

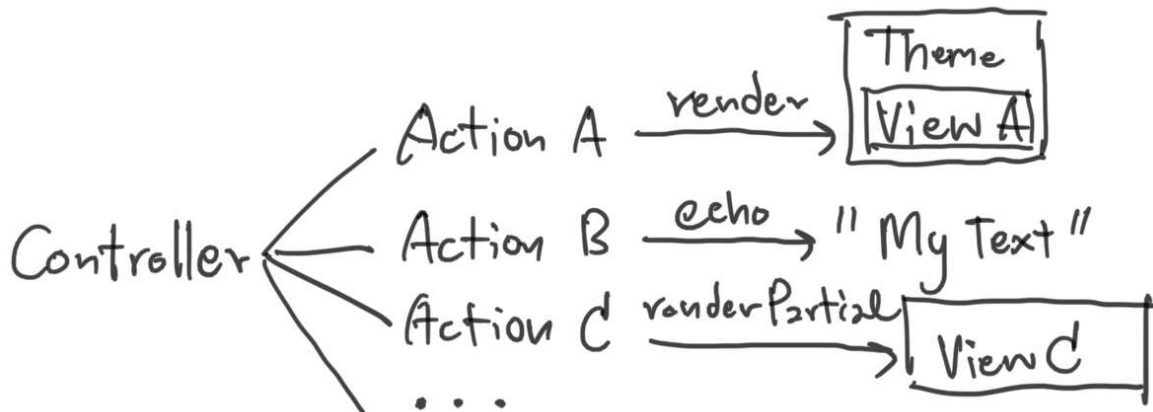
# การพัฒนาระบบด้วย Yii2: ตอนที่ 3 Controller Action และการปรับแต่ง Route เบื้องต้น

Published by **Pongpon Nilaphruek** on August 10, 2022

## ความเดิมตอนที่แล้ว

ใครยังไม่ได้ initail project ให้ไปดูบทความย้อนหลังได้ที่ <https://www.bigdata.rmutt.ac.th/?p=1754>

## เข้าใจเรื่อง Controller และ Action



ความสัมพันธ์ของ Controller และ Action

จากรูปด้านบน ให้เราลองเปิดไฟล์ `frontend\controllers\SiteController.php` ขึ้นมา เพื่อดูหลักการทำงาน สังเกตได้ว่าจะมี function `actionXxx()` อยู่ เช่น `actionIndex()` เป็นต้น

ปกติแล้ว Yii2 ทำการ route โดยใช้ keyword ของ function `actionXxx()` เป็นหลัก (ต้องพิมพ์คำว่า action นำหน้าชื่อของ function) เช่น `actionIndex()` ใน `SiteController` จะถูกเข้าถึงโดย

```
/index.php?r=site/index
```

ถ้าเป็น `index` อยู่แล้ว ผู้ใช้สามารถเข้าถึงได้โดยไม่ต้องใส่ `index` เพิ่มเติมเข้าไป หรือพูดง่าย ๆ ว่า

```
/index.php?r=site/index เทียบเท่า /index.php?r=site
```

```
public function actionB(){
    echo "Hello Yii2";
}
```

ลองเข้า URL ไปที่ `http://localhost/my_app/frontend/web/index.php?r=site/b` เพื่อทดสอบดู จะสังเกตผลลัพธ์ได้ว่า ระบบจะพิมพ์คำว่า Hello Yii2 เป็นข้อความปกติ โดยไม่ได้ render theme ที่เกี่ยวข้องเลย จะสอดคล้องกับภาพด้านบนว่า เราสามารถพิมพ์ text ที่เราต้องการออกมาบนหน้าเว็บของเราได้โดยตรง โดยใช้คำสั่ง echo หรือ print ได้เลย (แต่วิธีนี้ default ของระบบจะให้เรา print ข้อความออกมาไม่ยาวมากนัก ถ้าต้องการ print text ยาว ๆ ให้โยนไปให้ view แสดงผลแทน ซึ่งจะกล่าวต่อไป)

ลองเพิ่ม function actionA() เข้าไปใน SiteController ดังนี้

```
public function actionA(){
    return $this->render("a");
}
```

ลองเข้า URL ที่ `http://localhost/my_app/frontend/web/index.php?r=site/a` ตอนนี้ทุกคนจะต้องได้ error message ตามนี้

```
View not Found - yii\base\ViewNotFoundException
The view file does not exist: C:\xampp\htdocs\my_app\frontend\views\site\a.php
```

จากรูปด้านบนถ้าเราต้องการ render("a") เราจำเป็นต้องมี page PHP รองรับไว้ใน View ด้วย ในที่นี้ก็คือ View ที่อยู่ใน site นั้นเองครับ (มันจะสอดคล้องกัน เพราะเรากำลังเล่นอยู่ครับ SiteController) ดังนั้นให้ไปเพิ่มไฟล์ `frontend\views\site\a.php` และใส่ข้อความอะไรลงไปก็ได้เช่น

```
This is "a" page
```

