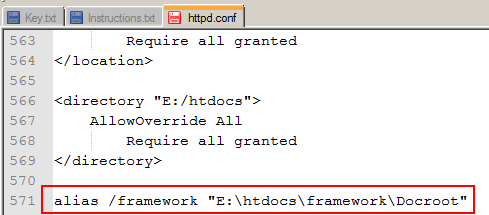
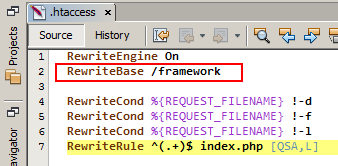
# Cài đặt

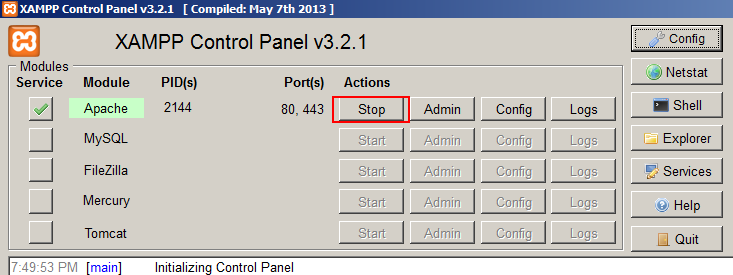
Đặt alias cho httpd.conf, vd: /**framework**



Đặt rewritebase cho **/Docroot/.htaccess**



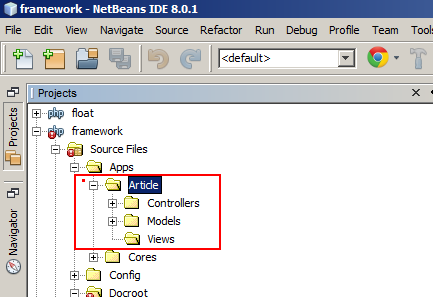
Restart Apache



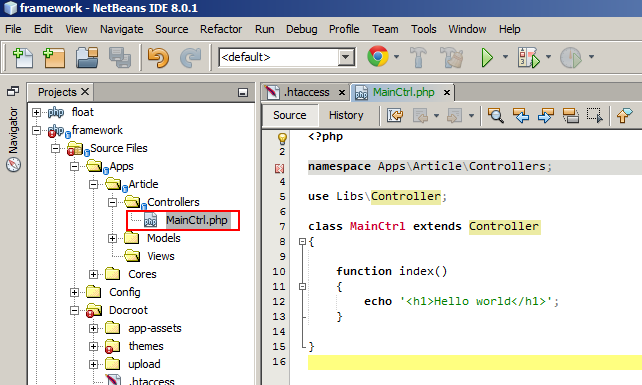
# Controller & routing

## Cơ bản

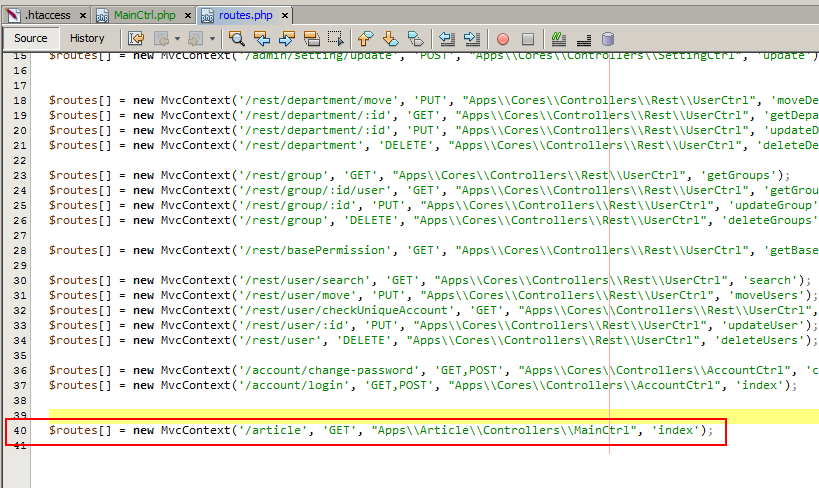
Tạo application mới tên là **article** theo cấu trúc thư mục sau:



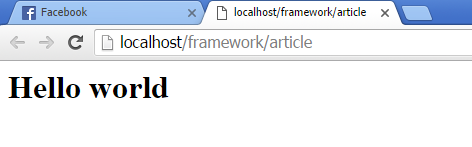
Tạo **MainCtrl.php**:



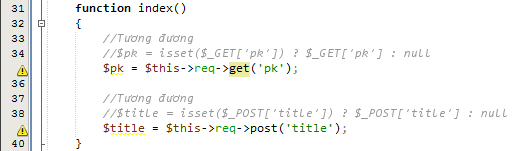
Trỏ đường dẫn vào controller qua **routes.php**:



Truy cập vào <http://localhost/framework/article>:



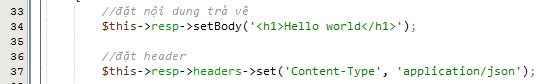
## Đối tượng xử lý Request



Đọc thêm hướng dẫn tại: <http://docs.slimframework.com/request/overview/>

## Đối tượng xử lý Response

Đối tượng xử lý việc trả dữ liệu về cho trình duyệt.

