archive/3.1.6.zip> để download source (chọn version 3.x).
  - Sau khi download xong bạn sẽ có được 1 file nén .zip, bạn vào folder **WWW/htdocs** của Web Server tạo một folder mới tên là Codeigniter sau đó copy file vừa download vào trong folder mới tạo này và giải nén tại đó.
  - Sau khi giải nén các bạn ra trình duyệt gõ localhost/codeigniter, nếu có kết quả xuất hiện thì bạn đã cài đặt thành công, ngược lại bạn cài đặt bị lỗi.

# Cấu Trúc Folder Codeigniter



- Trong đó folder System là bộ core của CI, chúng ta không được đụng tới, chỉ được phép gọi ra và xài.
- Folder user\_guide chỉ là folder document, các bạn xóa nó đi vì không cần thiết cho ứng dụng, nếu các bạn muốn để lại tham khảo thì cũng không sao.
- Folder Application là folder chứa source web trong quá trình mình phát triển. Mọi file đều nằm trong folder này và tùy vào loại file mà lưu những vị trí khác nhau. Ở những bài tới tôi sẽ hướng dẫn các bạn sau.

# Tạo Controller Trong Codeigniter

Tất cả các controller trong codeigniter đều được đặt trong thư mục **Application/Controllers** của CI. Mặc định khi cài đặt CI đã tạo một controller tên là **welcom.php**, bạn xóa file này đi và tạo một file **hello.php** và điền nội dung vào là:

```
if (!defined('BASEPATH'))
exit('No direct script access allowed');

class Hello extends CI_Controller {

    public function index() {
        echo 'Fast Track SE Hello Controller';
    }

    public function other(){
        echo 'Fast Track SE  Other Controller';
    }

}
```

Bây giờ bạn ra trình duyệt gõ đường dẫn “*localhost/codeigniter/index.php/hello/other*” màn hình sẽ xuất hiện dòng “*Fast Track SE Other Controller*”.

# Truyền Biến Vào Controller

Trong mô hình MVC của các Framework, biến truyền vào theo phương thức GET đều có dạng “*domain.com/controller/action/parameter1/parameter2/...*”, Trong Codeigniter cũng vậy để truyền biến vào Controller bạn sẽ có đường dẫn là “*domain.com/index.php/controller/action/parameter1/parameter2/...*” Trong hàm (Action) của controller ta sẽ nhận nó bằng cách truyền những biến có vị trí tương ứng với từng parameter trên url.

```
class Hello extends CI_Controller
{
    public function index($message = '')
    {
        echo 'Fasttrack.edu.vn ' . $message;
    }
}
```

Gõ đường dẫn “*localhost/codeigniter/index.php/hello/index/Hello Codeigniter Newbie*” thì màn hình sẽ xuất hiện chữ “*Fasttrack.edu.vn Hello Codeigniter*”

# Xác Định Controller Mặc Định

**Controller mặc định** là controller sẽ được gọi khi trên url bạn không gọi đến một controller nào. Bạn vào file **application/config/routes.php** kéo xuống phía dưới tìm đến dòng `$route['default_controller'] = "welcome";`. Tại đây bạn sửa giá trị của biến `$route['default_controller']` thành tên controller mà bạn muốn chạy mặc định, ví dụ tôi sẽ sửa thành `$route['default_controller'] = "hello/other";` sau đó ra trình duyệt gõ đường dẫn **“localhost/codeigniter”** thì mặc định nó sẽ chạy đến controller **“hello/other”**.



# Hàm Khởi Tạo

- Trong PHP hàm khởi tạo được quy ước là đặt trùng tên với tên Lớp hoặc là bạn đặt tên **\_\_construct()**.
- Trong lập trình hướng đối tượng thì tất cả các lớp đối tượng có hàm khởi tạo, hàm này sẽ chạy đầu tiên khi bạn khởi tạo một đối tượng mới.

```
class Hello extends CI_Controller
{
    // Hàm khởi tạo
    function __construct() {
        // Gọi đến hàm khởi tạo của cha
        parent::__construct();
    }

    public function index()
    {
        echo 'Fasttrack.edu.vn';
    }
}
```

# Xóa Đường Dẫn Index.php trong codeigniter

Để bỏ file index.php trên đường dẫn url trong codeigniter bạn làm như sau:

Tạo file .htaccess cùng cấp với file index.php, tức là ở ngoài cùng, sau đó copy nội dung này vào.

