VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

ĐỒ ÁN

**TỐT NGHIỆP ĐẠI HỌC**

NGÀNH CÔNG NGHỆ THÔNG TIN

**PHƯƠNG PHÁP TẠO PLUGIN CHO FRAMEWORK THƯƠNG MẠI ĐIỆN TỬ ECCUBE 3**

|  |  |  |
| --- | --- | --- |
| Sinh viên thực hiện | : | **Vũ Tiến Dũng** |
| Lớp | : | IS1 – K56 |
| Giáo viên hướng dẫn | : | **TS. Phạm Huy Hoàng** |

HÀ NỘI 5-2016

# PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: VŨ TIẾN DŨNG

|  |  |  |
| --- | --- | --- |
| Điện thoại liên lạc: 0165 451 8283 | | Email: tiendung\_px@bka.vn |
| Lớp: IS1 – K56 | Hệ đào tạo: Đại học chính quy | |

Đồ án tốt nghiệp được thực hiện tại: Bộ môn Truyền thông và Mạng máy tính, Viện CNTT&TT, Đại học Bách Khoa Hà Nội.

Thời gian làm ĐATN: Từ ngày 21/02/2016 đến 29/05/2016

1. Mục đích nội dung của ĐATN
   * Phương pháp tạo plugin cho hệ thống thương mại điện tử sử dụng framework eccube version 3
2. Các nhiệm vụ cụ thể của ĐATN
   * Tìm hiểu cấu trúc và công nghệ của framework thương mại điện tử eccube 3
   * Tìm hiểu cấu trúc căn bản của plugin cho eccube 3
   * Tìm hiểu các hookpoint để plugin sử dụng trong eccube 3
   * Tìm hiểu xử lý database từ plugin bằng migration của doctrine trên eccube 3
   * Demo 2 plugin quickOrder và exportCatalogue
3. Lời cam đoan của sinh viên:

Tôi – *Vũ Tiến Dũng* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Phạm Huy Hoàng*

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

*Hà Nội, ngày* *tháng* *năm*

Tác giả ĐATN

*Hoàng Công Thắng*

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

*Hà Nội, ngày* *tháng* *năm*

Giáo viên hướng dẫn

*TS.Phạm Huy Hoàng*

# TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Công nghệ tiên tiến hiện nay giúp doanh nghiệp biến Website của mình thành những siêu thị hàng hóa trên Internet, biến người mua thực sự trở thành những người chủ với toàn quyền trong việc lựa chọn sản phẩm, tìm kiếm thông tin, so sánh giá cả, đặt mua hàng, ký hợp đồng với hệ thống tính toán tiền tự động rõ rang, trung thực.

Thương mại điện tử (E-Commerce) là hình thái hoạt động kinh doanh bằng c ác phương pháp điện tử, là việc trao đổi thông tin kinh doanh thông qua các phương tiện công nghệ điện tử

Eccube là framework về thương mại điện tử hàng đầu được sử dụng tại Nhật Bản. Ngày 1/7/2015 eccube đã cho ra phiên bản 3 gồm đầy đủ các chức năng cơ bản của hệ thống thương mại điện tử cùng với đó là khả năng cho phép các plugin thao tác thay đổi cải thiện các chức năng sẵn có và tạo chức năng mới.

Đồ án này trình bày về cách tạo và các can thiệp có thể làm từ plugin cho các website thương mại điện tử sử dụng framework eccube 3.

# ABSTRACT OF THESIS

In large enterprise networks or server systems, the analysis and network traffic monitoring system is very important to ensure the maintenance of the stability and performance of the network. Ability to analyze and monitor the network traffic in the operating system is very important for network administrators. The inspection of network traffic flows to ensure network resources are used more sensibly. It helps administrators to recognize the quality of service (QoS), recognize the signs or risk of attacks Denial of Service (DoS), virus propagation, as well as numerous other incidents.

NetFlow is a Cisco IOS feature that allows traffic statistics through the router package. NetFlow performance monitoring, analysis, flow calculation package. Meet most of the requirements in the monitoring network.