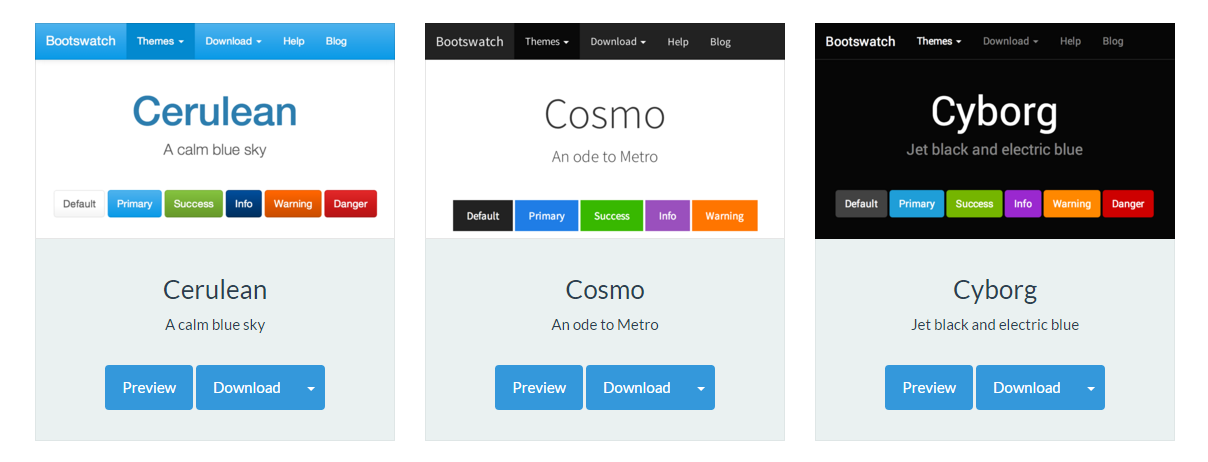


Đọc thêm hướng dẫn tại: <http://docs.slimframework.com/response/overview>

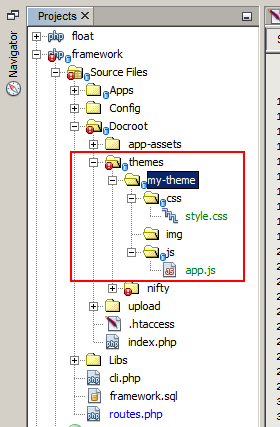
# View

## Theme

Theme là nơi chứa HTML, CSS, JS. Theme định nghĩa màu, kiểu chữ, giao diện,... Ví dụ một vài theme (mỗi hình dưới là một theme):

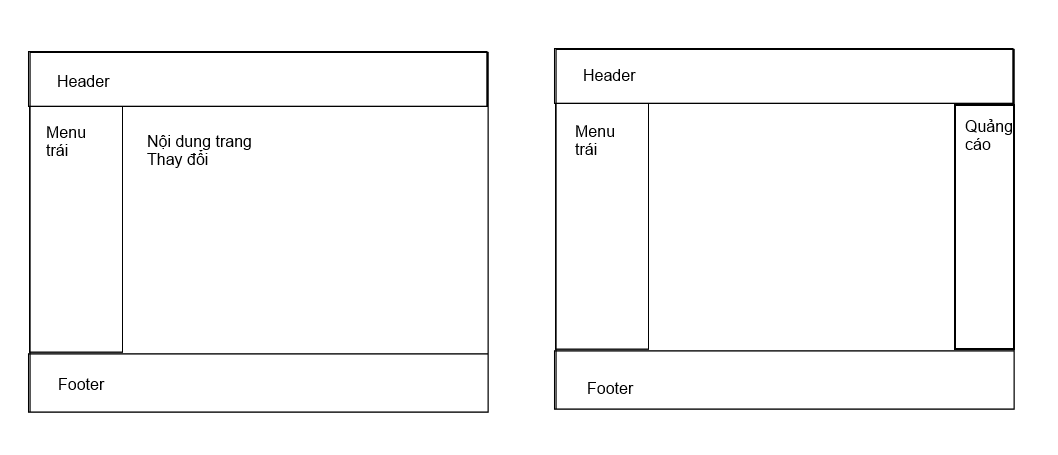


Tạo “**my-theme**” theo hình sau (có thể nhiều thư mục hơn tùy theme):