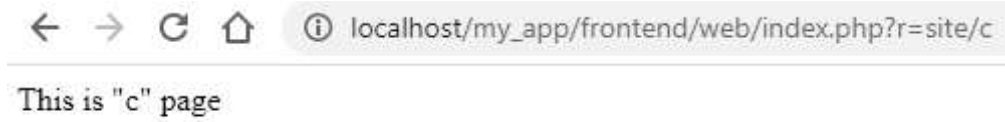
หลังจากนั้นลองรันใหม่ครับ เราจะต้องได้หน้าจอดังรูปด้านล่าง

This is "a" page

ลองทดสอบสร้าง actionC() แต่ใช้วิธีการ render อีกแบบหนึ่งดังนี้ และอย่าลืมไปสร้างไฟล์ `frontend\views\site\c.php` และใส่ข้อความอะไรลงไปก็ได้ด้วยนะครับ

```
public function actionC(){
    return $this->renderPartial("c");
}
```

ถ้าลองทดสอบแล้ว จะพบว่าเราจะเห็นเจตนาของระบบที่เราสังเกตได้ว่า มันจะเหมือน layout theme ของเราที่เกี่ยวข้องเลย เหมือนเราเขียนไฟล์ใหม่สด ๆ ตั้งแต่แรก เพียงแค่ใช้การ route เป็น index.php?r=site/c เท่านั้นครับ ปกติผมจะใช้ในกรณีที่เราไม่อยากไปยุ่งกับ template ใน theme ของเรา เช่น สร้าง JSON API เป็นต้น ก็สามารถทำได้



นอกจากนี้เรายังส่งข้อมูลมาที่ view ที่เราต้องการจาก controller ได้อีกด้วย เช่น return \$this->render('say', ['message' => \$message]); เป็นการส่ง message ไปให้ view ที่ชื่อว่า say เพื่อนำตัวแปร message ไปใช้ต่อได้ ถ้าเราต้องการใช้ใน view (ปกติแล้วจะต้องมีการส่งข้อมูลระหว่าง controller กับ view อยู่แล้วครับ ในสถานการณ์จริง)

## ลองปรับแต่ง Routing ให้สวยงาม โดยทำเป็น Pretty URL

ปกติแล้ว URL ที่ใช้แสดงผล เป็นช่องทางหนึ่งที่ผู้ใช้สามารถเห็นรายละเอียดได้ เช่น เขียนด้วยภาษาอะไร มีตัวแปรใดส่งไปบ้าง เป็นต้น จะเป็นการดีกว่าถ้าเราสามารถซ่อนข้อมูลพวกนี้ได้

Yii2 ได้มีมาตรฐานการทำ Pretty URL ไว้ ซึ่ง Pretty URL คือทำให้ URL นั้นสั้น กระชับ และซ่อนรายละเอียดบางอย่างได้ ปกติแล้ว Yii2 สามารถปรับแต่ง Routing ได้ตามที่เราต้องการ แต่ในที่นี้ผู้เขียนขอใช้มาตรฐานที่มาพร้อมกับ Yii2 เพื่อไม่ให้ dev ปรับแต่งอะไรยุ่งยากมาก สามารถดูข้อมูลเพิ่มเติมได้ที่

<https://www.yiiframework.com/doc/guide/2.0/en/runtime-routing>

พร้อมแล้ว ลุยกันเลย เริ่มต้นเปิดไฟล์ frontend\config\main.php และแก้ไขข้อมูลในส่วน URL Manager ให้เป็นดังนี้

```
'urlManager' => [
    'enablePrettyUrl' => true,
    'showScriptName' => false,
    'rules' => array(
        '<controller:\w+>/<id:\d+>' => '<controller>/view',
        '<controller:\w+>/<action:\w+>/<id:\d+>' => '<controller>/<action>',
        '<controller:\w+>/<action:\w+>' => '<controller>/<action>',
        'module/<module:\w+>/<controller:\w+>/<action:\w+>' => '<module>/<controller>/<action>',
    ),
],
```

ในกรณีที่ เป็น Apache ให้ทำการเพิ่มไฟล์ frontend\web\htaccess (ไฟล์ชื่อว่า .htaccess มี . นำหน้าด้วยนะครับ) และให้เพิ่ม code ต่อไปนี้ลงไป เพื่อปรับการ rewrite rule (โดย default XAMPP จะเปิด mod RewriteRule ให้อยู่แล้ว)

```
RewriteCond %{REQUEST_FILENAME} !-d
# Otherwise forward it to index.php
RewriteRule . index.php
```

เมื่อปรับแล้วให้ลองทดสอบการทำงานดู จะสังเกตว่า URL แบบเต็มเช่น index.php?r=site/a จะถูกเปลี่ยนเป็น /site/a แทน โดย URL เต็ม ๆ ในที่นี้คือ http://localhost/my\_app/frontend/web/site/a ซึ่งทำให้สามารถซ่อนรายละเอียดต่าง ๆ ได้ และมีความกระชับพอสมควร ลองทดสอบไปที่ http://localhost/my\_app/frontend/web/site/about และตำแหน่งอื่น ๆ ดูครับ