This project provides information about how to operate the integrated NetFlow in Cisco devices. Thereby setting and install NetFlow on a small server systems. Extract data for use NetFlow analysis on network traffic anomaly.

# LỜI CẢM ƠN

Trước hết, tôi xin gửi lời cảm ơn chân thành tới TS. Phạm Huy Hoàng, người đã tận tình dạy dỗ và hướng dẫn tôi trong quá trình hoàn thành đồ án cũng như trong học tập.

Đồng thời, tôi xin bày tỏ lòng biết ơn đến các thầy cô giáo trong Viện Công nghệ thông tin và Truyền thông – trường Đại học Bách Khoa Hà Nội, những người đã tận tâm giảng dạy, truyền đạt cho tôi những kiến thức cơ bản làm nền tảng cho việc thực hiện đồ án cũng như trong quá trình công tác sau này.

Tôi cũng xin gửi lời cảm ơn tới các anh chị tại trường Đại học Bách Khoa Hà Nội, các bạn, các tôi cùng làm việc trong phòng thí nghiệm, những người luôn ở bên cạnh giúp đỡ, động viên tôi trong quá trình hoàn thành đồ án.

Cuối cùng, với tất cả sự kính trọng, con xin bày tỏ lòng biết ơn sâu sắc tới bố mẹ và anh chị tôi trong gia đình đã luôn là chỗ dựa tinh thần vững chắc và tạo mọi điều kiện cho con ăn học nên người.

*Tôi xin chân thành cảm ơn !*

# MUC LỤC

Contents

[PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP 2](#_Toc451194867)

[TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP 3](#_Toc451194868)

[ABSTRACT OF THESIS 4](#_Toc451194869)

[LỜI CẢM ƠN 5](#_Toc451194870)

[MUC LỤC 6](#_Toc451194871)

[GIỚI THIỆU ĐỀ TÀI 7](#_Toc451194872)

[**1.** **Tổng quan** 7](#_Toc451194873)

[**2.** **Giới thiệu nội dung chính** 7](#_Toc451194874)

[CHƯƠNG I. CƠ SỞ LÝ THUYẾT 8](#_Toc451194875)

[**1.** **Giới thiệu Framework Eccube 3** 8](#_Toc451194876)

[**2.** **Giới thiệu về Plugin căn bản của Eccube 3** 16](#_Toc451194877)

[**3.** **Các loại hook point và cách sử dụng** 19](#_Toc451194878)

[Hook point riêng 20](#_Toc451194879)

[Hook point chung 21](#_Toc451194880)

[**4.** **Tạo page mới từ plugin** 24](#_Toc451194881)

[**5.** **Tạo menu mới trong admin từ Plugin** 25](#_Toc451194882)

[**6.** **Chỉnh sửa cấu trúc database dùng migration doctrine từ plugin** 25](#_Toc451194883)

[CHƯƠNG II. Plugin QuickOrder và ExportCatalogue: 30](#_Toc451194884)

# GIỚI THIỆU ĐỀ TÀI

1. **Tổng quan**

* Ngày nay khi ngày càng có nhiều website thương mại điện tử ra đời việc phát triển thêm các chức năng hay tối ưu các chức năng sẵn có giúp cho người dùng sử dụng website thêm thuận tiện sẽ giúp cho việc phát triển kinh doanh theo hình thức thương mại điện tử.
* Với các hệ thống thương mại điện tử lớn đang chạy việc thay đổi nâng cấp sẽ gặp rất nhiều khó khan. Khi muốn đưa một ý tưởng kinh doanh mới vào để chạy thử website cần có sự mềm dẻo để có thể thay đổi nhanh chóng. Bằng việc sử dụng plugin các website dùng eccube 3 có thể chạy thử một chức năng từ plugin và bỏ đi khi không cần thiết một cách rất linh hoạt và dễ dàng
* Đồ án thực hiện tìm hiểu cách tạo plugin cho hệ thống website dùng eccube 3 giúp tạo nên 1 cộng đồng làm plugin lớn mạnh phát triển

1. **Giới thiệu nội dung chính**

Đồ án được chia thành các phần chính như sau:

* CHƯƠNG I. CƠ SỞ LÝ THUYẾT LIÊN QUAN:

Chương này trình bày kiến thức căn bản các kỹ thuật của framework eccube version 3

Các kiến thức căn bản về cấu trúc plugin, các hookpoint và migration database của doctrine trong eccube 3

* CHƯƠNG II. Plugin QuickOrder và ExportCatalogue:

Chương này mô tả 2 plugin QuickOrder và ExportCatalogue

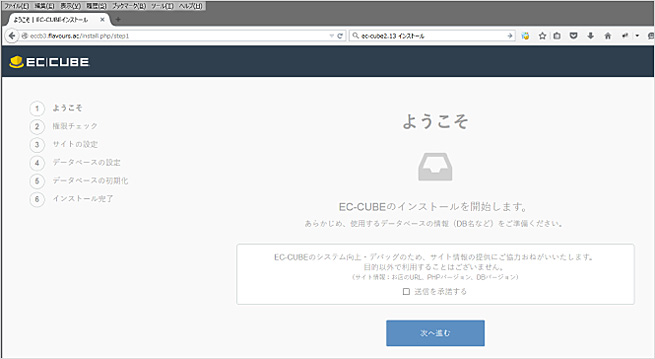
* KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN:

Trong phần này em xin phép trình bày đánh giá về hệ thống hiện tại cùng một vài hướng cải tiến mở rộng hệ thống chưa áp dụng được trong khuôn khổ đồ án.

# CHƯƠNG I. CƠ SỞ LÝ THUYẾT

1. **Giới thiệu Framework Eccube 3**
   1. Giới thiệu

* Eccube là framework chuyên về eCommerce rất thông dụng tại Nhật Bản.
* Hệ thống eccube được đánh giá là framework eCommerce số 1 tại Nhật.
* Eccube 3 sử dụng được xây dựng trên nền tảng của symfony có sử dụng doctrine orm (object orientation mapping) để dễ dàng thao tác và xử lý với database. Eccube 3 có thể cài đặt và sử dụng với mysql, postgresql…
* Eccube có sẵn jquery và sử dụng Template Engine twig để làm giao diện
* Eccube 3 yêu cầu apache 2.2 trở lên có mod\_rewrite, php 5.4 trở lên mysql 5.1 trở lên, postgresql 8.4 trở lên
  1. Tính năng
* Eccube 3 có khá đầy đủ các chức năng cần thiết của 1 website thương mại điện tử:
  + Tạo giỏ hàng (cart) cho cả thành viên và những khách ko phải thành viên.
  + Đặt hàng(Order): Eccube hỗ trợ đặt hàng cho các thành viên với các lựa chọn cách thanh toán cách chuyển hàng…
  + Admin có thể quản lý thành viên các đơn đặt hàng ….
  + Thống kê các sản phẩm được đặt hàng(dành cho admin) đây là 1 chức năng rất quan trọng giúp đánh giá và phát triển những sản phẩm được nhiều người quan tâm
  1. Các bước cài đặt Eccube 3
     1. Màn hình bắt đầu cài đặt



* + 1. Click「次へ進む」Các yêu cầu về permission đã có ở thư mục mặc định. Tham khảo nội dung đó và thay đổi permission nếu cần

Dưới đây là màn hình yêu cầu thay đổi permission thành (777)



Nếu permission đã được thiết lập đúng màn hình sẽ có thông báo như dưới đây:



* + 1. Click「次へ進む」sẽ hiển thị màn hình nhập thông tin cơ bản của website



* + 1. Sau khi nhập các thông tin của site click 「次へ進む」sẽ chuyển đến màn hình nhập thông tin của database



* + 1. Sau khi nhập thông tin của database click「次へ進む」sẽ chuyển đến màn hình xử lý tạo database sample nếu là install lần đầu không check 「データベースの初期化を行わない」  
       Tiếp tục click「次へ進む」



1. Chờ database được khởi tạo là quá trình cài đặt hoàn tất
2. Cuối cùng xóa file 「html/install.php」



Màn hình front end:



Màn hình admin:



1. **Giới thiệu về Plugin căn bản của Eccube 3**

**{PluginName}**

* config.yml (quan trọng): Ghi các thong tin cơ sở của plugin.
* PluginManager.php (quan trọng):Ghi lại các thao tác xử lý trên plugin như thời điểm install plugin,thời điểm update…
* ServiceProvider/\*.php:Các file được gọi bằng ServiceProvider được định nghĩa trong config.yml.
* event.yml:Định nghĩa các hookpoint được sử dụng.
* {EventName}.php:Ghi lại các xử lý can thiệp lên hookpoint.
* Resource/doctrine/\*.dcm.yml :Ghi định nghĩa bảng.
* Resource/doctrine/migration/Version{YYYYmmddHHiiss}.php:Định nghĩa việc khởi tạo hoặc xóa bảng
* Form/Type/\*.php, Form/Extension/\*.php:Ghi định nghĩa tự mở rộng của form.

**config.yml**

Mô tả file setting trong toàn bộ plugin.

[Các mục setting]

● name

Tên plugin được hiển thị sau khi install.(đặt tên tùy ý)

● version

Hiển thị version sau khi install.(tùy ý).Được sử dụng để quản lý việc upversion.

● code

Mã được chủ cửa hàng dùng để phân biệt plugin.(Dùng ký tự Anh/Được thống nhất trong các chủ cửa hàng)

● event

Ghi lại tên file xử lý nghiệp vụ event.Được đọc trong file {EventName}.php

● service

Có thể định nghĩa file dùng để đọc logic tự đặt ra bên ngoài event. (hàng đợi Yaml/Khả năng chỉ định phức số)

Các file mô tả ở đây được đọc trong ServiceProvider Directory.

● orm.path

Được dùng để mô tả Directory sắp xếp của file định nghĩa tạo bảng.

Ví dụ mô tả config.yml

name: Nội dung của category ※quan trọng   
code: CategoryContents ※quan trọng  
version: 1.0.0 ※quan trọng  
service:

CategoryContentsServiceProvider

orm.path:

/Resource/doctrine

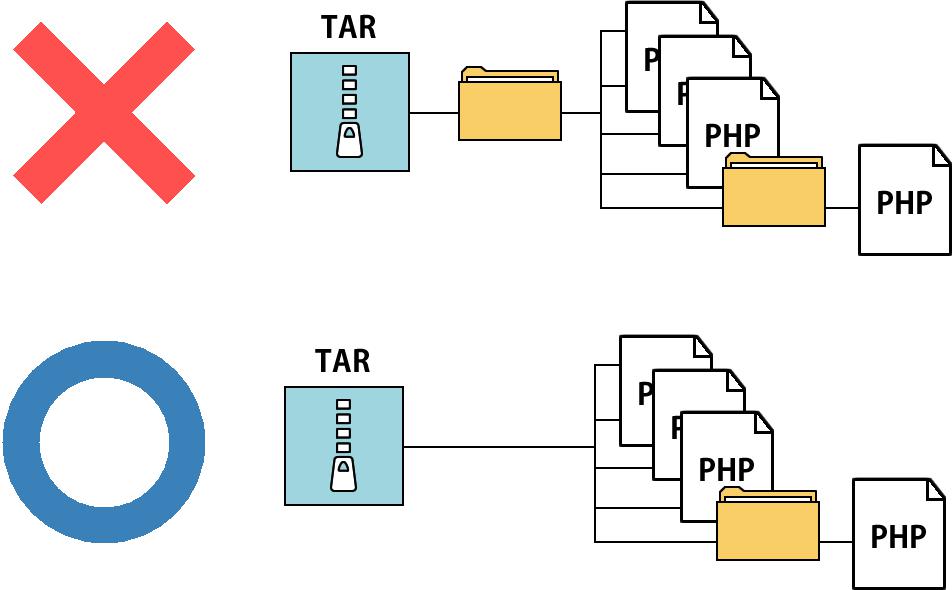
※Nếu thêm bảng bằng plugin,mô tả của orm.path là quan trọng.  
event: CategoryContentsEvent   
※Nếu them event thì mô tả của event là quan trọng.