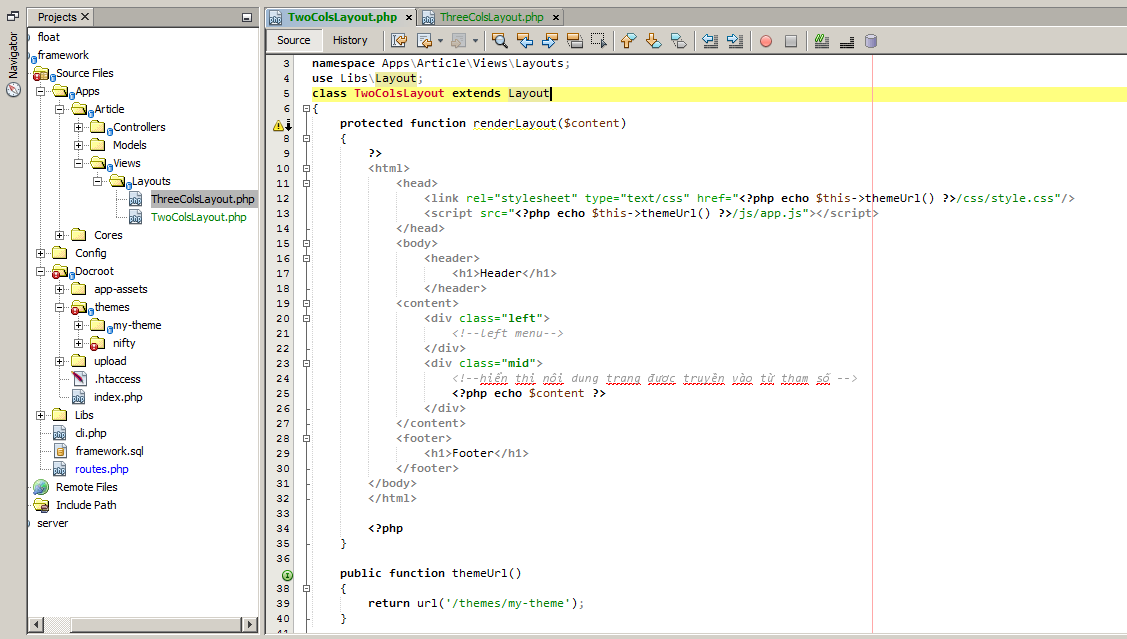
## Layout

Layout là khung lặp đi lặp lại của trang web, tạo layout cho phép sử dụng lại header, footer, sidebar,... Ví dụ dưới đây thể hiện 2 layout cùng theme nhưng bố cục hơi khác nhau.



Layout thứ 2 có thêm cột quảng cáo

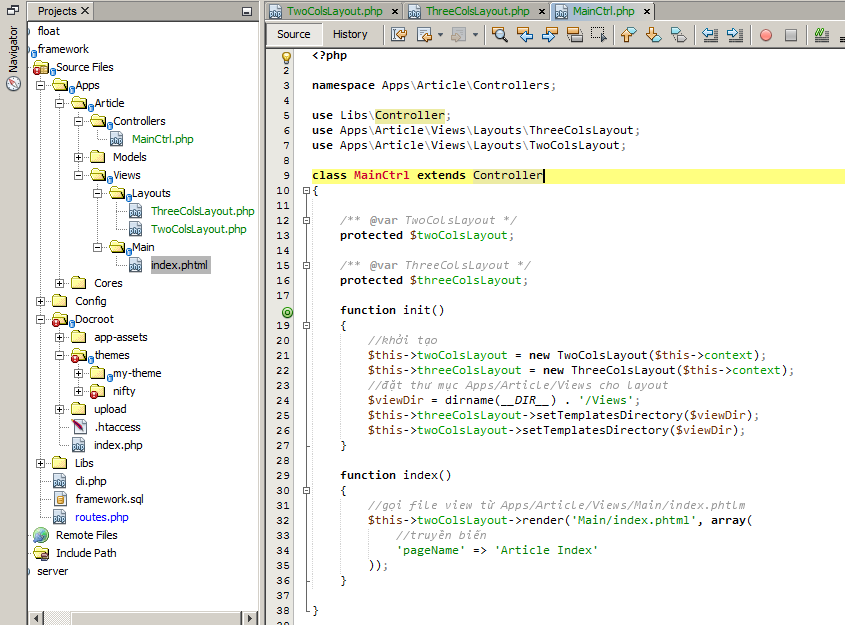
Tạo layout sử dụng **my-theme** như hình sau:



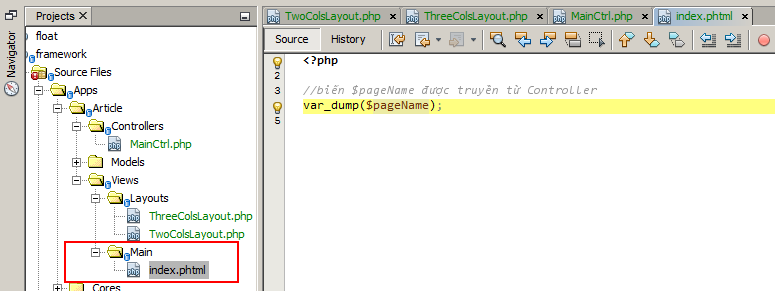
ThreeColsLayout có thêm cột phải:



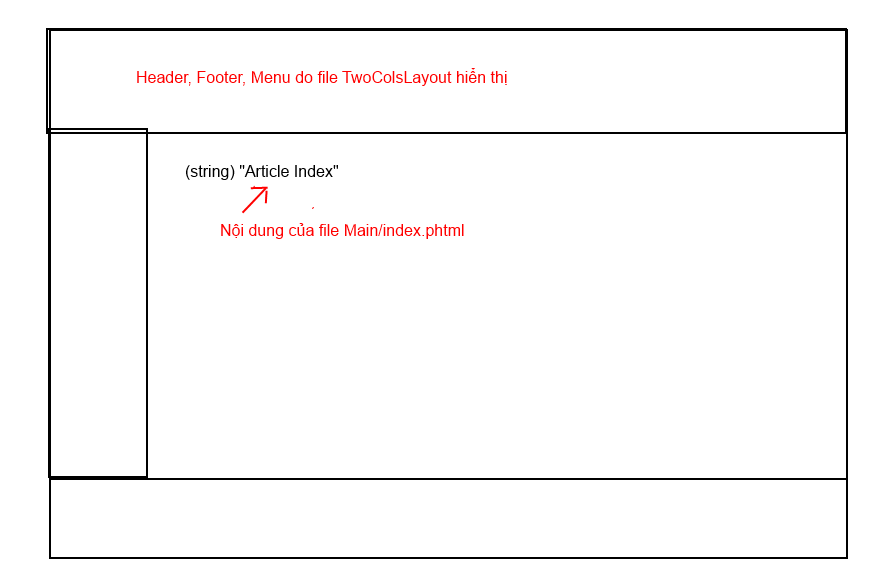
Từ MainCtrl, gọi layout:



Khai báo file Main/index.phtml



Chạy thử kết quả

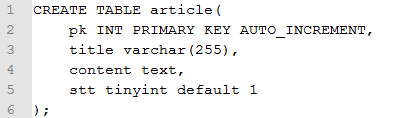


# Model

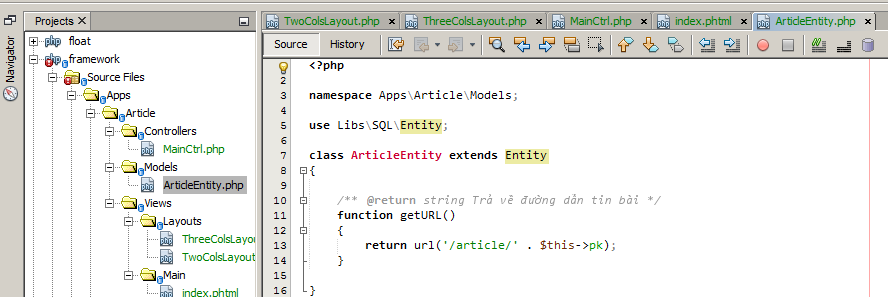
Model đảm nhiệm việc tương tác với database. Có hai loại model là Mapper và Entity.

## Entity

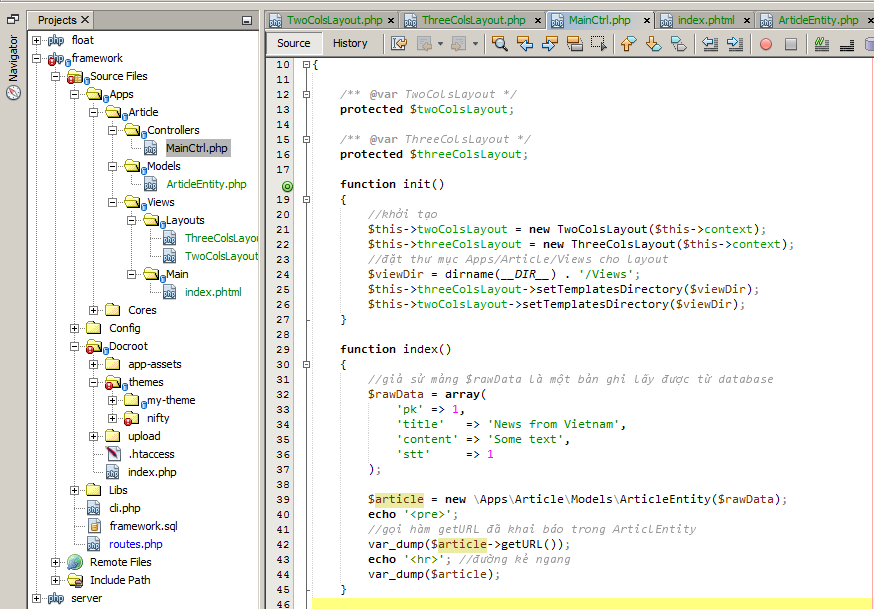
Entity là class chứa dữ liệu lấy từ database, vd bảng article:

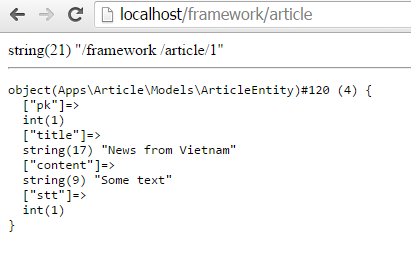


Tạo class ArticleEntity:



Sử dụng entity:



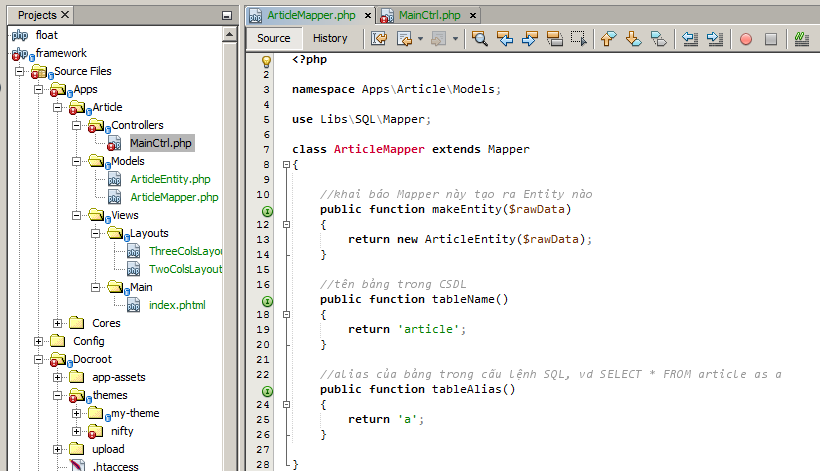


## Mapper

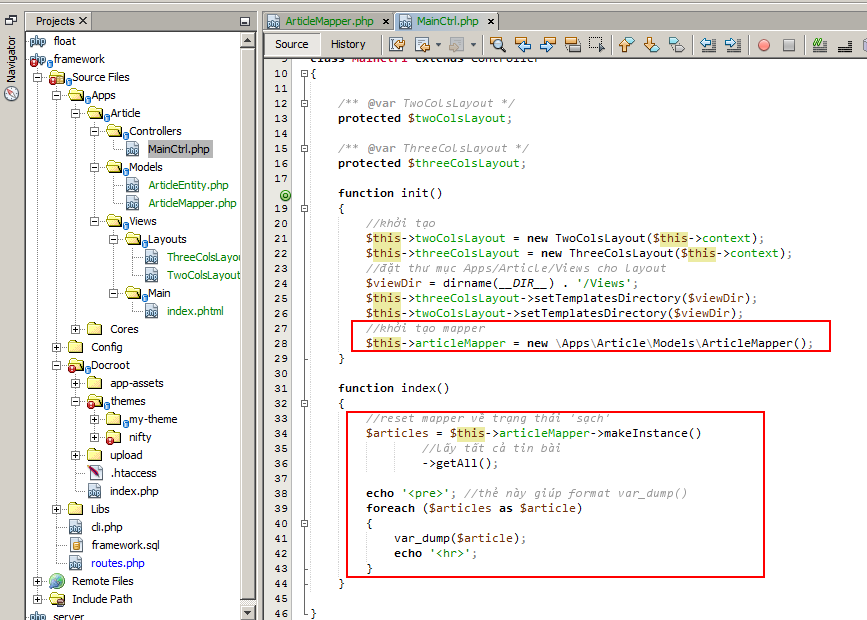
Mapper là class xử lý nghiệp vụ, giao tiếp với CSDL để tạo ra Entity. Thông thường lập trình viên phải viết SQL thuần “SELECT \* FROM article”, khi sử dụng Mapper không phải viết tay SQL mà chỉ cần gọi hàm.

### Khai báo Mapper

tạo class Article Mapper

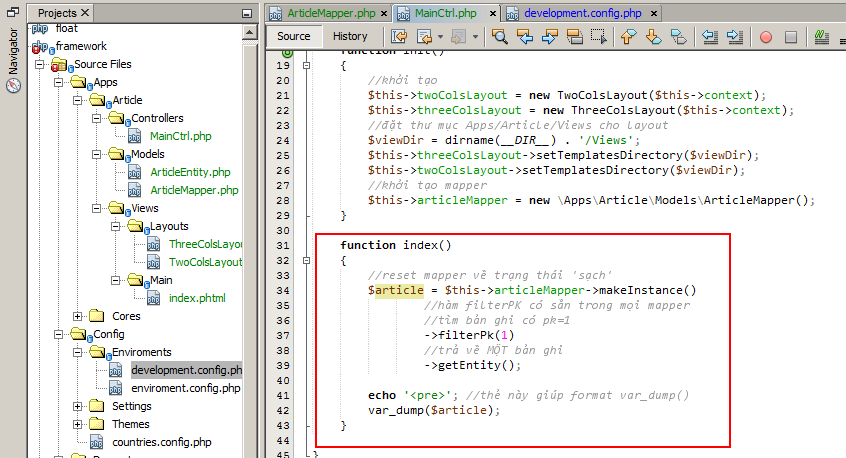


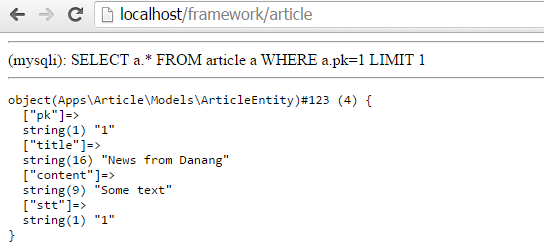
Sử dụng Article mapper để lấy danh sách tin bài:





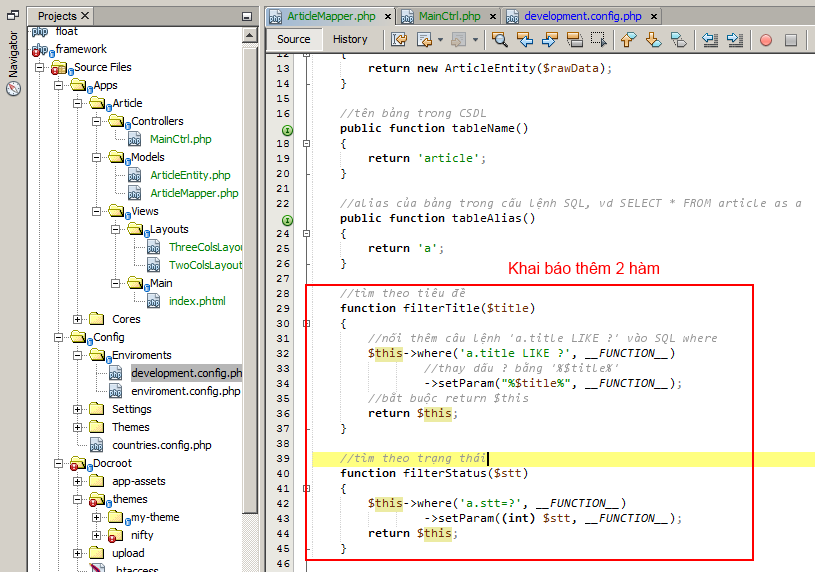
Sử dụng ArticleMapper để lấy MỘT bản ghi:

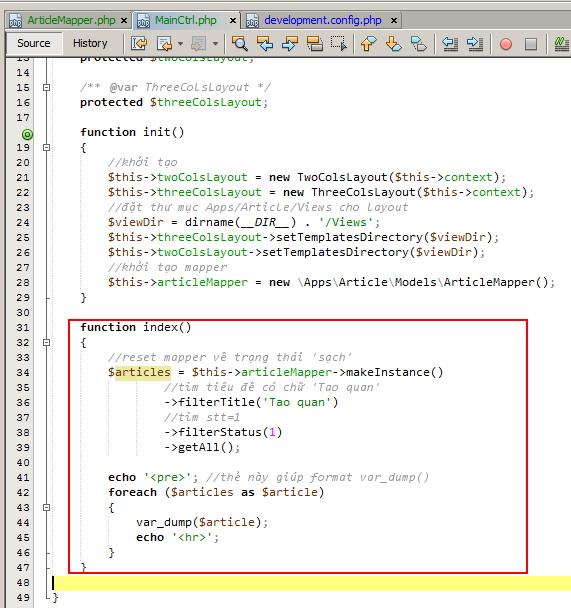


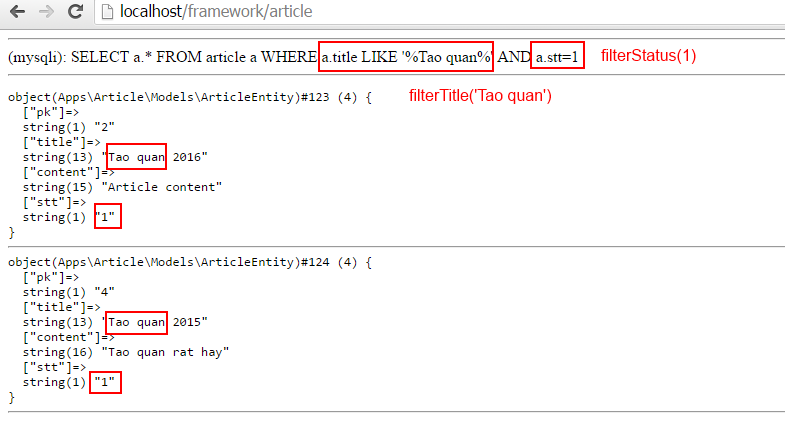


### Điều kiện lọc theo nghiệp vụ

Ta có thể khai báo điều kiện lọc theo nghiệp vụ, vd: muốn tìm kiếm tin bài theo tiêu đề và có trạng thái hoạt động (1=hoạt động, 0=không hoạt động).

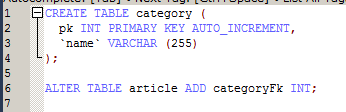


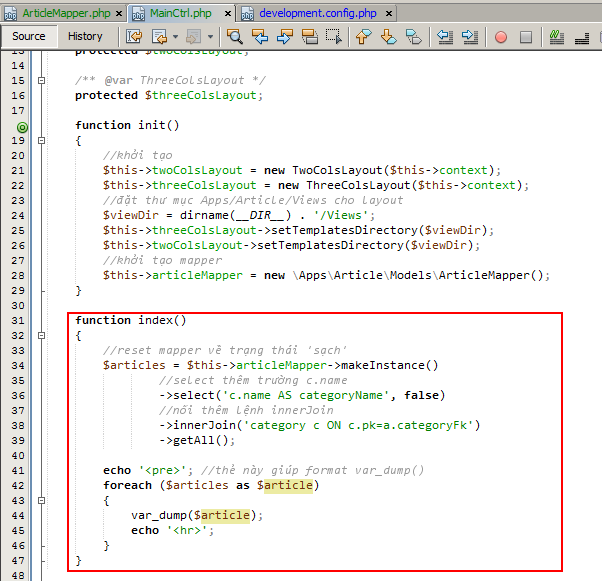


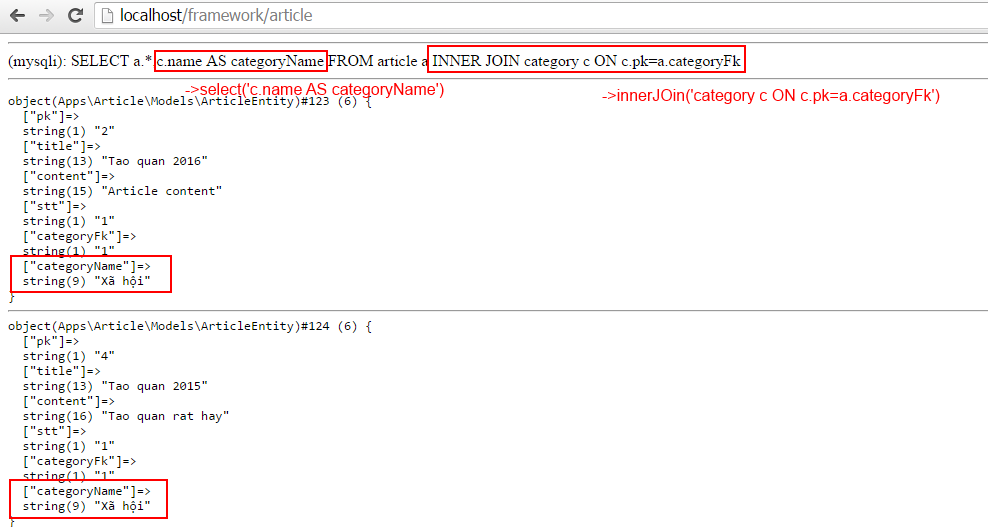


### Nối thêm câu lệnh SQL vào Mapper

Mapper cung cấp các hàm, cách viết gần giống viết SQL thuần. VD, muốn join bảng category để lấy thêm tiêu đề chuyên mục.



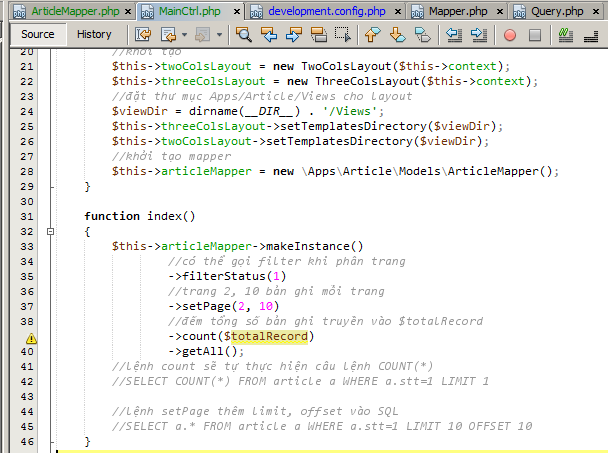




Các hàm nối SQL:

|  |  |
| --- | --- |
| select($field, $override = true) | $override =true: xóa hết các trường đã select    $override =false: nối thêm trường vào select trước |
| from($table) |  |
| join($table, $key=null) | nối câu lệnh JOIN  vd: ->join(‘category c ON c.pk=a.categoryFk’) |
| innerJoin($table, $key=null) |  |
| leftJoin($table, $key=null) |  |
| rightJoin($table, $key=null) |  |
| fullJoin($table, $key=null) |  |
| where($statement, $where\_key = null) | nối câu lệnh where |
| orderBy($fields) |  |
| groupBy($field) |  |
| limit($limit)  và  offset($offset) |  |

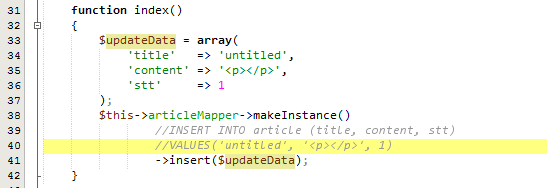
### Phân trang



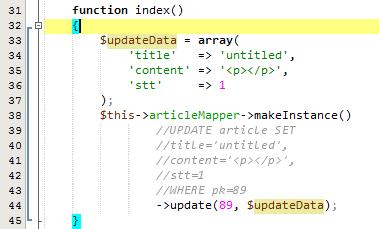
### Insert, Update, Delete

Để tiện sử dụng, mapper hỗ trợ các hàm insert, update, delete nhanh:

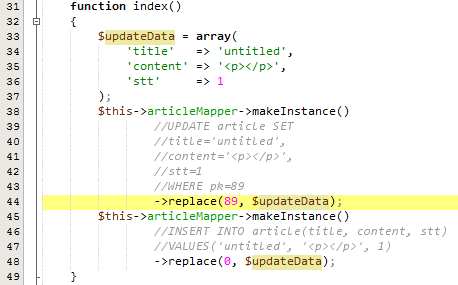
INSERT



UPDATE

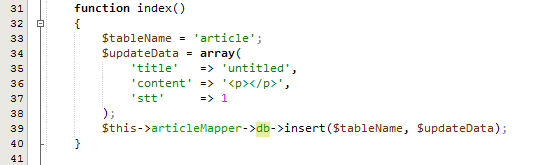


REPLACE

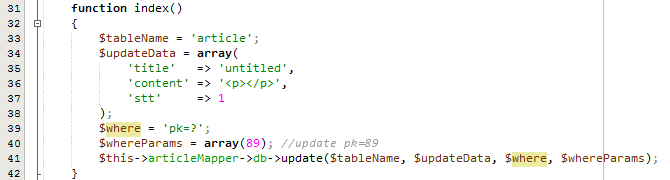


Nếu muốn thực hiện câu lệnh INSERT, UPDATE, DELETE sâu hơn sử dụng đối tượng $db:

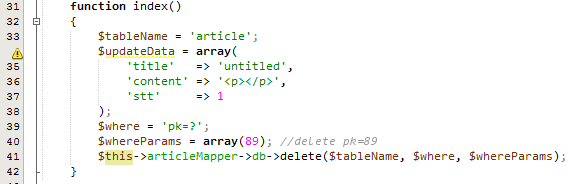
INSERT



UPDATE:



DELETE



### Custom SQL

Mapper hỗ trợ hầu hết query SELECT, UPDATE, INSERT cơ bản, tuy nhiên một số query phải viết SQL thuần.

VD: thống kê từng category có bao nhiêu tin bài.