Options +FollowSymlinks

RewriteEngine on

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteCond %{REQUEST\_FILENAME} !-d

RewriteRule ^(.+)/?\$ index.php/\$1 [L]

Sau đó bạn ra trình duyệt gõ URL “localhost/codeigniter/hello/other” thì CI tự động hiểu và gọi đến Controller **Hello** và hàm (Action) **other**

# Tạo Mới Một View Trong Codeigniter

- Trong Codeigniter tất cả các Views đều được đặt trong thư mục application/views. Các bạn vào folder đó tạo một file tên là login\_view.php, trong view các bạn tạo một form login như sau:

File `application/views/login_view.php`:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title></title>
5          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6      </head>
7      <body>
8          <form method="POST" action="">
9              Username : <input type="text" name="username" value=""/> <br/>
10             Password : <input type="password" name="password" value=""/> <br/>
11             <input type="submit" name="submit_login" value="Login"/>
12          </form>
13      </body>
14  </html>
```

# Load View Trong Controller

- **Load View:**

Sau khi có view login\_view.php ở trên, bạn vào tạo một controller mới với tên là Login có nội dung như sau:

File application/controllers/Login.php:

```
1  class Login extends CI_Controller
2  {
3      // Hàm load form login
4      public function load_form()
5      {
6          // Load view
7          $this->load->view('login_view');
8      }
9  }
```

# Load View Trong Controller

- **Load Nhiều View:**

Để load nhiều view bạn chỉ cần dùng cú pháp load view nhiều lần, ví dụ:

```
1  $this->load->view('view1');  
2  $this->load->view('view2');  
3  $this->load->view('view3');  
4  $this->load->view('view4');  
5  $this->load->view('view5');
```

- **Load View Ở Sub Folder**

```
1  $this->load->view('subfolder/view1');
```

# Load View Trong Controller

- **Truyền Dữ Liệu Qua View**

Để truyền dữ liệu qua view thì tất cả dữ liệu bạn phải đưa vào một **mảng kết hợp** \$data, Controller sẽ tự động tạo các biến bên view tương ứng với các key và các value trong mảng \$data đó. Mỗi dữ liệu có thể ở các kiểu như: float, double, int, string, object, array.

Trong controller Login: (File application/controllers/Login.php)

```
1  class Login extends CI_Controller
2  {
3      // Hàm load form login
4      public function form()
5      {
6          // Data cần truyền qua view
7          $data = array(
8              'title' => 'Đây là trang login',
9              'message' => 'Nhập Thông Tin Đăng Nhập'
10         );
11
12         // Load view và truyền data qua view
13         $this->load->view('login_form', $data);
14     }
15 }
```

# Load Model Trong Codeigniter

- **Cấu Hình Database**

Trong Codeigniter để kết nối với Database chúng ta phải cấu hình thông tin cho đúng. Bạn ở file application/config/database.phpsau đó kéo xuống bên dưới chỉnh lại một số thông tin sau:

```
1  $db['default']['hostname'] = 'localhost';
2  $db['default']['username'] = 'Tên Đăng Nhập Database'; // Chỉnh ở đây
3  $db['default']['password'] = 'Mật Khẩu Đăng Nhập'; // Chỉnh ở đây
4  $db['default']['database'] = 'Tên Database'; // Chỉnh ở đây
5  $db['default']['dbdriver'] = 'mysql';
6  $db['default']['dbtextareafix'] = '';
7  $db['default']['pconnect'] = TRUE;
8  $db['default']['db_debug'] = TRUE;
9  $db['default']['cache_on'] = FALSE;
10 $db['default']['cachedir'] = '';
11 $db['default']['char_set'] = 'utf8';
12 $db['default']['dbcollat'] = 'utf8_general_ci';
13 $db['default']['swap_textarea'] = '';
14 $db['default']['autoinit'] = TRUE;
15 $db['default']['stricton'] = FALSE;
```

# Tạo Mới Một Model Trong Codeigniter

- Tất cả các file model đều nằm trong thư mục application/models nên bạn vào đó tạo một file Users\_model.php có nội dung như sau:

```
1  class User extends CI_Controller{
2      public function __construct() {
3          parent::__construct();
4          $this->load->database();
5      }
6
7      public function index() {
8          $query=$this->db->query("SELECT * FROM USER order By id");
9          $data=$query->result_array();
10         return $data;
11     }
12
13 }
```

Trong đó Users\_model là tên model của bạn, nó phải giống với tên file Users\_model.php và ký tự đầu tiên phải ghi hoa. Mọi model đều phải kế thừa lớp CI\_Model thì mới hoạt động được. Vậy là bạn đã tạo xong model **Users\_model** rồi đó.



# Load Model Trong Controller

- Để load model trong controller ta dùng cú pháp: `$this->load->model('ten_model')`, và để gọi các hàm trong model ta dùng cú pháp: `$this->ten_model->ham()`.

```
1  class News extends CI_Controller
2  {
3      function news_list()
4      {
5          // Load model
6          $this->load->model('news_model');
7
8          // Gọi function trong model
9          $news_list = $this->news_model->getList();
10     }
11 }
```

# Load Model Trong Controller

- Trong ứng dụng nếu một model nào đó luôn luôn sử dụng thì bạn có thể dùng chức năng Autoload của Codeigniter, bạn mở file **application/config/autoload.php** và tìm đến dòng (hình như dòng cuối cùng) **\$autoload['model'] = array()**, sau đó bạn thêm model bạn cần load vào, ví dụ tôi thêm model news\_model: **\$autoload['model'] = array('news\_model')**.
- Sau khi thêm model vào autoload thì ở controller bạn chỉ cần sử dụng nó mà không cần phải load model đó ra

```
1 class News extends CI_Controller
2 {
3     function news_list()
4     {
5         // Gọi function trong model
6         $news_list = $this->news_model->getList();
7     }
8 }
```

# Q&A