**event.yml**  
File định nghĩa event sẽ được sử dụng.  
[Các mục setting]  
{HookPointName}:

* [{MethodName}, {Priority}]
* [{MethodName}, {Priority}]

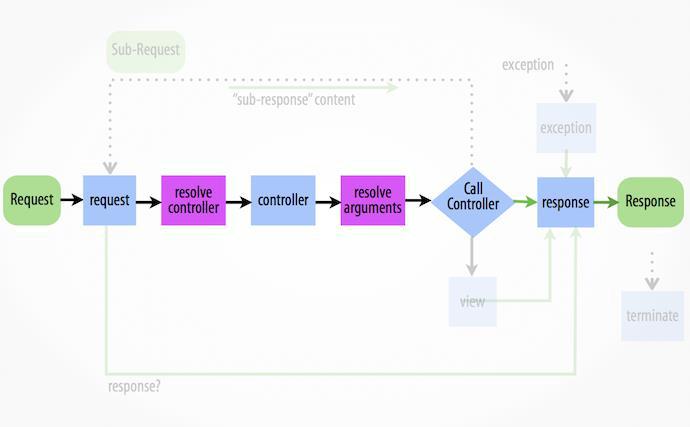
● HookPointName   
Ghi lại tên hookpoint được sử dụng.   
● Mô tả các method chứa trong file MethodName Event.  
● Priority   
Thứ tự chỉ định là FIRST / NORMAL / LAST   
Ví dụ về mô tả event.yml

1. front page event Product/list.twig:
2. [onRenderProductList, NORMAL]
3. admin page event admin.product.category.index.initialize:
4. [onFormInitializeAdminProductCategory, NORMAL] admin.product.category.index.complete:
5. [onAdminProductCategoryEditComplete, NORMAL]

**{EventName}.php**  
File được gọi khi mở rộng hookpoint hoặc form nhập vào.  
Mô tả xử lý nghiệp vụ khi mở rộng hookpoint hoặc form nhập vào.  
**PluginManager.php**  
File mà EC-CUBE sử dụng khi quản lý plugin.Được gọi khi install/uninstall plugin/enable/disable plugin.Migration sẽ gọi từ file này.  
Khi định nghĩa install(), uninstall(), enable(), disable(), update(),thì sẽ được sử dụng khi install,uninstall,enable,disable,update.  
**Cách tạo lưu trữ file plugin.**  
Để nén plugin ta dung tar.gz .Chú ý khi nén không nén folder. 

1. **Các loại hook point và cách sử dụng**

EC-CUBE3 Sử dụng nền tảng framework Symfony2, Components Silex。Vì thế、Các hoạt động bên trong từ xử lý request cho đến gửi trả lời đều được thực thi tại life cycle của Http Kernel Symfony2。 Ngoài ra mỗi bước xử lý ở Http Kernel đều có cơ chế gọi là event



Life cycle của http kernel： http://symfony [.com/doc/current/components/http\_kernel/introduction.h](http://symfony.com/doc/current/components/http_kernel/introduction.html)t  [m](http://symfony.com/doc/current/components/http_kernel/introduction.html)l

EC-CUBE3 có 2 loại hookpoint

* Hookpoint sử dụng event của Symfony2
* Controller xử lý bên trong ,template,gửi mail đều có hookpoint phát triển độc lập riêng

### Hook point riêng

**a. Hookpoint trong Controller**  
Trong các controller được định nghĩa sẵn các hookpoint để plugin có thể sử dụng. List các hook point tham khảo tại file ECCUBE3\src\Eccube\Event\EccubeEvents.php  
Muốn sử dụng event nào cần khai báo tại file event.php của plugin  
Ví dụ:  
event.yml:

admin.product.category.index.initialize:

-[onFormInitializeAdminProductCategory, NORMAL]

config.yml:  
name: Sample plugin  
code: SamplePlugin  
version: 1.0.0  
event: SampleEvent