ในที่นี้สามารถปรับที่ฝั่ง backend ได้ด้วย ด้วยการทำในลักษณะเดียวกัน แค่เปลี่ยนไปทำใน backend\config\main.php และ backend\web\htaccess แทน หลังจากนั้นลองทดสอบดู

หากใครใช้เป็น Nginx ให้ลองหาข้อมูลด้วย keyword ว่า "Yii2 Pretty URL with Nginx" จะมี community ที่ได้ทำตัวอย่างไว้

ท้ายที่สุดแล้ว ในบทความนี้เราได้รู้หลักการของ Controller Action และการปรับแต่ง Route เบื้องต้น เดี่ยวเราค่อย ๆ ไต่ระดับขึ้นไปครับ

Categories: **DEVELOPMENT** **YII2 DEVELOPMENT**

Tags: Development Web Yii2



## Related Posts

Yii is a fast, secure, and efficient PHP framework.  
Flexible yet pragmatic.  
Works right out of the box.  
Has reasonable defaults.

Get Started

#### DEVELOPMENT

### การพัฒนาระบบด้วย Yii2: ตอนที่ 6 การปรับแต่งธีม (Theming)

ความเดิมตอนที่แล้ว ย้อนดูบทความตอนที่ 5 ได้ที่ <https://www.bigdata.rmutt.ac.th/?p=1908> Theming คือ Theming คือวิธีการในการกำหนดรูปแบบการแสดงผลของระบบภาพรวมของเรา อย่างที่เราทราบว่า ปกติแล้วหน้าระบบเราจะมีส่วนการแสดงผลที่ต้องแชร์กันเสมอ ไม่ว่าจะเป็นส่วนของ head หรือ sidebar หรือ footer ปกติแล้ว Framework ทั่วไป จะแยกการทำงานของ "ธีม" เป็นอีกส่วนหนึ่ง เพื่อให้นักพัฒนาสามารถกำหนดรูปแบบตามที่ต้องการได้อย่างง่ายดาย การปรับแต่งธีม (Theming) Yii2 ได้เตรียมวิธีการให้เราสามารถปรับแต่งธีมได้อย่างหลากหลาย ถ้าพร้อมแล้วสามารถทำตามคลิปด้านล่างได้เลยครับ (ถ้ายังไม่ได้ขึ้นระบบให้ย้อนไปดูบทความก่อนหน้าได้)

Yii is a fast, secure, and efficient PHP framework.  
Flexible yet pragmatic.  
Works right out of the box.  
Has reasonable defaults.

-----  
[Get Started](#)

#### DEVELOPMENT

### การพัฒนาระบบด้วย Yii2: ตอนที่ 5 การทำ migration กับ database

ความเดิมตอนที่แล้ว ย้อนดูบทความตอนที่ 4 ได้ที่ <https://www.bigdata.rmutt.ac.th/?p=1900> Migration คือ Migration ถ้าให้แปลตรง ๆ ตาม dictionary แปลได้ว่า "การโยกย้าย" แต่ถ้าอยู่ใน framework จะเกี่ยวข้องกับ database เช่น การ create/alter table หรือ การ insert/update/delete data ใน table หรือ การกระทำใด ๆ ก็ตามที่เกี่ยวข้องกับระบบ database หรือกล่าวได้อีกนัยหนึ่งว่ากระบวนการทำ migration Read more...

Yii is a fast, secure, and efficient PHP framework.  
Flexible yet pragmatic.  
Works right out of the box.  
Has reasonable defaults.

[Get Started](#)

#### DEVELOPMENT

### การพัฒนาระบบด้วย Yii2: ตอนที่ 4 การใช้ built-in PHP web serve

ความเดิมตอนที่แล้ว ย้อนดูความตอนที่ 3 ได้ที่ <https://www.bigdata.rmutt.ac.th/?p=1845> Built-in PHP web serve การสร้าง web server ด้วย built-in PHP บน Yii2 Framework สามารถทำได้ง่าย และมีประโยชน์ในการทดสอบการทำงานอย่างง่าย และซ่อนรายละเอียดของ path ยาว ๆ ในเครื่องเราได้ (อันที่จริงแล้วตอนนำไปติดตั้งจริงหรือรันในเครื่องเราสามารถตั้งค่า vhost เพื่อ map กับ domain ได้เหมือนกัน) ทั้งนี้เราต้องคำนึงถึง template  
[Read more...](#)

## Links ภายนอก

[ระบบทะเบียนนักศึกษา](#)

[ระบบห้องเรียนออนไลน์ RMUTT D-LEARN](#)

[ห้องสมุดกลาง RMUTT LIBRARY](#)

[สมุดบันทึกกิจกรรม ACTIVITY RMUTT](#)

[กองทุนกู้ยืมเพื่อการศึกษา](#)

Follow Page



Share