Sau khi khai báo 2 file event.yml và config.yml như trên khi CartController chạy đến vị trí event front.cart.index.initialize được đặt sẽ gọi đến function onFormInitializeAdminProductCategory() của file SampleEvent.php trong plugin  
Với hookpoint này ta có thể mở rộng 1 form có sẵn chèn các xử lý vào controller  
Ví dụ: Thêm 1 text field vào form của page đăng ký category trong admin  
SampleEvent.php

public function onFormInitializeAdminProductCategory(EventArgs $event)  
{  
// Thêm form  
$builder = $event->getArgument('builder'); $builder->add('plg\_category\_content', 'textarea', array(  
'required' => false, 'label' => false, 'mapped' => false, 'attr' => array(  
'placeholder' => 'Nhập nội dung(Có thể dung thẻ html)',  
)));

**b. Hookpoint Template**  
Mỗi template đều có 1 hook point để có thể thay đổi chỉnh sửa từ plugin  
Ví dụ:   
event.yml

Product/list.twig:

-[onRenderProductList, NORMAL]

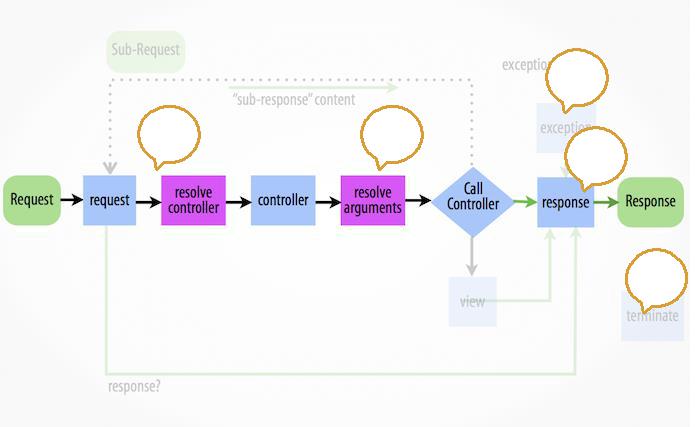
SampleEvent.php:

publlc function onRenderProductList(TemplateEvent $event)  
{  
$parameters = $event->getParameters();  
$CategoryContent = $this->app['category\_content.repository.category\_content'] ->find($Category->getId());  
*Chèn nội dung category vào code twig  
$snipet = '<div class="row">{{ CategoryContent.content | raw }}</div>'; $search = '<div id="result\_info\_box"';  
$replace = $snipet.$search;  
$source = str\_replace($search, $replace, $event->getSource());   
$event->setSource($source);*Thêm nội tung category vào tham số của twig $parameters['CategoryContent'] = $CategoryContent;   
$event->setParameters($parameters);  
}

Mỗi khi dùng template list.twig với code trên sẽ thêm 1 thẻ div vào trước thẻ div co id="result\_info\_box"

### Hook point chung

Hookpoint chung được chèn vào trong mỗi bước xử lý của life cycle httpKernel。



4

Các bước như sau đây

1. Request
2. Controller
3. Response
4. Exception
5. Terminate

**a. Hookpoint Request**  
Các event của hookpoint này:  
eccube.event.front.request  
eccube.event.admin.request  
eccube.event.route.[route].request  
**b. Hookpoint Controller**  
các event của hook point này là  
eccube.event.front.controller  
eccube.event.admin.controller  
eccube.event.route.[route].controller  
Với các hookpoint này ta có thể chèn thêm các xử lý trước khi các controller được gọi đến và sau khi đã rounting đường dẫn. Ta có thể set 1 controller khác khi access vào đường dẫn đã match   
Ví dụ:  
SampleEvent.php

public function onCallController(FilterControllerEvent $event)

{

$controller = new ProductController();

$param = array($controller,'index');

$event->setController($param);

}

Với ví dụ này trước khi 1 controller của frontend được gọi function onCallController sẽ thực thi và gọi function index của controller ProductController.php

**c. Hookpoint Response**  
các event của hookpoint này là:  
eccube.event.front.response  
eccube.event.admin.response  
eccube.event.route.[route].response  
Hookpoint này can thiệp vào xử lý gen ra file html của hệ thống.   
Ví dụ:

public function onRender(FilterResponseEvent $event)  
{  
$request = $event->getRequest();   
$response = $event->getResponse();  
$html = $response->getContent();  
*Xử lý thay đổi từ đây   
$crawler = new Crawler($html); $oldElement = $crawler->filter('#main');  
$oldHtml = $oldElement->html(); $newHtml = $oldHtml . $twig;   
$html = $crawler->html();  
$html = str\_replace($oldHtml, $newHtml, $html);*Kết thúc xử lý thay đổi.  
$response->setContent($html);   
$event->setResponse($response);  
}

**d. Hookpoint exception**  
Các event của hookpoint này:  
eccube.event.front.exception  
eccube.event.admin.exception  
eccube.event.route.[route].exception  
Ví dụ:

public function onCallController(GetResponseForExceptionEvent $event)

{

//Lấy exception

$event->getException();

}

**e. Hookpoint TERMINATE**  
Các event của hookpoint này:  
eccube.event.front.terminate  
eccube.event.admin.terminate  
eccube.event.route.[route].terminate

1. **Tạo page mới từ plugin**

Eccube 3 cho phép tạo 1 route mới từ plugin. Mỗi khi enable plugin function register() của file PluginServiceProvider.php sẽ được kiểm tra và tạo các route giúp thêm đường dẫn mới và page mới cho hệ thống website

Ví dụ:

Config.php

Khi thêm một trang mới,cũng như khi mở rộng một màn hình hoặc một chức năng có sẵn đầu tiên luôn phải mô tả các thông tin của plugin vào file setting cơ bản.   
Trường hợp nào cũng phải tạo file config.yml,sau đó nhập thông tin plugin như name,code,version.  
ví dụ:

name: category content   
code: CategoryContent   
version: 1.0.0  
service:

-{tên file tùy ý}

/{PluginName}/ServiceProvider/{tên file tùy ý}.php sẽ được gọi dựa trên mô tả này

PluginServiceProvider.php

Bằng cách ghi vào register() như dưới đây,ta có thể định nghĩa routing mới.  
Định nghĩa routing tương tự như src/Eccube/ControllerProvider/FrontControllerProvider.php  
Ví dụ:   
Plugin/{PluginName}/ServiceProvider/{ServiceProvider được định nghĩa trong config.yml}.php

public function register(BaseApplication $app)  
{  
$app->match(  
'/sample', *accessURL ‘/Plugin/{PluginName}/SampleController::sample’*tên controller và method)->bind('sample'); // tên routing  
}

Tạo Controller

Plugin/{PluginName}/SampleController.php namespace Plugin¥{PluginName}¥Controller;  
use Eccube¥Application;  
class SampleController  
{  
public function sample(Application $app)  
{  
return ‘Hello, Plugin World!!’;  
}  
}

1. **Tạo menu mới trong admin từ Plugin**

Các option trong admin được lấy từ file ECCUBE3\src\Eccube\Resource\config\nav.yml.dist  
Để thêm 1 option trong admin bằng plugin cũng tương tự như tạo route cho page mới sẽ định nghĩa option mới ở file trong thư mục serviceProvider của plugin  
Ví dụ: tạo 1 option con dưới cùng trong option 受注管理 ở admin

PluginServiceProvider.php

public function register(BaseApplication $app)  
{  
$app['config'] = $app->share($app->extend('config', function ($config) {  
$config['nav'][1]['child'][] = array(  
'id' => 'order\_sample',  
'name' => 'サンプル',  
'url' => 'sample',  
);  
return $config;  
}));  
}

1. **Chỉnh sửa cấu trúc database dùng migration doctrine từ plugin**

Từ Plugin ta có thể tạo bảng thay đổi cấu trúc bảng cấu trúc database bằng cách dùng migration doctrine mỗi khi plugin được enable  
**Chú ý: Sử dụng tiền tố plg\_ trước tên bảng tạo từ plugin**  
Ví dụ tạo bảng mới categroy content bằng plugin. Khi install và enable plugin sẽ tạo thêm trong database 1 bảng category content với đầy đủ tính năng của orm doctrine.  
\*\*1. Tạo cấu trúc plugin  
├── config.yml  
├── Entity  
│ └── CategoryContent.php  
├── event.yml  
├── Migration  
│ └── Version20160218160500.php  
├── PluginManager.php  
├── Repository  
│ └── CategoryContentRepository.php  
├── Resource  
│ ├── doctrine  
│ │ └── Plugin.CategoryContent.Entity.CategoryContent.dcm.yml  
└── ServiceProvider

└── CategoryContentServiceProvider.php

**config.yml**

name: Category Content   
code: CategoryContent   
version: 1.0.0  
service:

CategoryContentServiceProvider

orm.path:

/Resource/doctrine

**PluginManager.php**

<?php  
namespace Plugin\CategoryContent;

use Eccube\Plugin\AbstractPluginManager;

class PluginManager extends AbstractPluginManager  
{

//khi install plugin sẽ chạy function up của file trong Migration

public function install($config, $app)

{

$this->migrationSchema($app, \_\_DIR\_\_.'/Migration', $config['code']);

}

//khi uninstall plugin sẽ chạy function down của file trong Migration

public function uninstall($config, $app)

{

$this->migrationSchema($app, \_\_DIR\_\_.'/Migration', $config['code'], 0);

}

public function enable($config, $app)

{

}

public function disable($config, $app)

{

}

public function update($config, $app)

{

}

}

**CategoryContentServiceProvider.php**

Tạo mapping với repository mới để dễ dàng gọi và sử dụng

<?php  
namespace Plugin\CategoryContent\ServiceProvider;

use Eccube\Application;  
use Silex\Application as BaseApplication;  
use Silex\ServiceProviderInterface;

class CategoryContentServiceProvider implements ServiceProviderInterface  
{

public function register(BaseApplication $app)

{

//Repository

$app['category\_content.repository.category\_content'] = $app->share(function () use ($app) {

return $app['orm.em']->getRepository('Plugin\CategoryContent\Entity\CategoryContent');

});

}

public function boot(BaseApplication $app)

{

}

}

**Version20160218160500.php**

File tạo xử lý thay đổi database

<?php

namespace DoctrineMigrations;

use Doctrine\DBAL\Migrations\AbstractMigration;  
use Doctrine\DBAL\Schema\Schema;

class Version20160218160500 extends AbstractMigration  
{

public function up(Schema $schema)

{

$this->createPluginTable($schema);

}

public function down(Schema $schema)

{

$schema->dropTable('plg\_category\_content');

}

protected function createPluginTable(Schema $schema)

{

$table = $schema->createTable("plg\_category\_content");

$table->addColumn('category\_id', 'integer');

$table->addColumn('content', 'text', array('notnull' => false));

$table->setPrimaryKey(array('category\_id'));

}

}

**CategoryContent.php**

Entity của bảng categoryContent

<?php  
namespace Plugin\CategoryContent\Entity;

class CategoryContent extends \Eccube\Entity\AbstractEntity  
{

private $id;

private $content;

public function getId()

{

return $this->id;

}

public function setId($id)

{

$this->id = $id;

return $this;

}

public function getContent()

{

return $this->content;

}

public function setContent($content)

{

$this->content = $content;

return $this;

}

}

**Plugin.CategoryContent.Entity.CategoryContent.dcm.yml**

File để mô tả thông tin của bảng thêm mới trong database  
Plugin\CategoryContent\Entity\CategoryContent:

type: entity

table: plg\_category\_content

repositoryClass: Plugin\CategoryContent\Repository\CategoryContentRepository

id:

id:

type: integer

nullable: false

unsigned: false

id: true

column: category\_id

generator:

strategy: NONE

fields:

content:

type: text

nullable: true

lifecycleCallbacks: { }

**CategoryContentRepository.php**  
Repository/CategoryContentRepository.php

namespace Plugin\CategoryContent\Repository;

use Doctrine\ORM\EntityRepository;

class CategoryContentRepository extends EntityRepository  
{  
}

Sau khi cài plugin sẽ tạo bảng plg\_category\_content và ta có thể sử dụng repository như các repository có sẵn của eccube 3 cho bảng plg\_category\_content trong database

# CHƯƠNG II. Plugin QuickOrder và ExportCatalogue: