

# การพัฒนาเว็บแอปพลิเคชัน



เล่มที่ 1

CR.

EAKKABIN  
JAIKEAWMA

350.-

# การพัฒนาเว็บแอปพลิเคชัน



เล่มที่ 1

กานต์  
fenics128la

CR.

EAKKABIN  
JAIKEAWMA

## การพัฒนาเว็บแอปพลิเคชันด้วย Yii Framework เล่มที่ 1 (เหมาะสำหรับผู้เริ่มต้น)

Developing Web Application with Yii Framework Volume 1 (Basic)

เขียนโดย เอกบิน ใจแก้วมา (Eakkabin Jaikewama)

### ข้อมูลหนังสือ

เนื้อหา : เหมาะสำหรับผู้เริ่มต้น หรือ ผู้ที่สนใจพัฒนา Yii Framework อีกเช่นเดียวกัน

คำอธิบาย : ทำความรู้จัก PHP MVC, PHP OOP, Yii Framework การตั้งค่าระบบเบื้องต้น การใช้งาน Gii Code Generator การเชื่อมต่อฐานข้อมูล การใช้งานฐานข้อมูล เพิ่ม, ลบ, แก้ไข, บันทึก, ค้นหา การสร้างฟอร์มรับข้อมูล แนะนำเทคนิคที่น่าสนใจต่างๆ การใช้งาน Extension ต่างๆ ทั้งภายในและดาวโหลดมาใช้งานจากภายนอกและสุดท้าย “ตัวอย่าง” ระบบการจำหน่ายสินค้าบนเว็บ E-Commerce เบื้องต้น

จำนวน : 193 หน้า

ราคา 350 บาท

สนใจติดต่อ  
fenics128@gmail.com

### ัดพิมพ์และจำหน่ายโดย

Drive Software and Developer Center

### ข้อมูลติดต่อ

E-Mail : eakkabin\_it@hotmail.com

Facebook : <https://www.facebook.com/drivesoftcenter>

Facebook Group : <https://www.facebook.com/groups/drivesoftcenter/>

Website : <http://www.drivesoftcenter.net>

## คำนำ

ก่อนหนังสือเล่มนี้จะถูกตีพิมพ์ การเรียบเรียงเนื้อหาในหนังสือ จะต้องผ่านกระบวนการการศึกษา การรวบรวมข้อมูล เนื้อหาประกอบกับทดลองฝึกปฏิบัติไปด้วย สำหรับการพัฒนาเว็บแอ��เพลิเคชั่นด้วยภาษา PHP โดยใช้ PHP Framework เป็นตัวขับเคลื่อน ตามโครงสร้าง MVC และการออกแบบ PHP OOP ซึ่งผมก็ได้ทำการศึกษาเรื่องนี้มาเป็นเวลาเกือบ 3 ปี ทำให้มีประสบการณ์ในการแก้ไขปัญหาต่าง ๆ ที่อาจจะเกิด โดยคำสั่งของ PHP OOP, MVC ได้อ่านจากเรื่อง หากรู้ที่มีประสบการณ์ด้านนี้สามารถศึกษา และเรียนรู้การใช้งาน PHP Framework ได้แบบทุกด้าน แต่ ณ ปัจจุบันนี้ PHP Framework ที่เป็นที่น่าจับตา มองก็มีอยู่หลายตัวอย่างเช่น Laravel Framework และ Yii Framework ในหนังสือเล่มนี้เป็นต้น เหตุผลที่เลือกเขียนหนังสือ การพัฒนาเว็บแอ��เพลิเคชั่น ด้วย Yii Framework ขึ้น ก็เพื่อต้องการถ่ายทอดองค์ความรู้ต่าง ๆ ที่ผ่านการศึกษา การเก็บรวบรวมข้อมูลและทดลองปฏิบัติ จนแตกเกิดเป็นไอเดียทำหนังสือเล่มนี้ขึ้นมานั่นเอง

หนังสือ การพัฒนาเว็บแอ��เพลิเคชั่น ด้วย Yii Framework เล่มนี้ จะมีเนื้อหาที่เน้นพื้นฐานแทนทั้งหมดของระบบ Yii Framework ที่มีและจำเป็นต้องทำความรู้จัก เพราะจากประสบการณ์ที่พบเจอนักพัฒนาเว็บแอ��เพลิเคชั่นหลาย ๆ คนมักจะให้ความสำคัญกับคำว่าพื้นฐานน้อยเกินไป ซึ่งจะส่งผลกระทบต่อการแก้ไขปัญหาเมื่อพบเจอระบบ Error แต่หากเราได้ผ่านการเรียนรู้และเข้าใจถึงขั้นตอนกระบวนการทำงาน วิธีการใช้งานเบื้องต้นของระบบ PHP Framework มาพอเข้าใจในระดับหนึ่งแล้ว จะสามารถทำให้เราแก้ไขปัญหาเหล่านี้ลำบากใจ ได้อ่านราวดีรวมมากขึ้น

สิ่งที่จะได้จากหนังสือการพัฒนาเว็บแอ��เพลิเคชั่น ด้วย Yii Framework เล่มที่ 1 นี้ จะเริ่มต้นด้วยการเรียนรู้วิธีการใช้งาน PHP OOP และ PHP MVC พร้อมอธิบายความเกี่ยวข้องของ Yii Framework กับ OOP, MVC ลำดับต่อไปจะเรียนรู้การตั้งค่าระบบ Windows รวมถึงวิธีดาวน์โหลดและติดตั้ง Yii Framework ซึ่งมีขั้นตอนที่ไม่ยุ่งยาก หลังจากที่เราเข้าใจความหมายการดาวน์โหลดการติดตั้งแล้ว บทต่อไปเราจะเริ่มต้นการพัฒนาเว็บแอฟฯ ด้วย Yii Framework ซึ่งจะประกอบไปด้วย การตั้งค่าระบบ Config การใช้งาน Gii System การติดต่อฐานข้อมูล การดึงข้อมูล เพิ่ม, ลบ, แก้ไข, บันทึก, ค้นหา การใช้งานฟอร์ม การติดตั้งและใช้งาน Yii Extension รวมเทคนิคที่น่าสนใจ การใช้งาน Zii Widget และสุดท้ายจะเป็น Workshop ระบบซื้อสินค้าออนไลน์เบื้องต้น

หนังสือเล่มนี้จึงเหมาะสมสำหรับคนที่ต้องการเก็บไว้เป็นเอกสารอ้างอิงประกอบการพัฒนาเว็บแอปพลิเคชัน ด้วย Yii Framework จะช่วยอำนวยความสะดวกแก่ผู้ใช้งานมาก เมื่อเราต้องพนับเงื่อนไขปัญหาในช่วงระหว่างการพัฒนาเว็บแอปฯ ซึ่งจะส่งผลให้เราสามารถแก้ไขปัญหาได้อย่างรวดเร็วและถูกต้องตามหลักการทำงานของ Yii Framework และทำให้เว็บแอปพลิเคชันที่เราสร้างออกไปนั้นมีคุณภาพและเป็นที่พอใจของทุกๆ คน

ผู้เขียนเรียงหัวใจว่า เนื้อหาในเล่มนี้ จะช่วยให้นักพัฒนาศึกษาแนวทางการแก้ไขปัญหาต่างๆ และปฏิบัติได้เป็นอย่างดี

เอกสารนี้  
ผู้เขียน



fenicssoft@gmail.com

# สารบัญ

หัวเรื่อง	หน้า
คำนำ	๑
สารบัญ	๑
บทนำ	๑
บทที่ 1 พื้นฐาน PHP OOP, MVC	
การเขียนโปรแกรมเชิงวัตถุ Object-Oriented Programming	1
แนะนำคำศัพท์สำคัญ (OOP)	1
คลาส (Class)	2
ออบเจกต์ (Object)	2
เมธอด (Method)	3
พร็อพเพอร์ตี้ (Property)	4
ค่าพารามิเตอร์ (Parameter)	4
สแตติกเมธอด (Method Static) และ สแตติกพร็อพเพอร์ตี้ (Property Static)	5
การสืบทอดของออบเจกต์	6
การอ้างอิงออบเจกต์	6
Public, Private, Protected (คีล์วิร์ด)	8
PHP MVC (Model-View-Controller)	8
PHP OOP และ MVC เกี่ยวกับ Yii Framework อ่าย่างไร	10
บทที่ 2 Yii Framework คืออะไร	
ความหมาย Yii Framework	13
พัฒนาและออกแบบตามแนวคิด 3 แบบหลักของ Yii Framework	13
แนะนำส่วน Front Controller	14
กระบวนการทำงานของ Yii Framework	14
ข้อดีของ Yii Framework มีอะไรบ้าง ?	15
ความสามารถของ Yii เมื่อเทียบกับ Framework อื่น ๆ	15
ดาวโหลด Yii Framework v1.1.15 released on Jun 29, 2014	16
เตรียมความพร้อมก่อนการติดตั้ง (เครื่องมือ ตั้งค่าระบบ) ที่สำคัญ ๆ	16

การตั้งค่า Path ของ Windows (จากตัวอย่างเป็น Windows 7)	16
ติดตั้ง Yii Framework v1.1.15 released on Jun 29, 2014	17
แนะนำส่วนประกอบที่สำคัญต่าง ๆ ที่เราต้องรู้	17
ไฟล์เครื่อง assets	17
ไฟล์เครื่อง css	18
ไฟล์เครื่อง images	18
ไฟล์เครื่อง protected	18
ไฟล์เครื่อง themes	18
ตรวจสอบระบบเบื้องต้นหลังการติดตั้ง (requirements) Yii Framework	19
วิธีการใช้งาน PHP PDO extension	20
ดาวน์โหลดไฟล์ติดตั้ง Yii Framework v.1.1.15 (ปรับแต่ง โดย DriveSoftCenter.Net)	20
<b>บทที่ 3 ตั้งค่าระบบก่อนเริ่มพัฒนาเว็บแอ��พลิเคชัน</b>	
การกำหนดชื่อเว็บแอ��พลิเคชันของเรา	21
การทำงานระบบ Gii Code Generator	21
ปิดใช้งานระบบ urlManager จัดการ URL	22
สร้างไฟล์ .htaccess	23
การตั้งค่าการเชื่อมต่อฐานข้อมูล	24
กำหนด default ภาษาที่ใช้แสดงเริ่มต้น	24
กำหนด TimeZone ให้กับเว็บแอ��พลิเคชัน	25
ตั้งค่า Messages และคงข้อความเดือนเป็นภาษาไทย	25
<b>บทที่ 4 วิธีการใช้งาน Gii Code Generator</b>	
ตรวจสอบระบบ Gii เบื้องต้น	27
ส่วนที่ 1 Controller Generator	28
การกำหนดสิทธิ์การเข้าถึงภายใต้ไฟล์ Controller	29
ฟังก์ชัน accessRules()	30
ตัวอย่าง การกำหนดสิทธิ์การเข้าใช้งาน Controller เบื้องต้น	30
ส่วนที่ 2 Crud Generator	32
ส่วนที่ 3 Form Generator	33
ส่วนที่ 4 Model Generator	34
กำหนด Attribute ให้กับเมช.od Rules ในโมเดล	34
ส่วนที่ 5 Module Generator	36

<b>บทที่ 5 การเชื่อมต่อฐานข้อมูล</b>	
การเชื่อมต่อฐานข้อมูล MySQL	39
การเชื่อมต่อฐานข้อมูลต่าง ๆ นอกเหนือจาก MySQL	39
การเชื่อมต่อฐานข้อมูลแบบ Multiple-database	40
การดึงข้อมูลจากฐานข้อมูล ด้วยวิธีการแบบต่าง ๆ	41
Data Access Object (DAO) : CSqlDataProvider	41
การจัดการ Transaction เป็นอย่างดี	42
JOIN Table : Data Access Object (DAO)	44
คำสั่ง MySQL : Left Join	44
คำสั่ง MySQL : Right Join	44
คำสั่ง MySQL : Inner Join	44
Active Record (AR) : CActiveDataProvider	44
Join Model : Active Record (AR)	46
การสร้างความสัมพันธ์ (relations) ระหว่าง Model ด้วยกันเอง	46
การเขียนความสัมพันธ์ เพื่อเรียก โภง โนเมเดล เข้าด้วยกัน	48
โดยยินดีของความสัมพันธ์	48
ความสัมพันธ์แบบ HAS_ONE	49
ความสัมพันธ์แบบ BELONGS_TO	49
ความสัมพันธ์แบบ HAS_MANY	50
ความสัมพันธ์แบบ MANY_MANY	50
Active Record (AR) : การดึงข้อมูลแบบชาร์มดา	51
การดึงข้อมูลจากฐานข้อมูล	51
การ Insert ข้อมูล	51
การ Update ข้อมูล	52
การ Delete ข้อมูล	52
Array Data (AD) : CArrayDataProvider	52
<b>บทที่ 6 การใช้งาน Active Form</b>	
Form แบบ Widget CActiveForm	53
คำสั่ง : errorSummary	53
คำสั่ง : error	54
คำสั่ง : labelEx	54

คำสั่ง : textField	54
คำสั่ง : textArea	54
คำสั่ง : passwordField	55
คำสั่ง : numberField	55
คำสั่ง : telField	55
คำสั่ง : fileField	55
คำสั่ง : dateField	56
คำสั่ง : timeField	56
คำสั่ง : dropDownList	56
คำสั่ง : emailField	56
คำสั่ง : urlField	57
คำสั่ง : rangeField	57
คำสั่ง : hiddenField	57
คำสั่ง : checkBox	57
คำสั่ง : checkBoxList	58
คำสั่ง : radioButton	58
คำสั่ง : radioButtonList	58
คำสั่ง : submitButton	58
คำสั่ง : listBox	59
คำสั่ง : resetButton	59
คำสั่ง : ajaxButton	59
คำสั่ง : ajaxSubmitButton	60
สรุปการสร้างฟอร์มแบบ CActiveForm	60
<b>บทที่ 7 ติดตั้งและใช้งาน Extension</b>	
การดาวน์โหลด Extension	61
การติดตั้ง Extension	61
การตั้งค่าระบบ (หลังติดตั้ง Extensions)	62
การเรียกใช้งาน Extension	62
แนะนำ Extension ที่จำเป็นและน่าสนใจ	63
Yii Booster version 4.0.1	63
ดาวน์โหลด Yii Booster v.4.0.1	63

ดาวโหลด Twitter Bootstrap Theme v.3.1.1	64
ตั้งค่าการเชื่อมต่อ Yii Booster v.4.0.1	64
Yii User Modules v.0.3.61	66
ดาวโหลด Yii User Modules v.0.3.61	66
นำเข้า Database (.SQL)	67
การใช้งาน Yii User Extension	68
Yii Rights Modules v.1.3.0	71
ตั้งค่า Yii Rights Modules	71
นำเข้า Database (.SQL)	72
วิธีการใช้งาน Yii Rights Modules ร่วมกับ Yii User Modules	73
กำหนด Permissions ให้กับ User	76
การเขต Role	76
การกำหนด Rule ของ User	77
บทที่ 8 รวมเทคนิคต่าง ๆ ที่น่าสนใจ	
การแทรกคำสั่ง CSS JavaScript	79
ตำแหน่งไฟล์ การแทรกคำสั่ง CSS JavaScript	79
นำเข้าไฟล์ JavaScript ของ Yii Framework	79
นำเข้าไฟล์ CSS, JavaScript จาก Widget	80
การแทรกคำสั่ง MetaTag	81
การแสดงผล “รูปภาพ” ด้วยคำสั่ง CHtml::image()	81
การแสดงผล “ลิงก์” ด้วยคำสั่ง CHtml::link()	82
การแสดงผล “ลิงก์” ด้วยคำสั่ง CHtml::ajaxLink()	82
การแสดงผล “ปุ่ม” ด้วยคำสั่ง CHtml::button()	83
การแสดงผล “รายการข้อมูล” ด้วยคำสั่ง CHtml::listData()	83
การใช้งาน CCaptcha Widget	84
การใช้งาน Session	85
การใช้งาน Cookie	86
สร้างฟังก์ชันสำหรับใช้งาน Cookie	87
การ redirect สำหรับ Yii Framework	88
การ render สำหรับ Yii Framework	89
ระบบ Log ของ Yii Framework	90

สร้างคลาส (Class) ไว้ใช้งาน แบบเรียกผ่าน คำสั่ง Yiiapp();	92
สร้างเว็บแอพพลิเคชัน รองรับหลายภาษา (ไทย, อังกฤษ, จีน เป็นต้น)	93
สร้างเมนู “ลดลงภาษา”	94
สร้าง Module เพื่อแยกระบบการทำงานออกจากระบบหลัก	97
เพิ่ม,ลบ,แก้ไข ไฟล์รูปภาพ ด้วยฟังก์ชัน CUploadFile	103
รู้ก่อน ป้องกันก่อน ขั้นตอนการอัพโหลดไฟล์ขึ้น Hosting	106
สิ่งที่ต้องทำหลังการอัพโหลดไฟล์ขึ้น Server เรียบร้อยแล้ว	108
<b>บทที่ 9 การใช้งาน Zii Widget</b>	
ตัวอย่าง : Widget CBreadcrumbs	109
ตัวอย่าง : Widget CDetailView	109
ตัวอย่าง : Widget Cmenu	110
ตัวอย่าง : Widget CGridView	110
ตัวอย่าง : Widget CJuiAccordion	111
ตัวอย่าง : Widget CJuiAutoComplete	111
ตัวอย่าง : Widget CJuiButton	112
ตัวอย่าง : Widget CJuiDatePicker	113
ตัวอย่าง : Widget CJuiDialog	113
ตัวอย่าง : Widget CJuiProgressBar	114
ตัวอย่าง : Widget CJuiSlider	114
ตัวอย่าง : Widget CJuiSliderInput	114
ตัวอย่าง : Widget CJuiTabs	115
<b>บทที่ 10 ตัวอย่างกรณีศึกษาระบบซื้อขายสินค้าออนไลน์พื้นฐาน</b>	
อธิบายขั้นตอนการทำงานของระบบ	117
จัดเตรียมเครื่องมือก่อนเริ่มพัฒนาระบบ	117
สร้างเว็บแอพพลิเคชัน เจียนคำสั่ง yiic command	118
ตั้งค่าระบบ	118
ดาวโหลด Yii Booster version 4.0.1	120
ดาวโหลด Twitter Bootstrap Theme v.3.1.1	120
ดาวโหลด SEO Extension v.0.9.2	121
ติดตั้ง SEO Extension v.0.9.2	121
สร้างโมเดลข้อมูลด้วย Gii System	121

ติดตั้ง Yii User Extension v.0.3.61	122
สร้าง Module ชื่อว่า “backoffice”	122
ดาวโหลด รูปภาพสินค้า	122
ดาวโหลดไฟล์ Theme + Layoutst ทั้งหมด	123
ระบบสั่งซื้อสินค้า	132
ระบบการแจ้งชำระเงิน	134
ระบบการติดต่อทางร้าน	135
ระบบจัดการข้อมูลสินค้า ข้อมูลการตั้งข้อ การชำระเงิน และข้อมูลการติดต่อ	135
การจัดการข้อมูล ด้วย Gii Bootstrap Generator	136
<b>ภาคผนวก ก</b>	
คู่มือการใช้งาน Yii Booster v.4.0.1	139
ตัวอย่าง : Widget Alert	140
ตัวอย่าง : Widget Badge	141
ตัวอย่าง : Widget Label	141
ตัวอย่าง : Widget Progress	142
ตัวอย่าง : Widget Editable Field	142
ตัวอย่าง : Widget Editable Detail View	142
ตัวอย่าง : Widget Editable Column	143
ตัวอย่าง : Widget TbHighCharts (Basic)	144
ตัวอย่าง : Widget TbHighCharts (Zeros in data)	144
ตัวอย่าง : Widget TbHighCharts (Javascript callback in options)	145
ตัวอย่าง : Widget TbHighCharts (Official demo)	146
ตัวอย่าง : Widget Active Form (Vertical)	147
ตัวอย่าง : Widget Active Form (Inline Form)	147
ตัวอย่าง : Widget Active Form (Horizontal Form)	148
ตัวอย่าง : Widget Active Form (Inputs showcase)	148
ตัวอย่าง : Widget Button	156
ตัวอย่าง : Widget Button Group	157
ตัวอย่าง : Widget Switch	161
ตัวอย่าง : Widget Select2	161
ตัวอย่าง : Widget Typeahead input	162

ตัวอย่าง : Widget Time Picker	163
ตัวอย่าง : Widget Date Picker	163
ตัวอย่าง : Widget Grid View	164
ตัวอย่าง : Widget Panel	165
ตัวอย่าง : Widget Carousel	167
ตัวอย่าง : Widget Collapse	168
ตัวอย่าง : Widget Detail View	169
ตัวอย่าง : Widget Tabs	169
ตัวอย่าง : Widget Thumbnails	170
ตัวอย่าง : Widget Wizard	171
ตัวอย่าง : Widget Modals	171
ตัวอย่าง : Widget Breadcrumbs	172
ตัวอย่าง : Widget Menu	173
ตัวอย่าง : Widget Navigation Bar	174



# บทนำ

ในการพัฒนาเว็บแอปพลิเคชัน ด้วยภาษา PHP Framework ส่วนใหญ่แล้ว เราต้องรู้จักคำว่า PHP OOP (Object-Oriented Programming) และ PHP MVC (Model-View-Controller) กันจะก่อน ซึ่งอาจจะอธิบายในทราบในบทนี้ แต่ก่อนจะไปในหัวข้อนี้ เรายังรู้ความหมายของการพัฒนาเว็บแอปพลิเคชันที่ออกแบบ 4 แบบ ดังต่อไปนี้

1) การเขียนแบบธรรมชาติ (**PHP Basic**) พื้นฐาน โดยจะเปียนตามพื้นฐานของภาษา PHP ธรรมชาติ การสร้างฟังก์ชันขึ้นมาใช้งาน หรือจะเรียกแบบบ้าน ๆ ก็คือ “วิธีการเขียนสด ๆ”  **ทดสอบการใช้งานเครื่องมือ**

2) การเขียนแบบ OOP (Object-Oriented Programming) คือ การเรียกใช้ออบเจกต์คลาส (Object Class) หรือเมธอด (Method) เพื่อการทำงานช้า ๆ กัน ได้แก่ ฟังก์ชันเดียว ยกตัวอย่างเช่น ฟังก์ชัน แปลงวัน ที่จาก ‘2013-12-01’ มาเป็นวันที่แบบไทย ‘1 ธันวาคม 2556’ แบบนี้เป็นต้น

3) การเขียนแบบใช้ PHP Framework เข้ามายื่นในการพัฒนาเว็บแอปฯ โดย Framework ต่าง ๆ เหล่านี้ได้ถูกพัฒนาจากกลุ่มนักพัฒนาซอฟแวร์รวมตัวกันพัฒนาขึ้นมา เพื่อช่วยให้การพัฒนาเว็บแอปพลิเคชันนั้นเสร็จได้อย่างรวดเร็ว โดยที่เราไม่ต้องไปสร้างฟังก์ชันอะไรมากมาย เพราส่วนใหญ่แล้ว PHP Framework จะสร้างฟังก์ชันที่จำเป็นเตรียมไว้ให้เราเรียกใช้งานแล้ว หรือเราเองก็ยังสามารถสร้างฟังก์ชันใหม่ขึ้นมาตามที่เราต้องการก็ได้อีกหนึ่งกัน ซึ่ง PHP Framework นี้ จะอยู่ในรูปแบบของ Code PHP ซึ่งหมายความว่ารับนักพัฒนาเว็บแอปฯ ที่ชอบการเขียน Code โดยตรง

4) การเขียนโปรแกรมแบบใช้ CMS Framework เข้ามายื่น CMS ย่อมาจาก Content Manager System หรือ ระบบบริหารจัดการเนื้อหาเว็บไซต์ ซึ่ง CMS นี้ ส่วนใหญ่แล้ว หมายความว่ารับนักพัฒนาเว็บแอปฯ ที่ไม่ค่อยถนัดเรื่องเขียน Code โดยตรง ซึ่งทาง CMS Framework เองก็ได้ถูกพัฒนาขึ้น เพื่อตอบสนองให้แก่บุคคลธรรมชาติทั่วไป ที่ต้องการอยากมีเว็บไซต์เป็นของตนเอง ซึ่งไม่จำเป็นต้องเกยรีบเรียงเขียนโปรแกรมมาก่อน ก็สามารถสร้างเว็บแอปพลิเคชัน เป็นของตนเองได้ ต้องการแค่เพียงการศึกษาขั้นตอนการใช้งานเครื่องมือต่าง ๆ ผ่าน คู่มือ CMS Framework ที่เราเลือกใช้นั้น ๆ

## โครงหน้าที่จะอ่านหนังสือเล่มนี้

หมายความว่ารับนักพัฒนาที่ไม่ที่กำลังสนใจการพัฒนาเว็บแอปพลิเคชัน ด้วย Yii Framework ซึ่งหนังสือเล่มนี้ จะมอบสิ่งที่ต้องการให้ Yii Framework คือและคู่มือการใช้งานระบบต่าง ๆ ได้จากเนื้อหาในหนังสือเล่มนี้

เนื้อหาที่หนังสือเล่มนี้ครอบคลุม

ต่อไปจะเป็นเนื้อหาที่มีทั้งหมดในหนังสือเล่มนี้

บทที่ 1 พื้นฐาน PHP OOP, MVC เพื่อเข้าใจกระบวนการทำงาน Yii Framework

บทนี้เราจะเรียนรู้ พื้นฐาน PHP OOP, MVC เพื่อเข้าใจกระบวนการทำงาน Yii Framework ทั้งหมด

บทที่ 2 Yii Framework กืออะไร

บทนี้เราจะทำความเข้าใจ Yii Framework และความสามารถต่าง ๆ

บทที่ 3 ตั้งค่าระบบก่อนเริ่มพัฒนาเว็บแอปพลิเคชัน

บทนี้เราจะมาเรียนรู้การตั้งค่าระบบ ก่อนเริ่มต้นพัฒนาระบบ เช่น เปิดระบบ Gii, เชื่อมต่อฐานข้อมูล

บทที่ 4 วิธีการใช้งาน Gii Code Generator

บทนี้เราจะศึกษาวิธีการใช้งานระบบ Gii Code Generator

บทที่ 5 การเชื่อมต่อฐานข้อมูล พร้อมวิธีใช้งานฐานข้อมูลแบบต่าง ๆ

บทนี้เราจะศึกษาวิธีการเชื่อมต่อฐานข้อมูล พร้อมวิธีใช้งานฐานข้อมูลแบบต่าง ๆ

บทที่ 6 การใช้งาน Active Form พร้อมกรณีตัวอย่าง

บทนี้เราจะศึกษาการใช้งาน Active Form พร้อมรูปภาพและคำสั่ง กรณีตัวอย่าง

บทที่ 7 ติดตั้งและใช้งาน Extension สำหรับ Yii framework

บทนี้เราจะมาศึกษาวิธีการติดตั้งและใช้งาน Extension สำหรับ Yii framework

บทที่ 8 รวมเทคนิคต่าง ๆ ที่น่าสนใจ

บทนี้เราจะมาศึกษาเทคนิคต่าง ๆ ที่น่าสนใจ เพื่อตกแต่งเว็บแอปพลิเคชันให้มีประสิทธิภาพมากขึ้น

บทที่ 9 การใช้งาน Zii Widget By Yii Framework

บทนี้เราจะมีศึกษาวิชาการใช้งาน Zii Widget By Yii Framework

บทที่ 10 ตัวอย่างกรณีศึกษาระบบที่ขายสินค้าออนไลน์ พื้นฐาน

บทนี้เราจะศึกษาขั้นตอนกระบวนการพัฒนาระบบที่ขายสินค้าออนไลน์ พื้นฐาน

ภาคผนวก ก

บทนี้เราจะเรียนรู้คู่มือการใช้งาน Yii Booster v.4.0.1

# การพัฒนาเว็บแอปพลิเคชัน ด้วย Yii Framework

## บทที่ 1 พื้นฐาน PHP OOP, MVC



### การเขียนโปรแกรมเชิงวัตถุ Object-Oriented Programming :: OOP

การเขียนโปรแกรมเชิงประวัตถุ จะเป็นการเขียนโปรแกรมในรูปแบบหนึ่งที่กำลังมาแรง แห่งทางโค้ด ณ. เวลานะนี้ คำว่า “เชิงวัตถุ” จะหมายถึง ความสามารถที่ได้นำเอาฟังก์ชันและตัวแปร ต่าง ๆ ทั้งหมด มาพนักความสัมพันธ์เข้าหากันและมองลึกลงไปให้อยู่ในรูปแบบของ “օบเจกต์ (Object)” หรือ “ตัวแทน” ซึ่งมันจะหมายความว่า แทนที่เราจะต้องเขียน “ฟังก์ชัน (Function)” ต่าง ๆ มากมาย เก็บไว้ตามไฟล์หรือแยกไฟล์เดอร์เก็บฟังก์ชันนั้นไว้ เพื่อแบ่งแยกตามความสามารถของแต่ฟังก์ชันไว้ วิธีแก้ไขปัญหาดังกล่าวข้างต้นนี้ เราสามารถใช้วิธีการรวมฟังก์ชันและตัวแปรต่าง ๆ ให้อยู่ในรูปของ对象 (Object) เดียว กันได้ เพื่อกำหนดความสามารถของobject ว่าสามารถทำอะไรได้บ้าง สรุปง่าย ๆ ก็คือ “การแบ่งแยกการทำงานเฉพาะของobject” นั่นเอง

#### แนะนำคำศัพท์สำคัญ (OOP)

คำศัพท์ที่เราจะได้เห็นต่อไปนี้ ล้วนแต่เป็นคำศัพท์ทั่วไปที่เราต้องจดจำและนำไปใช้ ซึ่งมันจะได้เห็นอยู่ตลอดระยะเวลาการเขียนโปรแกรมแบบ OOP นี้ คำศัพท์ที่จะแนะนำมีทั้งหมด 5 คำศัพท์ ดังนี้

**คลาส (Class) :** เป็นแบบจำลองของชุดคำสั่งในการสร้างของobject ซึ่งในคลาสจะประกอบไปด้วยเมธอดหรืออพเพอร์ตี้ 用来以供使用 หรือใช้สำหรับการสร้างของobject

**օบเจกต์ (Object) :** เป็นตัวแทนที่ได้หลังจากการสร้างของobject คลาสจากคลาสสำเร็จหรือเป็นตัวคลาสในการเชื่อมต่อ กับ เมธอด และพร็อพเพอร์ตี้ ที่อยู่ภายในคลาส (Class) นั้น ๆ

**เมธอด (Method) :** ฟังก์ชันต่าง ๆ ที่อยู่ในคลาส แต่เมื่อถูกเปลี่ยนเป็นของobjectแล้วจะเรียกว่าเมธอด

**พร็อพเพอร์ตี้ (Property) :** เป็นตัวแปรของของobject หรือตัวแปรที่อยู่ภายในคลาสที่ถูกสร้างเป็นของobject

**พารามิเตอร์ (Parameter) :** เป็นค่าของตัวแปรที่ถูกส่งเข้าไปในตัวของobject เพื่อใช้กำหนดค่าให้กับเมธอดหรือพร็อพเพอร์ตี้ ที่อยู่ภายในของobject นั้น ๆ

## คลาส (Class)

แม่แบบของชุดคำสั่งในการสร้างออบเจกต์ ซึ่งในคลาสจะประกอบไปด้วยเมธอด หรือพร็อปเพอร์ตี้ เพื่อใช้สำหรับการสร้างออบเจกต์

## การประกาศคลาส

```
// คลาส Datetime
class Datetime {
    public $date; // พร็อปเพอร์ตี้ data
    // เมธอด thai_datetime()
    public function thai_datetime($datetime) {
        /* คำสั่งแปลงวันที่เป็นภาษาไทย */
        $this->date = $datetime;
        return $this->$date; // ส่งค่าพารามิเตอร์ date กลับออกไประหว่างคำสั่ง
    }
}
```

Class

Function ที่อยู่ใน Class  
จะเรียก Method

## จากตัวอย่าง :

เราต้องการสร้าง คลาสชื่อว่า “Datetime” เพื่อใช้แปลงวันที่จาก ‘2013-12-01’ มาเป็นวันที่แบบไทย ๆ เช่น ‘1 ธันวาคม 2556’ ซึ่งในคลาสนี้ จะประกอบไปด้วย พร็อปเพอร์ตี้ 1 ตัว คือ \$date และเมธอด 1 ตัว คือ thai\_datetime(\$datetime) พร้อมสามารถรับค่าพารามิเตอร์ \$datetime เข้าไปในเมธอด thai\_datetime() ด้วย

สังเกต ให้มาระบุว่าเราจะมีคำสั่งพิเศษอยู่ 1 คำสั่งนั้นก็คือคำสั่ง \$this อยู่ภายในคลาสด้วย ซึ่งมันก็หมายถึง คำสั่งที่มีการอ้างอิงตัวแปรหรือฟังก์ชันใด ๆ ก็ตามที่อยู่ภายในคลาสนี้และสุดท้ายเราจะสังเกตเห็นคำสั่งอีก 1 คำสั่งคือ return ซึ่งคำสั่งนี้คือการคืนค่าใด ๆ ออกมาจากเมธอด มันจะมีประโยชน์ตรงที่ เราจะสามารถนำค่าที่ถูก return ค่ากลับมานานั้นไปทำอย่างอื่นได้ต่อไป

## ออบเจกต์ (Object)

เป็นตัวแทนที่ได้หลังจากการสร้างออบเจกต์จากคลาสสำเร็จหรือเป็นตัวกลางในการเชื่อมต่อกับเมธอดและพร็อปเพอร์ตี้อยู่ภายในคลาส (Class) นั้น ๆ

## การสร้างออบเจกต์

```
$objDate = new Datetime(); // การสร้างออบเจกต์ คือ พิมพ์คำสั่ง new แล้วตามด้วยชื่อ คลาส
$date = $objDate->thai_datetime('2013-12-01'); // ส่งค่าพารามิเตอร์ที่ ให้กับเมธอด thai_datetime()
```

## การใช้ออบเจกต์

```
$objDate = new Datetime(); // ตัวแปร $objDate รับค่าจาก ออบเจกต์ Datetime
// ส่งค่าพารามิเตอร์ '2013-12-01' ไปยังเมธอดและเก็บค่าที่คืนกลับมาไว้ในตัวแปร $date
$date = $objDate->thai_datetime('2013-12-01');
echo "วันนี้คือ :: วัน".$date; // แสดงผลลัพธ์ $date
// output :: วันนี้คือ :: วันอาทิตย์ที่ 01 ธันวาคม พ.ศ. 2556 // ผลลัพธ์ที่ได้
$objDate->date = '2013-12-01' // ตัวอย่าง การกำหนดค่าให้กับ พร็อปเพอร์ตี้ ภายในออบเจกต์
$objDate->thai_datetime('2013-12-01'); // การเรียกใช้เมธอด
```

### จากตัวอย่าง :

เราได้สร้างตัวแปร \$objDate ขึ้นมา เพื่อเก็บค่าของออบเจกต์ Datetime() เอาไว้ จากตัวอย่างเราจะเห็นว่าการเรียกใช้งานออบเจกต์นั้นทำได้ง่ายมาก ๆ แค่ พิมพ์คำสั่ง new แล้วตามด้วยชื่อคลาส เราเก็บไว้ได้ออบเจกต์แล้ว ตามตัวอย่างคือ new Datetime() เพียงเท่านี้เราเก็บไว้ได้ออบเจกต์ไปใช้งานแล้ว

เราสามารถเรียกใช้งาน เมธอดหรือพร็อปเพอร์ตี้ต่าง ๆ ที่อยู่ภายใต้คลาสที่ถูกสร้างเป็นออบเจกต์ไว้ แล้ว และ จากตัวอย่าง ผู้ต้องการส่งค่าพารามิเตอร์เข้าไปผ่านเมธอด โดยกำหนดดังนี้ thai\_datetime('2013-12-01') หรือหากต้องการส่งค่าพารามิเตอร์เข้าไปยังพร็อปเพอร์ตี้สามารถทำได้ เช่นเดียวกัน ก็คือ \$objDate->date = '2013-12-01' เมื่อเมธอดได้รับค่าพารามิเตอร์ที่ถูกส่งเข้าไปแล้ว กระบวนการทำงานของเมธอดก็จะเริ่มทำงานใช้ในกรณีที่เมธอดนั้นได้มีการกำหนดให้สามารถรับค่าพารามิเตอร์จากภายนอกเข้ามาได้

โดยตัวอย่างผมสร้างคลาส Datetime เพื่อใช้แปลงวันที่ เป็นวันที่ภาษาไทย และเมื่อเมธอดทำงานเสร็จ เมธอดก็จะ return ค่าวันที่เป็นภาษาไทยกลับออกมานะ ด้วยคำสั่ง return นั่นเอง สุดท้ายก็ส่งให้มันแสดงผลลัพธ์ออกมา โดยใช้คำสั่ง echo ตามด้วยชื่อตัวแปร \$date (ตามตัวอย่าง) ผลลัพธ์ที่ได้ก็คือ วันอาทิตย์ที่ 01 ธันวาคม พ.ศ. 2556 เป็นต้น

เพิ่มเติม เครื่องหมายคำสั่งพิเศษที่เราเห็นได้จากตัวอย่างข้างบนนี้ เราจะสังเกตเห็นเครื่องคำสั่ง -> ซึ่งสำหรับภาษา PHP แล้ว เราเรียกมันว่า “ออบเจกต์ โอเปอเรเตอร์ (Object operator)” ซึ่งก็คือคำสั่งที่ใช้ระบุการเชื่อมต่อระหว่าง ออบเจกต์ กับ พร็อปเพอร์ตี้ หรือ ออบเจกต์ กับ เมธอด ที่อยู่ในออบเจกต์นั้น ๆ

### เมธอด (Method)

ฟังก์ชันต่าง ๆ ที่อยู่ในคลาส แต่เมื่อถูกเปลี่ยนเป็นออบเจกต์แล้วเราจะเรียกว่าเมธอด (Method) แทน

```
class Datetime { // คลาส Datetime
    public function thai_datetime($datetime) { // thai_datetime() และรับค่าพารามิเตอร์ $datetime
        /* คำสั่งแปลงวันที่เป็นภาษาไทย */
        return $date; // ส่งค่าพร็อปเพอร์ตี้ $date กลับออกไป
    }
}
```

```
// การเรียกใช้งาน
```

```
$objDate = new Datetime(); // สร้างตัวแปร $objDate ไว้เก็บค่าอ่อนเจ็กต์ Datetime()  
$objDate->thai_datetime('2013-12-01'); // ส่งค่าพารามิเตอร์ '2013-12-01' เข้าไปยัง thai_datetime
```

## พร็อปเพอร์ตี้ (Property)

ตัวแปรที่เป็นของอ่อนเจ็กต์ หรือ ตัวแปรที่อยู่ภายใต้คลาสที่ถูกสร้างเป็นอ่อนเจ็กต์

ตัวอย่าง : การใช้งาน พร็อปเพอร์ตี้ (Property)

```
// สร้างคลาส
```

```
class Datetime { // คลาส Datetime()  
    public $date; // พร็อปเพอร์ตี้ date  
    public function thai_datetime($datetime) {  
        return true;  
    }  
}
```

```
// การเรียกใช้งาน
```

```
$objDate = new Datetime(); // สร้างอ่อนเจ็กต์ Datetime() และบันทึกค่าอ่อนเจ็กต์ไว้ใน $objDate  
$objDate->date = '2013-12-01'; // กำหนดค่าให้กับ พร็อปเพอร์ตี้ date  
echo $objDate->date; // แสดงผลลัพธ์ออกมา
```

## ค่าพารามิเตอร์ (Parameter)

ค่าพารามิเตอร์ คือ ค่าที่มีการส่งผ่านตัวอ่อนเจ็กต์ไปยังเมธอดที่อยู่ภายใต้อ่อนเจ็กต์ ในหนึ่งเมธอดสามารถรับค่าพารามิเตอร์ได้ทั้งหลาย ๆ พารามิเตอร์ โดยใช้เครื่องหมาย “,” คั่นระหว่างพารามิเตอร์ตัวที่ 1 และค่าพารามิเตอร์ตัวที่ 2 ในส่วนเมธอดของเรายังสามารถกำหนดค่าเริ่มต้นหรือค่ามาตรฐานให้กับค่าพารามิเตอร์ได้ ดังตัวอย่าง การกำหนดค่าพารามิเตอร์ที่หลาย ๆ ตัว

```
// รูปแบบ
```

```
public function date('ค่าพารามิเตอร์1','ค่าพารามิเตอร์2','ค่าพารามิเตอร์3'){ ..... }
```

```
// ตัวอย่าง
```

```
public function date($type,$date,$time,...){.....}
```

## เพิ่มเติม :

การกำหนดค่าเริ่มต้นให้กับพารามิเตอร์ เราจะสามารถทำได้เช่นกัน เมื่อจากเราจะต้องมีการป้องกัน กรณีที่เราลืมกำหนดค่าให้กับพารามิเตอร์ มันอาจจะส่งผลกระทบต่อการทำงานของเมธอดได้ เพราะการทำงานของเมธอดบางเมธอดนั้น จะไม่สามารถทำงานได้ด้วยตัวเองได้ แต่ต้องมีการรับค่าตัวแปร พารามิเตอร์ เข้ามา เพื่อเริ่มกระบวนการทำงานได้ ดังเช่นตัวอย่างที่เราได้ศึกษาในบทนี้มา เมธอด thai\_datetime(\$datetime) ต้องมีรับค่าพารามิเตอร์ ซึ่งมันก็คือ “วันที่” เพื่อใช้ในการแปลงจาก วันที่ ‘2013-12-01’ มาเป็น “วันอาทิตย์ที่ 01 ธันวาคม พ.ศ. 2556”

## 静态ติกเมธอด (Method Static) และ สเตติกพร็อปเพอร์ตี้ (Property Static)

หลังจากที่เราได้ทำการรู้จักกับคำศัพท์เมธอดและพร็อปเพอร์ตี้แล้ว ล้วนตัวไปเราควรทำความรู้จักความหมายของคำว่า “static” กันต่อ ซึ่งคำว่า static นั้นจะหมายถึง การเรียกใช้งาน พร็อปเพอร์ตี้หรือเมธอดใด ๆ ก็ได้ โดยที่เราไม่ต้องสร้างออบเจกต์ขึ้นมา ก่อน หลักการสร้าง พร็อปเพอร์ตี้และเมธอดแบบ static ก็คือ มีหลักการง่ายมาก ๆ เพียงแค่เราว่างคำว่า static ไว้หน้าชื่อพร็อปเพอร์ตี้หรือฟังก์ชันที่เราต้องการได้เลย ยกตัวอย่างเช่น เมธอด public static function thai\_datetime() และหากเป็นพร็อปเพอร์ตี้จะต้องทำไป เก็บค่าสั่งตามนี้เลย public static \$date เป็นต้น

ตัวอย่าง : บังส่วนของคลาส Datetime();

```
// คลาส Datetime
class Datetime {
    public static $date; // เพิ่มคำสั่ง static ไว้หน้าชื่อ พร็อปเพอร์ตี้
    public static function thai_datetime($datetime) { // เพิ่มคำสั่ง static ไว้หน้าชื่อ เมธอด
        /* คำสั่งแปลงวันที่เป็นภาษาไทย */
        return $date; // ส่งค่าพร็อปเพอร์ตี้ date กลับออกไป
    }
}
```

การเรียกใช้งาน : สเตติกเมธอดและสเตติกพร็อปเพอร์ตี้

```
$date = Datetime::thai_datetime('2013-12-01'); // การกำหนดค่าให้กับสเตติกเมธอด
echo "วันนี้คือ :: วัน".$date; // คำสั่งแสดงผลลัพธ์ echo
// output :: วันนี้คือ :: วันอาทิตย์ 01 ธันวาคม พ.ศ. 2556 // ผลลัพธ์ที่ได้
Datetime::date = '2013-12-01'; // การกำหนดค่าให้กับสเตติกพร็อปเพอร์ตี้
```

จากตัวอย่าง :

เราจะเห็นเครื่องหมาย “::” (Double Colon) ซึ่งมีความหมายว่า Scope Resolution Operator หรือชื่อภาษาไทยเรียกว่า “เครื่องหมายกำหนดขอบเขต” ในภาษา PHP นั้นถูกใช้สำหรับการเรียกใช้พร็อปเพอร์ตี้หรือเมธอดที่เป็นประเภท static ได้เท่านั้น

เพิ่มเติม สำหรับ สเตติกพร็อปเพอร์ตี้นั้นหากมีกำหนดเป็นแบบ static แล้ว พร็อปเพอร์ตี้ที่ถูกกำหนดจะไม่ใช่ พร็อปเพอร์ตี้ของออบเจกต์ ซึ่งมันจะแตกต่างจากพร็อปเพอร์ตี้ทั่วไป ถึงแม้ว่ามันจะอยู่ภายในคลาสเดียวกันก็ตามที่และอีกอย่างหนึ่ง static สเตติกเมธอด ก็จะไม่สามารถเรียกใช้ผ่านคำสั่ง \$this ได้ เนื่องจากเราไม่ได้มีการสร้างหรือประกาศตัวแปรอ่อนเจก์ต์มาก่อนนั่นเอง

## การสืบทดสอบของออบเจกต์

การสืบทดสอบหมายถึงความสัมพันธ์ระหว่างออบเจกต์แบบหนึ่ง ซึ่งมันจะเป็นเหมือนกับการสืบทดสอบพัฒนาจากแม่สู่ลูก ดังนั้นในแง่ของการเขียนโปรแกรมเชิงวัตถุนี้ การสืบทดสอบแบบนี้ จะถือเป็นเรื่องสำคัญมาก哉ที่เดียว สมมุติเราไม่คลาสแม่กับคลาสลูกการสืบทดสอบของออบเจกต์ของคลาสลูกสามารถสืบทดสอบจากคลาสแม่ได้เพียง 1 ออบเจกต์เท่านั้น

แต่ในทางกลับกันตัวที่เป็นคลาสแม่ จะถูกสืบทดสอบคุณสมบัติจากคลาสลูกไปได้ไม่จำกัดจำนวนครั้ง โดยคุณสมบัติต่าง ๆ ที่มีในคลาสแม่ คลาสลูกก็จะมีคุณสมบัติแบบนั้นอยู่เช่นเดียวกัน

ตัวอย่าง : การสืบทดสอบของออบเจกต์

```
// คลาสแม่
class Datetime { // คลาส Datetime
    public static $date; // พรีอปเพอร์ต date
    public static function thai_datetime($datetime) { // เมธอด thai_datetime()
        /* คำสั่งแปลงวันที่เป็นภาษาไทย */
    }
}

// คลาสลูก
class Date extends Datetime { // ใช้คำสั่ง extends ตามด้วยคลาสที่ต้องการสืบทดสอบคุณสมบัติ
    // คำสั่งต่อ ๆ
}

// ทดสอบการสืบทดสอบคุณสมบัติ
$objDate = new Date(); //Date() ชื่อคลาสลูก ที่สืบทดสอบคุณสมบัติมาจาก Datetime() คลาสแม่
$date = $objDate->thai_datetime('2013-12-01'); //เนื่องจากการสืบทดสอบจากคลาสแม่ เราจึงใช้เมธอดนี้ได้
echo "วันนี้คือ :: วัน".$date; //แสดงผลลัพธ์ echo
```

จากตัวอย่าง :

การทดสอบการสืบทดสอบคุณสมบัติจากคลาสแม่นั้น เราจะใช้คำสั่ง extends และตามด้วยชื่อคลาสแม่ที่เราต้องการสืบทดสอบคุณสมบัติ เมื่อเรา extends คลาสแม่เสร็จ เรายังสามารถเรียกใช้งานเมธอดหรือพรีอปเพอร์ตต่าง ๆ ที่อยู่ในคลาสแม่ได้ โดยเราต้องการแค่ต้องสร้างออบเจกต์จากคลาสลูก เช่นตัวอย่าง new Date() คลาส Date() ก็ชื่อคลาสลูกที่ถูกสืบทดสอบคุณสมบัติจากคลาสแม่ Datetime() เท่านั้น ทำให้เราสามารถเรียกใช้งานเมธอด \$objDate->thai\_datetime('2013-12-01') ที่อยู่ภายในคลาสแม่ได้

การอ้างอิงออบเจกต์

การอ้างอิงออบเจกต์นั้นเป็นเรื่องง่ายมาก ๆ ถ้าเราเขียนโปรแกรมแบบฟังก์ชันเหมือนเดิม แต่เมื่อไหร่ก็ตามที่เราเข้ามาสู่กระบวนการออบเจกต์แล้วมันจะต่างกันออกไป เพราะส่วนใหญ่แล้วข้อมูลเก็บทุกประเภทจะทำงานแบบ copy-on-write นั่นหมายความว่า ถ้าเราเรียก \$a=\$b; เราจะต้องคิดว่าค่าของ \$b ต้องเหมือนกับ \$a แต่หากเป็นการอ้างอิงออบเจกต์นั่น จะเป็นอีกแบบหนึ่ง เราลองมาดูตัวอย่างกันเลย

## ตัวอย่าง : การอ้างอิงออบเจกต์

```
// การสร้างออบเจกต์และการอ้างอิงออบเจกต์
$objDate1 = new Date(); // สร้างออบเจกต์
$objDate1->date = '2013-12-01';
$objDate2 = $objDate1; // การอ้างอิงออบเจกต์
$objDate2->date = '2013-12-02';

// ทดสอบ
echo "(1) วันที่ :: ".$objDate1->date;
echo "(2) วันที่ :: ".$objDate2->date;

// แสดงผลลัพธ์
// output : (1) วันที่ :: '2013-12-02' (2) วันที่ :: '2013-12-02'

// การตรวจสอบว่ามีการอ้างอิงคลาสหรือพร็อปเพอร์ตี้เดียวกันหรือไม่ ใช้เครื่องหมาย "==" 
If($objDate1 == $objDate2) echo 'อ้างอิงจากคลาสหรือพร็อปเพอร์ตี้เดียวกัน';
// การตรวจสอบว่ามีการอ้างไปยังออบเจกต์เดียวกันหรือไม่ ใช้เครื่องหมาย "===" 
If($objDate1 === $objDate2) echo 'อ้างอิงจากคลาสหรือพร็อปเพอร์ตี้เดียวกัน';
```

### จากตัวอย่าง :

การอ้างอิงออบเจกต์สามารถทำได้ โดยใช้งานเครื่องหมาย “=” เพื่อใช้เชื่อมระหว่างตัวแปรออบเจกต์ ทั้ง 2 ตัว เช่นดังตัวอย่าง \$objDate2 = \$objDate1; ซึ่งหมายถึงค่าของตัวแปร \$objDate2 จะถูกอ้างอิงไปถึง ออบเจกต์ Date() ที่ถูกสร้างและเก็บค่าออบเจกต์ไว้ในตัวแปร \$objDate1

โดยการอ้างอิงแบบนี้เราจะเห็นได้จากตัวอย่างว่า เมื่อเราได้กำหนดค่าให้กับพร็อปเพอร์ตี้ date ใน ออบเจกต์ Date() แล้ว ครั้งแรกกำหนดค่าพร็อปเพอร์ตี้ \$objDate1->date = '2013-12-01' และครั้งที่ 2 กำหนดค่าพร็อปเพอร์ตี้ \$objDate2->date = '2013-12-02'

แต่เมื่อเราต้องการแสดงผลลัพธ์ออกมา เราจะพบเห็นว่าค่าพร็อปเพอร์ตี้ date ของ ตัวแปร \$objDate1 กับตัวแปร \$objDate2 ค่าที่อุ่นมาจะเหมือนกัน คือ '2013-12-02' ซึ่งนั่นก็หมายความว่า พร็อปเพอร์ตี้ date ของ \$objDate1 ที่กำหนดไว้ในครั้งแรก ได้ถูกค่าของพร็อปเพอร์ตี้ date ของ \$objDate2 ของครั้งที่ 2 ได้เขียนทับ ค่าของพร็อปเพอร์ตี้ date ของ \$objDate1 นั่นเอง

**เพิ่มเติม :** จากตัวอย่างด้านบนนี้ ผู้อ่านได้ก่อตัวถึงการตรวจสอบการอ้างอิงว่ามีการอ้างอิงจากพร็อปเพอร์ตี้หรือเมธอดใดเดียวกันหรือไม่ โดยใช้เครื่องหมาย “==” ในการตรวจสอบ ส่วนการตรวจสอบการอ้างอิง จากออบเจกต์เดียวกันหรือไม่ โดยใช้เครื่องหมาย “==” ในการตรวจสอบเช่นกัน

ผลลัพธ์ที่ได้หลังจากการตรวจสอบ หากพบว่ามีการอ้างอิงมาจากแหล่งเดียวกันทั้ง 2 ตัวแปร PHP จะ คืนค่า True (ใช่) กลับอุ่นมาต่อ

## Public, Private, Protected (คีย์เวิร์ด)

Public : เป็นค่าเริ่มต้นที่ PHP กำหนดให้เรากรณีที่เราไม่ได้กำหนดคีย์เวิร์ดใด ๆ ลงไป แต่ตัว PHP จะกำหนดให้เราโดยอัตโนมัติว่า เมมเมอร์หรือพาร์ส์นี้เป็น Public แต่อย่างแนะนำให้เรากำหนดคีย์เวิร์ดไว้ด้วยเสมอคือกว่า เพราะจะช่วยแสดงให้เราหรือนักพัฒนาท่านอื่นได้ทราบว่าพาร์ส์หรือเมมเมอร์นี้สามารถถูกเรียกใช้งานได้จากทุก ๆ ที่ ของเว็บแอปพลิเคชัน เราได้

Private : คีย์เวิร์ดนี้ จะใช้สำหรับเมมเมอร์หรือพาร์ส์ที่เราต้องการเรียกใช้ได้ภายในคลาสที่ถูกประกาศไว้เท่านั้น เรายังสามารถนำห้ามเมมเมอร์หรือพาร์ส์ที่ไม่มีความจำเป็นต้องถูกเรียกใช้งานจากภายนอกให้ประกาศเป็น Private ทั้งหมด ดังนั้นห้ามเมมเมอร์หรือพาร์ส์ไหน ที่เราคิดว่าไม่ถูกเรียกใช้งานจากภายนอกได้ หรือคิดว่าเมมเมอร์หรือพาร์ส์ตี่นั้น ไม่มีความสามารถเรียกใช้จากภายนอกได้ เราจึงกำหนดเป็น Private เสีย

Protected : คีย์เวิร์ดตัวนี้ มีความน่าสนใจย่างมาก เพราะ PHP Framework ส่วนใหญ่แล้ว จะมีเมมเมอร์และพาร์ส์ตี่แบบนี้อยู่มากมาย เนื่องจากเบื้องต้นแล้ว ตัวมันเองสามารถถูกเรียกใช้ได้จากคลาสของตัวมันเอง และยังสามารถถูกเรียกใช้ได้ตามคลาสที่ได้สืบทอด過來 เช่นกัน

## PHP MVC (Model-View-Controller)

แนวคิดที่ใช้อธิบายความสัมพันธ์ของ 3 ส่วนหลัก ๆ ที่มีเชื่อมต่อเข้าด้วยกันเป็นการพัฒนาระบบงาน โครงสร้างของระบบงานที่ถูกออกแบบตามดีไซน์แพทเทิร์น Model-View-Controller นั้น จะทำให้ระบบงานมีความยืดหยุ่น ง่ายต่อการพัฒนาและแก้ไข เช่น การแยกส่วนการแสดงผลและการประมวลผลออกกัน ดังนั้นเราจึงสามารถนำข้อมูลไปแสดงผลได้หลายรูปแบบ เช่นการแสดงผลแบบตาราง HTML และข้อมูลรูปแบบ JSON หรือ XML เป็นต้น ซึ่ง MVC จะประกอบไปด้วย 3 ส่วนดังนี้

**Model – data layer :** เป็นส่วนที่รับข้อมูลมาเข้าห้องหมนที่ถูกส่งมา�ังโมเดลเป็นส่วนสุดท้ายและผลลัพธ์การทำงานทั้งหมด ถูกดึงออกมาจากโมเดล โดยโมเดลอาจจะเป็นฐานข้อมูลหรือระบบไฟล์ ก็ได้

**View – presentation layer :** เป็นส่วนแสดงผลที่เรา นำเสนอข้อมูลที่ได้จากโมเดลต่าง ๆ มาสร้างเป็นหน้าแสดงผลลัพธ์ เพื่อส่งต่อข้อมูล Web Browser ของผู้ใช้งาน

**Controller – application flow layer :** เป็นส่วนกลางที่ควบคุมการทำงานของระบบต่าง ๆ และการให้เกี่ยวกับข้อมูล พร้อมการตัดสินใจว่าในแต่ละคำขอของจากผู้ใช้งานนั้น ควรจะดึงข้อมูลจากโมเดล (Model) อะไรออกมาและจะส่งผลลัพธ์ไปยังวิว (view) อะไร เพื่อแสดงผลลัพธ์กลับไปยังผู้ใช้งาน

โดยทั่วไประบบงานที่ใช้แนวคิดแพทเทิร์น MVC เป็นโครงสร้างหลักในการพัฒนาซอฟแวร์นั้น จะมีกำหนดให้ 1 คน โตรลเลอร์ต่อหน่วยการทำงานเท่านั้น โดยก่อนเข้าทำงานในคอนโทรลเลอร์จะมีสิ่งที่เรียกว่า Router ทำหน้าที่ดักจับคำขอของจากผู้ใช้งานและจัดการว่า แต่ละคำขอร้องที่เข้ามาต้องการทำงานกับคอนโทรลเลอร์ใด และต่อจากนั้นกอนโทรลเลอร์แต่ละตัวจะเรียกใช้งาน โมเดล ต่าง ๆ ที่มีในระบบ เช่น ข้อมูลสมัครสมาชิก ข้อมูลสมาชิก ข้อมูลการขายสินค้า เป็นต้น

## Controller (คอนโทรลเลอร์)

โดยพื้นฐานของคอนโทรลเลอร์แล้ว จะทำหน้าที่รับคำสั่งจาก HTTP ต่าง ๆ อาทิเช่น GET, POST, SESSION, FILES และนำข้อมูลตั้งกล่าวมาประมวลผล ตามคำสั่งที่อยู่ในคอนโทรลเลอร์

```
class MainController { // สร้างคอนโทรลเลอร์ Main
    $page = $_GET['page']; // สร้างตัวแปร $page ให้รับค่าจาก HTTP GET['page']
    $model = new Main_Model(); // เชื่อมต่อโมเดล Main_Model
    $date = $model->findAll(); // ค้นหาข้อมูลฐานข้อมูลทั้งหมด
    require_one $page.-view.php'; // ดำเนินการเปิดไฟล์
}
```

## Model (โมเดล)

ตามโครงสร้างพื้นฐาน MVC แล้ว เราจะทำการกำหนดว่า 1 โมเดล ต่อ 1 คอนโทรลเลอร์ หรือ หลายโมเดล ต่อ 1 คอนโทรลเลอร์ก็สามารถทำได้ ขึ้นอยู่กับการออกแบบและเขียนคำสั่งควบคุมในตัวคอนโทรลเลอร์ จะเป็นตัวกำหนด ในโมเดลอาจมีเมธอดตามแยกชั้นในคอนโทรลเลอร์ จากตัวอย่างการสร้างคอนโทรลเลอร์เราได้สร้างคอนโทรลเลอร์ MainController และมีแยกชั้น findAll() ดังนี้ในโมเดล เราต้องสร้างคลาส Main\_Model ซึ่งมีเมธอด findAll() ดังตัวอย่าง

ตัวอย่าง : การสร้าง Model (โมเดล)

```
class Main_Model { // สร้างโมเดล Main_Model
    public function findAll() { // สร้างเมธอด findAll()
        $val = array('name', 'address', 'tel');
        return $val; // คืนค่ากลับออกไปเป็นรูปแบบ Array
    }
}
```

## View (วิว)

สำหรับการแสดงผล นั้นจะทำงานหลังจากแยกชั้นได้ ๆ ในคอนโทรลเลอร์ทำงานเสร็จเรียบร้อยแล้ว ในตัวอย่างนี้ เราจะตั้งคลาสชื่อว่า ShowPhoto มีเมธอด render() ทำหน้าที่นำข้อมูลที่ได้มาออกไปแสดง ผลตามความต้องการ

ตัวอย่าง : การคลาส ShowPhoto และเมธอด render()

```
class ShowPhoto { // สร้างคลาส ShowPhoto
    public function render() { // สร้างเมธอด
        $html = '';
        $return = sprint($html,"images/img.jpg",300,500);
        return $return;
    }
}
```

```
$model = new ShowPhoto(); // สร้างออบเจกต์ ShowPhoto()
echo $model->render(); // อ่านค่าเมธอด
// output ::  // คำสั่ง HTML IMG
```

## PHP OOP และ MVC เกี่ยวกับ Yii Framework อย่างไร

OOP (Object-Oriented Programming) จะเป็นการมองทุก ๆ อย่างเป็นเหมือนวัตถุชิ้นหนึ่ง ซึ่งจะมีความสัมพันธ์ระหว่างฟังก์ชันกับตัวแปรต่าง ๆ ทั้งหมด ภายในแต่ละคลาสนั่นเอง อีกทั้งยังสามารถจำกัดขอบเขตการทำงานของแต่ละคลาสได้อย่างชัดเจน เช่น คลาส “ขายสินค้า” มีเมธอด “คำนวณยอดขาย” “บันทึกข้อมูล” “พิมพ์ใบเสร็จ” แบบนี้เป็นต้น

คำสัพท์ที่เราควรต้องจำเอาไว้อย่างให้ลืมก็คือ คลาส (Class) օบเจกต์ (Object) เมธอด (Method) พร็อปเพอร์ตี้ (Property) และค่าพารามิเตอร์ ซึ่งคำสัพท์เหล่านี้ ล้วนแต่เป็นคำสัพท์ขึ้นพื้นฐานที่ถูกกำหนดขึ้นมาจาก หลักการเขียนโปรแกรมเชิงวัตถุมาตั้งแต่ต้น เราจะเจอกำคำสัพท์เหล่านี้อยู่ตลอดเวลาที่เรออยู่กับสิ่งที่เรียกว่า “օบเจกต์”

สุดท้ายที่สำคัญไม่แพ้กันก็คือ “คีย์เวิร์ด” Public, Private, Protected คีย์เวิร์ดเหล่านี้จะถูกกำหนดไว้หน้า พร็อปเพอร์ตี้ หรือเมธอด ต่าง ๆ เพื่อใช้แสดงถึงคุณสมบัติและความสามารถในการเรียกใช้งาน จะระบุบนเบตการประกาศใช้ได้จากล่วงหน้าได้ นั่นเอง ซึ่ง คีย์เวิร์ด Public ถึงเราจะกำหนดหรือไม่กำหนดก็ตาม แต่ในภาษา PHP แล้ว มองว่าหากเราไม่ได้ใส่ คีย์เวิร์ด Public ไว้นานหน้า พร็อปเพอร์ตี้หรือเมธอดภาษา PHP จะมองว่าพร็อปเพอร์ตี้และเมธอด เหล่านั้นเป็น Public โดยอัตโนมัติทั้งหมด ดังนั้นการเขียนพร็อปเพอร์ตี้และเมธอดที่ดี เราจึงควรกำหนดคีย์เวิร์ดไว้ด้วยเสมอ

แนวคิด MVC (Model-View-Controller) ใช้อธิบายความสัมพันธ์ของ 3 ส่วนหลัก ๆ ที่มีเชื่อมต่อเข้าด้วยกัน เป็นการพัฒนาระบบงาน โครงสร้างของระบบงานที่ถูกออกแบบเป็น MVC นั้น จะทำให้ระบบงานมีความยืดหยุ่น ง่ายต่อการพัฒนาและแก้ไข เช่น การแยกส่วนการแสดงผลและการประมวลผลออกจากกันและกัน เพื่อนำข้อมูลไปแสดงผลได้หลายรูปแบบ เช่นการแสดงผลแบบตาราง HTML และข้อมูลรูปแบบ JSON เป็นต้น

Yii Framework ถูกพัฒนาขึ้นมา โดยอาศัยหลักการออกแบบแพทเทิร์น MVC และจุดเด่นของ PHP OOP รวมตัวกัน นำมาเป็นหลักการขึ้นพื้นฐานในการออกแบบและพัฒนา Yii Framework ขึ้นมา ซึ่ง MVC ในความหมายของ Yii Framework นั้นจะหมายถึง หลักการออกแบบ (Design Pattern) รูปแบบหนึ่ง ซึ่งเป็นที่นิยมมากในการใช้การออกแบบและประยุกต์ใช้กับเว็บแอ��泲ิเคชัน ซึ่งเจ้า MVC นี้จะแบ่งแอ��泲ิเคชันตามบทบาทหน้าที่ (Roles of objects) โดยแบ่งออกเป็น 3 บทบาทด้วยกัน

1. Model (M) หน้าที่เป็นตัวแทนของข้อมูล เพื่อเปลี่ยนเป็นօบเจกต์และเป็นส่วนจัดการข้อมูล
2. View (V) หน้าที่ในการแสดงผล เช่น ข้อความ form input หรือ (User Interface: UI) เป็นต้น
3. Controller (C) หน้าที่รับคำสั่งและเรียกใช้อบเจกต์ตัวอื่น ๆ (Model และ View) ให้ทำงานร่วมกัน และหน้าที่เป็นตัวควบคุมเส้นทางการทำงานของคำสั่งต่าง ๆ ด้วย

โดย MVC นั้นจะช่วยแยกส่วนการประมวลผล ออกมายังการแสดงผล (User Interface: UI) ส่วนผลให้ง่ายต่อการแก้ไขในแต่ละส่วน ซึ่งจะไม่กระทบกับส่วนอื่นๆ

**ตัวอย่าง :** การสร้างโมเดล (Model) เชื่อมต่อฐานข้อมูล ใน Yii Framework

ตัวอย่าง โมเดลชื่อว่า “Comment” ใน Yii Framework จะมีการสืบทอดคุณสมบัติมาจากการคลาส CActiveRecord ซึ่งเป็นคลาสแม่ที่ Yii Framework ได้ถูกออกแบบไว้เป็นคลาสหลัก ทุก ๆ ครั้งที่มีการสร้างโมเดลใหม่ ต้องมีการสืบทอดคุณสมบัติมาจากการคลาส CActiveRecord เท่านั้น

**โมเดล : Comment**

```
class Comment extends CActiveRecord {
    public function tableName() {
        return 'tbl_comment'; // ชื่อตาราง "tbl_comment" ในฐานข้อมูล
    }
    public function rules(){
        /* ส่วนการกำหนดคุณสมบัติให้กับ form input เช่น field :: Name ต้องเป็นชนิดตัวอักษร */
    }
    public function relations(){
        /* ส่วนในการทำ Relations หรือ ความสัมพันธ์ ระหว่าง Model ด้วยกัน */
    }
    public function attributeLabels(){
        /* ส่วนที่กำหนดข้อความ Label ให้กับ field ในงานร่วมกับ form input */
    }
}
```

**ตัวอย่าง : วิว (View)**

ในส่วนของ วิว (view) ใน Yii Framework นั้น ได้กำหนดให้เป็นที่ใช้เขียนคำสั่งในการแสดงผลลัพธ์ เช่น คำสั่งภาษา HTML PHP JAVASCRIPT JQUERY เป็นต้น ที่ถูกครอบคลุมด้วย Layout อีกหนึ่งชั้น นั้นก็หมายความว่า เมื่อเราได้สร้าง Layout ขึ้นมา เพื่อใช้เป็นโครงสร้างหลักในการกำหนดส่วนที่ต้องการแสดงผลลัพธ์ตามที่ได้ออกแบบไว้แล้ว ไฟล์ View ที่เราสร้างขึ้นก็เป็นเพียงไฟล์ที่ใช้สำหรับแสดงผลในส่วนของเนื้อหา (content) เท่านั้น เพราะหลักการออกแบบ Layout ขึ้นมาแล้ว จะช่วยลดปริมาณการเขียนคำสั่งที่ถูกใช้งานช้า ๆ กัน ในทุก ๆ หน้า ตัวอย่างเช่น คำสั่ง HTML TAG (html, body, title, meta) หรือคำสั่งดึงไฟล์ CSS หรือ JAVASCRIPT เข้ามาใช้ภายในหน้าเว็บ โดยปกติแล้วคำสั่ง HTML TAG ต่าง ๆ เหล่านี้ จะต้องมีการกำหนดไว้ตั้งแต่ต้นที่หน้าเว็บ ไซต์ต้องมีเป็นไปตามมาตรฐานตามหลักการเขียนโปรแกรมภาษา HTML อยู่แล้ว

**ตัวอย่าง : คำสั่ง view()**

```
<p> ยินดีตอนรับสู่ :: เว็บไซต์ DriveSoftwareCenter.Com </p>
<p>ศูนย์รวมความรู้ทางด้านการพัฒนาซอฟแวร์ของคนไทย</p>
```

## ตัวอย่าง : คอนโทรลเลอร์ (Controller)

Yii Framework ได้มีการออกแบบและกำหนดคลาสหลัก ๆ ไว้ให้เราเลือกใช้อยู่แล้ว อย่างเช่น โมเดล เราก็ต้องสืบทอดคุณสมบัติมาจากคลาส CActiveRecord ในคอนโทรลเลอร์นี้ ที่เขียนดีกว่าการสร้าง โมเดลขึ้นมาใหม่ Yii Framework จึงได้สร้างคลาสหลักไว้ให้เราได้สืบทอดคุณสมบัติ นั่นก็คือคลาส Controller เมื่อได้ก็ตามที่เราต้องการสร้าง คอนโทรลเลอร์ ขึ้นมาใหม่ เราต้องทำการสืบทอดคุณสมบัติ (extends) คลาสหลักหรือคลาสแม่ เข้ามาด้วย

ในตัวอย่างนี้ เราจะสร้าง แออกชัน ชื่อว่า “actionIndex()” ขึ้นมา และเขียนคำสั่งดึงข้อมูล “Comment” จากโมเดล Comment โดยใช้คำสั่ง Comment::model()->findAll() ในการดึงข้อมูล คำว่า findAll() หมายถึงการดึงข้อมูลออกมาทั้งหมด และสุดท้าย เรียกคำสั่ง \$this->render() เพื่อสั่งให้ Yii Framework เปิด อ่านไฟล์ ชื่อว่า webboard.php ซึ่งไฟล์จะอยู่ในโฟล์เดอร์ views ที่อยู่ในไดเรกทอรี่ [sketeton]/protected/views/comment/webboard.php ขออธิบายเสริม คำว่า “comment” ก็จะต้องตั้งตามชื่อ Controller ที่เราต้องการเปิดอ่านไฟล์ เช่นในตัวอย่างเราใช้ CommentController ดังนั้นเราจะเรียกคำสั่งดังนี้ \$this->render('webboard',array('model'=>\$model)); สังเกตใหม่เราจะคำสั่งใหม่อีกแล้วนั่นก็คือ array('model'=>\$model) มันก็หมายถึงให้แนบตัวแปรชื่อว่า ‘model’ ส่งเข้าไปในไฟล์ webboard.php ด้วย และในไฟล์ webboard.php เองก็จะสามารถใช้งานตัวแปร \$model ได้เลย ไม่ต้องประกาศใช้ตัวแปร

แต่ยังไม่ใช่ท่านนี้ เรา yang สามารถกำหนดแบบตัวแปร ได้หลาย ๆ ตัว โดยใช้เครื่อง “,” ในการกันกลางระหว่าง 2 ตัวแปร ได้ เช่น array('model'=>\$model, 'model2' => \$model2) เป็นต้น

## ตัวอย่าง : CommentController.php

```
class CommentController extends Controller {  
    public function actionIndex() { // เมธอดนี้ จะถูกสร้างขึ้นอัตโนมัติ เมื่อสร้างผ่าน Gii Code Generate  
        $model = Comment::model()->findAll(); // เชื่อมต่อ Model Comment และเพื่อดึงข้อมูล  
        // คำสั่ง render คือ สั่งให้เปิดอ่านไฟล์ชื่อว่า webboard.php และแนบตัวแปร $model,$name  
        $this->render('webboard',array(  
            'model'=>$model,  
            'name' => 'ทดสอบการแนบค่าตัวแปร'  
        ));  
    }  
}
```

## การพัฒนาเว็บแอปพลิเคชัน ด้วย Yii Framework

### บทที่ 2 Yii Framework คืออะไร



เนื้อหาในบทเรียนนี้ เราจะมาทำความรู้จักกับ PHP Framework ที่มีชื่อว่า Yii PHP Framework ซึ่ง PHP Framework ที่ได้รับความนิยมจากนักพัฒนาเพิ่มมากขึ้น ณ. เวลาขณะนี้ มีอยู่ประมาณ 5 ตัวด้วยกัน ประกอบไปด้วย Cake PHP Framework, Codeigniter Framework, Zend Framework, Laravel Framework และก็ตัว Yii Framework

ในบทนี้ เราจะได้เรียนรู้ความหมายของ Yii Framework กระบวนการทำงานต่าง ๆ ส่วนประกอบที่สำคัญต่าง ๆ พร้อมทั้งคำอธิบายขั้นตอนการดาวน์โหลดและติดตั้ง Yii Framework ปิดท้ายด้วยการตรวจสอบระบบหลังจากการติดตั้ง yii เสร็จเรียบร้อยแล้ว

#### ความหมาย Yii Framework

Yii Framework (อ่านว่า อาย) เป็น PHP Framework ที่กำลังมาแรงและได้รับความนิยมมากที่สุดในตอนนี้ สร้างตามแนวคิดของ MVC (Model-View-Controller) โดยเป็นแนวคิดที่ได้รับความนิยมอย่างมาก ในหมู่นักพัฒนาซอฟต์แวร์ในปัจจุบัน การพัฒนาซอฟต์แวร์ตามแนวคิดนี้ ได้รับอิทธิพลจากการเขียนโปรแกรมแบบ OOP คือ การแบ่งโปรแกรมหรือแอปพลิเคชันออกเป็นองค์ประกอบย่อย ๆ แต่ละองค์ประกอบทำหน้าที่หลักเพียงอย่างเดียว สุดท้ายทุก ๆ องค์ประกอบต้องทำงานร่วมกันอย่างเป็นแอปพลิเคชันที่สมบูรณ์

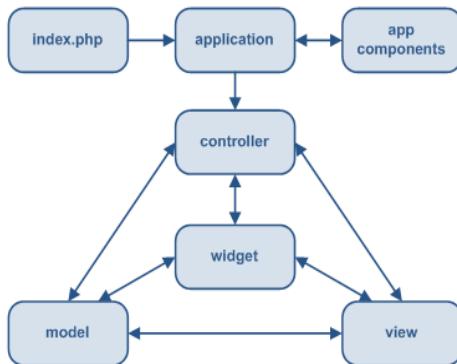
#### พัฒนาและออกแบบตามแนวคิด 3 แบบหลักของ Yii Framework

Yii Framework ได้ถูกพัฒนาและออกแบบตามแนวคิด 3 แบบหลัก ๆ นั่นก็คือ คำที่ออกแบบเสียงว่า อาย ที่มาจากการคำน้ำหน้าของ 3 คำต่อไปนี้

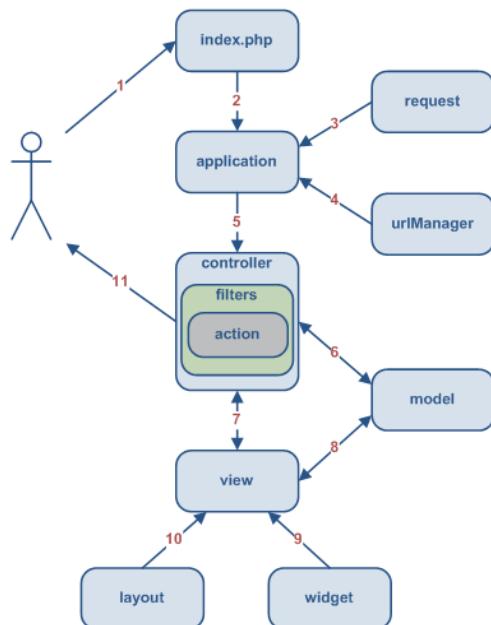
1. Easy (ง่าย)
2. Efficient (มีประสิทธิภาพ)
3. Extensible (สามารถขยายออกได้)

## แนะนำส่วน Front Controller

นอกจาก Yii Framework ข้างมีส่วนของ Front Controller ซึ่งเรียกว่า Application จะเป็น Context สำหรับการประมวลผล Request ต่าง ๆ โดย Application ตัดสินใจว่า Request แต่ละตัวจะส่งไปยัง Controller อะไร เพื่อทำงานต่อไป

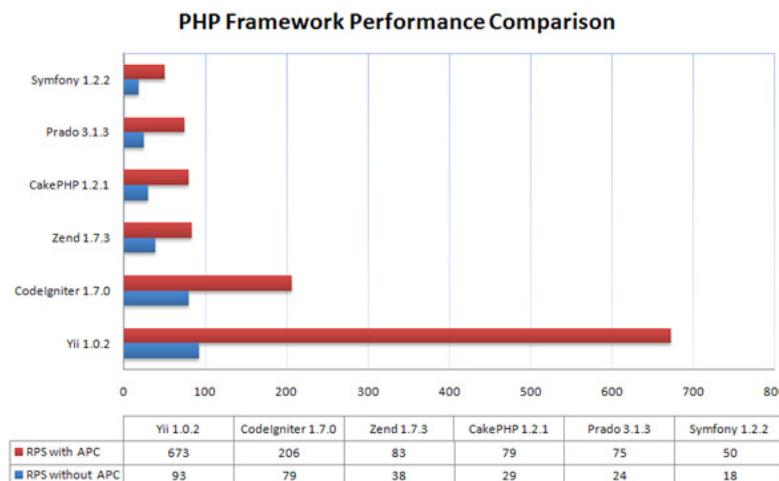


## กระบวนการทำงานของ Yii Framework



## ข้อดีของ Yii Framework มีอะไรบ้าง ?

1. MVC framework ที่สร้างด้วยภาษา PHP
2. เหมาะสมสำหรับการพัฒนาระบบงานขนาดใหญ่ เช่น CMS, e-Commerce , Blog เป็นต้น
3. ประสิทธิภาพการทำงานดี มีความรวดเร็ว เมื่อเทียบ PHP Framework ตัวอื่น ๆ
4. สนับสนุนการทำงานแบบ PHP OOP MVC
5. ใช้ PHP 5.1.0 ขึ้นไป
6. Component-Based / Event-Based framework



## ความสามารถของ Yii เมื่อเทียบกับ Framework อื่น ๆ

ความสามารถของ Yii เมื่อเทียบกับ PHP Framework เจ้าอื่นๆ นั้นถือว่า Yii Framework มีระบบที่รองรับและครอบคลุมการใช้งานมากที่สุด จึงเป็นเหตุผลที่ว่าทำไมถึงเลือก Yii Framework มาเป็น Framework คู่กาย คู่ใจ เพราะมันทำได้แบบทุกอย่างนี้แหล่



## ดาวโหลด Yii Framework v1.1.15 released on Jun 29, 2014

1. เข้าไปที่ เว็บไซต์ <http://www.yiiframework.com/>
2. คลิกปุ่ม Download Yii || ดาวโหลดที่นี่ <http://goo.gl/fL3jJS>
3. เลือกที่เก็บไฟล์ แนะนำหากติดตั้ง Appserv ให้เก็บไว้ในโฟล์เดอร์ C:\AppServ\www\ แต่ถ้าเป็น XAMPP ให้เก็บไว้ที่ โฟล์เดอร์ C:\xampp\htdocs\
4. แตกไฟล์ yii-1.1.15.022a51.tar.gz เสร็จ และเปลี่ยนชื่อ โฟล์เดอร์เป็น “yii” (แนะนำ)

## เตรียมความพร้อมก่อนการติดตั้ง (เครื่องมือ ตั้งค่าระบบ) ที่สำคัญ ๆ

เราจะมาเตรียมความพร้อมเครื่องมือ อุปกรณ์การพัฒนา การตั้งค่าระบบที่จำเป็นต่อการพัฒนาเว็บแอปพลิเคชัน ด้วย Yii Framework กันนะครับ โดยหลัก ๆ แล้วเครื่องมือหรืออุปกรณ์การพัฒนาเว็บแอปพลิเคชันนี้ เหล่านักพัฒนาเว็บแอปฯ จะเลือกเครื่องมือตามความถนัดหรือหารดware ที่เครื่องคอมพิวเตอร์ นั้นสามารถรองรับได้

โดยในบทเรียนนี้ผมได้เลือกเครื่องมือ อุปกรณ์ การพัฒนาเว็บแอปฯ

1. NetBeans 7.8 HTML5 & PHP Installer for Windows/English (en)
2. XAMPP Win32 v.1.8.3 (Apache 2.4.4,MySQL 5.5.32,PHP 5.4.19,phpMyAdmin 4.0.4)
3. Yii Framework v1.1.15 released on Jun 29, 2014

## การตั้งค่า Path ของ Windows (จากตัวอย่างเป็น Windows 7)

1. เข้าที่ My Computer คลิกขวา Properties และ เลือกเมนู Advanced system settings
2. เลือก Tab Advanced จากนั้น คลิกปุ่ม Environment Variables.. และเข้าไปที่ System Variables เลือก รายการ Path จากนั้นกดปุ่ม edit..
3. ให้เพิ่ม Path ที่อยู่ของไฟล์ PHP และ Path ที่อยู่ของไฟล์ Yii Framework

ที่สำคัญ : C:\xampp\php;C:\xampp\htdocs\yii\framework\;

Path PHP : C:\xampp\php;

Path Yii Framework : C:\xampp\htdocs\yii\framework\;

4. กดปุ่ม OK เป็นอันตั้งค่า Path ของ PHP และ Yii Framework ได้สำเร็จแล้ว

## ติดตั้ง Yii Framework v1.1.15 released on Jun 29, 2014

1. กดปุ่ม Start กรอกข้อมูลในช่อง “ค้นหา” ให้พิมพ์ว่า “cmd.exe” แล้ว เรายังพบรายการไฟล์ชื่อว่า “cmd.exe” (สีดำ) ให้เราคลิกรายการไฟล์ “cmd.exe” ที่ค้นหาพบ

2. เมื่อเข้าสู่โปรแกรม Command Prompt (cmd.exe) ให้เราพิมพ์คำว่า (XAMPP Server) C:\xampp\htdocs\yii\framework\ แล้วกดปุ่ม Enter (C: ใช้ในการนี้ที่เก็บไฟล์ yii ไว้ในโฟลเดอร์ C: นะครับ)

3. เมื่อเราเข้ามาอยู่ในโฟลเดอร์ C:\xampp\htdocs\yii\framework\ แล้ว พอดังจากเราเข้าไปยังไฟล์เครื่องที่เก็บไฟล์ Yii Framework ของเราแล้ว ที่นี่เราจะมาทำการพิมพ์คำสั่งติดตั้ง Yii Framework เพื่อสร้างเว็บแอ��เพล็กชั่น ด้วย Yii Framework กันเลย

4 โดยพิมพ์คำว่า “yiic webapp [PathProject]” ตามด้วยที่อยู่ไฟล์ที่ต้องการเก็บไฟล์เว็บของเราไว้ สมมุติเราต้องการสร้างเว็บที่ชื่อว่า “tutorial\_yii” ให้พิมพ์คำว่า “yiic webapp C:\xampp\htdocs\tutorial\_yii” แล้วกดปุ่ม Enter

5. หลังจากนั้น ระบบจะถามว่าเราต้องการสร้าง Web Application ในไฟล์เครื่องนี้ใช่หรือไม่ ให้เราพิมพ์ยืนยันไปเลยด้วยคำว่า “yes” จากนั้นกดปุ่ม Enter

6. หากเราสร้าง Web Application สำเร็จ ระบบจะแจ้งคำว่า “Your application has been created successfully under C:\xampp\htdocs\tutorial\_yii”

### แนะนำส่วนประกอบที่สำคัญต่าง ๆ ที่เราต้องรู้

ส่วนประกอบสำคัญต่าง ๆ ที่ได้มานะลังทำการติดตั้ง Yii Framework เสร็จเรียบร้อยแล้ว ส่วนประกอบที่สำคัญมีดังต่อไปนี้

1. ไฟล์เครื่อง assets
2. ไฟล์เครื่อง CSS
3. ไฟล์เครื่อง images
4. ไฟล์เครื่อง protected
5. ไฟล์เครื่อง themes

### ไฟล์เครื่อง assets

ไฟล์เครื่อง assets นี้ ส่วนใหญ่จะเก็บไฟล์ css, javascript, jquery, images, font ต่าง ๆ ที่ถูกสร้างขึ้นจากตัว extensions เพื่อจะได้เรียกใช้ widget ต่าง ๆ ที่มีอยู่ใน extensions นั้น ซึ่งในแต่ละ extensions ที่เรานำเข้ามาติดตั้งในเว็บแอพฯ เราจะมีไฟล์เครื่อง assets อยู่ในไฟล์เครื่อง extensions ด้วยเสมอ เมื่อเรามีการเรียกใช้งาน extensions ใด ๆ ตัว extensions ดังกล่าว จะทำการคัดลอกไฟล์ในไฟล์เครื่อง assets ของตัวมันเอง เอาไปไว้ในไฟล์เครื่องหลักที่ใช้รวมกันของ assets รูปแบบ assets ที่ถูกคัดลอกมา จะเป็นรหัสชื่อไฟล์เครื่อง

## ไฟล์เดอร์ css

ไฟล์เดอร์ CSS จะใช้เก็บไฟล์ CSS ที่เราสร้างขึ้นหรือไฟล์ CSS จากภายนอกที่เราต้องการใช้ ส่วนใหญ่แล้วจะถูกเรียกใช้ในไฟล์ Layout หรือไฟล์ View ต่างๆ

## ไฟล์เดอร์ images

ไฟล์เดอร์ images ที่ใช้เก็บไฟล์รูปภาพต่างๆ ไฟล์รูปไอคอน รูปโลโก้ รูปพื้นหลัง เป็นต้น ส่วนใหญ่แล้ว รูปภาพที่ใช้ทั้งหมดในเว็บแอพฯ นี้จะถูกเก็บรวมกันไว้กันที่ไฟล์เดอร์ images นี้ ทั้งหมด

## ไฟล์เดอร์ protected

ไฟล์เดอร์ protected ถือเป็นไฟล์เดอร์สำคัญที่สุด ซึ่งมันก็คือไฟล์เดอร์ที่เก็บไฟล์ คอนโทรลเลอร์ (Controller) โมเดล (Model) วิว (View) เอ็กซ์เทนด์ชั่น (Extensions) โมดูล (Modules) การตั้งค่า (Config) คอมโพเนนท์ (Components) เป็นต้น

ไฟล์เดอร์นี้เปรียบเสมือนเป็นไฟล์เดอร์ที่เก็บรวมไฟล์ที่สำคัญต่างๆ ทั้งหมดไว้ เพื่อที่จะทำให้ระบบเราทำงานได้ อาจจะเหมือนร่างกายของมนุษย์ก็เป็นได้ สุดท้ายเลย หากเราเกิดพบปัญหาเนื่องจากระบบทำงานผิดพลาด เราสามารถเข้ามาแก้ไขไฟล์ในไฟล์เดอร์ protected ได้เลย

## ไฟล์เดอร์ themes

สุดท้ายก็ไฟล์เดอร์ themes อันแล้วก็เปลี่ยนได้ตรงๆ ไปเลยครับ มันก็คือไฟล์เดอร์ที่เก็บไฟล์ Layout Themes สำหรับเว็บแอพฯ เราన่ำเงອ หากเราต้องการติดตั้ง Themes จากภายนอก เราต้องคลิกไฟล์ Themes ทั้งหมด จากภายนอก และนำมายกเข้าในไฟล์เดอร์ Themes และตั้งชื่อไฟล์เดอร์ให้ตรงกับชื่อ Themes ก็เป็นอันเสร็จแล้วครับ

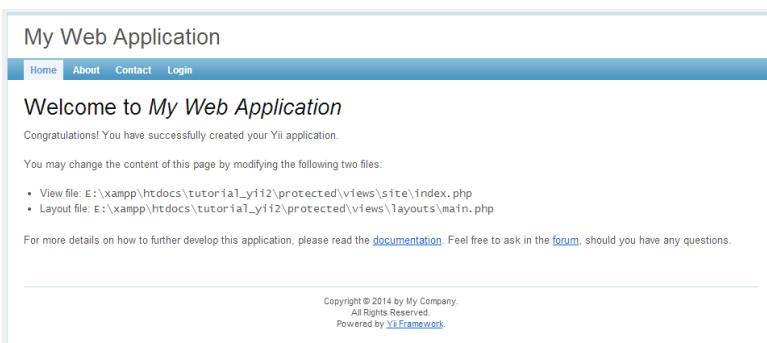


## ตรวจสอบระบบเบื้องต้นหลังการติดตั้ง Yii Framework

ก่อนอื่นเราต้องเปิดใช้งาน Appserv หรือ XAMPP จะก่อน เพื่อที่เราจะได้ใช้งาน localhost กันได้ (กรณีติดตั้ง Appserv Server จะทำงานให้อัตโนมัติ แต่ในส่วนของ xampp นั้น เราต้องคลิกปุ่มที่โปรแกรมxampp ก่อนครับ แนะนำให้สร้างเครื่อง 1 ครั้งหลังการติดตั้งโปรแกรม Appserv หรือ xampp)

**ทดสอบ :** เริ่มแอพพลิเคชั่นว่าทำงานได้หรือป่าว

เข้าไปที่ URL [http://localhost/tutorial\\_yii/index.php](http://localhost/tutorial_yii/index.php)



## ตรวจสอบ : ความพร้อมของระบบ

ความพร้อมของระบบ (เวอร์ชัน PHP, การติดตั้งตัวเชื่อมต่อฐานข้อมูล เป็นต้น)

1. ให้เข้าไปที่ URL <http://localhost/yii/requirements/>

2. ให้ตรวจสอบว่าระบบต่าง ๆ พร้อมใช้งานหรือไม่ ถ้าสถานะแสดงว่า Passed แปลว่าพร้อมใช้งาน แต่ถ้าแสดงสถานะขึ้นว่า Warning แปลว่ายังไม่ได้ติดตั้ง ส่วนสถานะตัวสุดท้ายคือ Failed แปลว่าระบบนี้ผิดพลาด ส่วนใหญ่แล้วจะเกี่ยวข้องกับฐานข้อมูลที่เราต้องการจะเชื่อมต่อ เช่น MySQL, SQLite, PostgreSQL, Oracle, MSSQL เป็นต้น

3. หาก PDO extension, PDO SQLite extension, PDO MySQL extension ขึ้นว่า “Passed” และดูว่า เราสามารถเริ่มพัฒนาเว็บแอพพลิเคชั่น ของเราราได้เลย เนื่องจากจะเชื่อมต่อฐานข้อมูล MySQL หากต้องการเชื่อมต่อฐานข้อมูลชนิดอื่น ๆ สามารถติดตั้ง extension เพิ่มเติมตามชนิดฐานข้อมูล

### Details

Name	Result	Required By	Memo
PHP version	Passed	<a href="#">Yii Framework</a>	PHP 5.1.0 or higher is required.
\$_SERVER variable	Passed	<a href="#">Yii Framework</a>	
Reflection extension	Passed	<a href="#">Yii Framework</a>	
PCRE extension	Passed	<a href="#">Yii Framework</a>	
SPL extension	Passed	<a href="#">Yii Framework</a>	
DOM extension	Passed	<a href="#">ChtmlPurifier</a> , <a href="#">CWsdlGenerator</a>	
PDO extension	Passed	All <a href="#">DB-related classes</a>	
PDO SQLite extension	Passed	All <a href="#">DB-related classes</a>	Required for SQLite database.
PDO MySQL extension	Passed	All <a href="#">DB-related classes</a>	Required for MySQL database.

## วิธีเปิดการใช้งาน PHP PDO extension

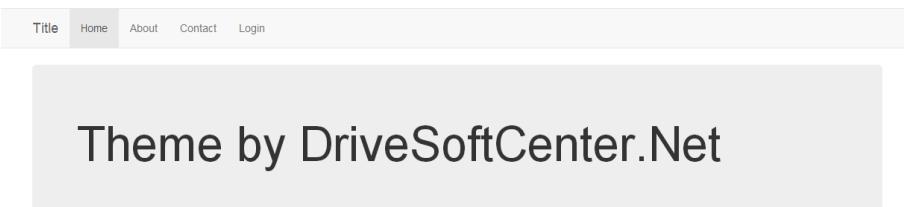
- กรณีติดตั้ง AppServ Server ให้เข้าไปที่ “ไดเรกทอรี่ C:\Windows\php.ini”
- กรณีติดตั้ง XAMPP Server ให้เข้าไปที่ “ไดเรกทอรี่ C:\xampp\php\php.ini”
- คลิกขวาที่ไฟล์ “php.ini” เลือก เปิดไฟล์ด้วยโปรแกรม EditPlus หรือตามความต้องการและให้กันหาคำว่า “extension=php\_pdo.dll, extension=php\_pdo\_mysql.dll, extension=php\_pdo\_sqlite.dll” ภายในไฟล์ “php.ini”

- เมื่อค้นหาเจอ 3 คำนี้แล้ว ให้เราตรวจสอบว่าหน้าคำมีข้อความ “;” อยู่หรือไม่ ถ้ามีให้เราเอาข้อความ “;” นี้ลบพื้นที่ของให้หมดทั้ง 3 คำ และทำการ Restart Server (AppServ, XAMPP) ด้วยนะครับ

```
598 ;extension=php_mssql.dll
599 extension=php_mysql.dll
600 extension=php_mysql.dll
601 ;extension=php_ocis.dll
602 ;extension=php_openssl.dll
603 ;extension=php_pdo.dll
604 ;extension=php_pdo_firebird.dll
605 ;extension=php_pdo_mssql.dll
606 ;extension=php_pdo_mysql.dll
607 ;extension=php_pdo_ocis.dll
608 ;extension=php_pdo_ocis.dll
609 ;extension=php_pdo_odbc.dll
610 ;extension=php_pdo_pgsql.dll
611 ;extension=php_pdo_sqlite.dll
612 ;extension=php_pgsql.dll
613 ;extension=php_psspell.dll
614 ;extension=php_shmop.dll
615 ;extension=pho_snmo.dll
```

## ดาวน์โหลดไฟล์ติดตั้ง Yii Framework v.1.1.15 (ปรับแต่ง โดย DriveSoftCenter.Net)

ดาวน์โหลดไฟล์ติดตั้ง Yii Framework v.1.1.15 + Yii Booster v.4.0 + Theme Bootstrap + SEO Extensions ซึ่งจะมาพร้อมกับ Yii Booster v.4.0 + Theme Bootstrap + SEO Extensions หลังการติดตั้งมาให้ภายในตัว โดยที่เราไม่ติดตั้งเพิ่มเติมภายหลังอีก || ดาวน์โหลดที่ <http://goo.gl/OfCNQp>



Congratulations! You have successfully created your Yii application.

You may change the content of this page by modifying the following two files:

- View file: E:\xampp\htdocs\PhpProject1\themes\bootstrap\views\site\index.php
- Layout file: E:\xampp\htdocs\PhpProject1\themes\bootstrap\views\layouts\main.php

For more details on how to further develop this application, please read the documentation. Feel free to ask in the [forum](#), should you have any questions.

Copyright © 2014 by DriveSoftCenter.Net

All Rights Reserved.

Powered by [Yii Framework](#).

ติดตาม ภาคผนวก คู่มือการใช้งาน Yii Booster v.4.0.1 (ท้ายเล่ม)

## การพัฒนาเว็บแอปพลิเคชัน ด้วย Yii Framework

### บทที่ 3 ตั้งค่าระบบก่อนเริ่มพัฒนาเว็บแอปพลิเคชัน

บทนี้เราจะมาเรียนรู้วิธีการตั้งค่า Config หรือกำหนดค่าเริ่มต้นให้กับเว็บแอปพลิเคชันที่สำคัญก่อนเริ่มลงมือเขียนโปรแกรมด้วย Yii Framework ให้นี่ เราจำเป็นต้องมีการกำหนดหรือตั้งค่าเบื้องต้นให้กับเว็บแอปฯ ซะก่อน อย่าง เช่น ตั้งค่าการเชื่อมต่อฐานข้อมูล กำหนดชื่อเว็บแอปพลิเคชัน แบบนี้เป็นต้น ซึ่งมันจะสามารถทำให้เว็บแอปฯ นั้น ทำงานได้ตามปกติหรือตามมาตรฐานที่ผู้พัฒนา Yii Framework "ได้กำหนดไว้ดังเดตตัน"

การตั้งค่า Config ทั้งหมดที่จะได้เรียนรู้ในบทเรียนนี้ จะแก้ไขไฟล์ที่ main.php [ไฟล์เดียวเท่านั้น] ไฟล์จะอยู่ภายใต้โฟลเดอร์ config ที่อยู่ในไดเรกทอรี่ [skeleton]/protected/config/ เนื่องจากเป็นไฟล์ที่สำคัญที่สุดของเว็บแอปฯ การตั้งค่าส่วนใหญ่ เช่น การติดตั้ง Extensions การสร้างคอมโพเน็นท์ ติดตั้งโมดูลใหม่ ตั้งค่าฐานข้อมูล เป็นต้น ไฟล์ main.php จะมีรูปแบบการตั้งค่าหรือกำหนดค่าเป็นแบบข้อมูล Array ทั้งหมด หากใครยังไม่เข้าใจรูปแบบข้อมูลแบบ Array ผูกขอแนะนำให้ลองไปศึกษามาก่อนนะครับ เพราะส่วนใหญ่แล้วในตัว Yii Framework จะมีการใช้งานรูปแบบของข้อมูลแบบ Array แทนทั้งหมด ถ้าเราไม่ศึกษา Array มาพอเข้าใจ ผูกขอแนะนำว่า คุณจะเรียนรู้ Yii Framework ได้ช้ามาก ๆ ครับ (แนะนำให้ศึกษา Array มา ก่อนนะครับ)

#### การกำหนดชื่อเว็บแอปพลิเคชันของเรา

กำหนดชื่อเว็บแอปพลิเคชันใหม่ ตามที่เราต้องการ โดยปกติ Yii Framework ได้กำหนดให้ 'name' เป็นค่าว่า 'name'=>'My Web Application' เราสามารถแก้ไขเป็น 'name' => 'Tutorial Yii Framework' หรือชื่อเว็บแอปพลิเคชันของเราเองก็ได้

คำสั่ง : PHP

ไดเรกทอรี่ : [skeleton]/protected/config/main.php

```
return array(
    // แก้ไขชื่อเว็บแอปฯ เป็น "Tutorial Yii Framework" หรือตามที่เราต้องการ
    'name'=>'Tutorial Yii Framework', // แก้ไขใน array 'name' เท่านั้นนะครับ
);
```

#### เม็ดการทำงานระบบ Gii Code Generator

Gii เป็นเครื่องมือที่ใช้ในการ Generate โค้ดต่าง ๆ เพื่อใช้ในการสร้างเว็บแอปพลิเคชัน ให้เราเข้าไปแก้ไขด้วยการลบ Comment หรือเครื่องหมาย /\* ... \*/ ของ Gii ออกและทำการกำหนด password ให้กับ Gii ตามที่เราต้องการ จากตัวอย่างจะกำหนดรหัสเป็น 12345 เพื่อใช้ในการเข้าไปใช้งานระบบ Gii Code Generator ครับ

คำสั่ง : PHP

ไฟล์ config/main.php

```
return array(
    .....
    'modules'=>array(
        'gii'=>array(
            'class'=>'system.gii.GiiModule',
            'password'=>'12345', // กำหนดรหัสผ่าน
            'ipFilters'=>array('127.0.0.1','::1'),
        ),
    ),
    .....
```

### เปิดใช้งานระบบ urlManager จัดการ URL

การตั้งค่าในส่วนของ Component ผู้จะทำการตั้งค่า URL ใหม่นั่นคือ โดยให้เราลบ Comment หรือ เครื่องหมาย /\* ... \*/ ในส่วนของ urlManager เพื่อทำการ Format การเรียกใช้ URL ใหม่

คำสั่ง : PHP

ไฟล์ config/main.php

```
return array(
    .....
    'components'=>array(
        'urlManager'=>array(
            'urlFormat'=>'path',
            'rules'=>array(
                'gii'=>'gii', // เพิ่มเติมคำสั่งนี้ลงไป
                'gii/<controller:\w+>'=>'gii/<controller>', // เพิ่มเติมคำสั่งนี้ลงไป
                'gii/<controller:\w+>/<action:\w+>'=>'gii/<controller>/<action>', // เพิ่มเติมคำสั่งนี้ลงไป
                '<controller:\w+>/<id:\d+>'=>'<controller>/view',
                '<controller:\w+>/<action:\w+>/<id:\d+>'=>'<controller>/<action>',
                '<controller:\w+>/<action:\w+>'=>'<controller>/<action>',
            ),
            'showScriptName' => false, // เพิ่มเติมคำสั่งนี้ลงไป
        ),
    ),
    .....
```

เราสามารถกำหนดให้ URL ใหม่ได้ โดยใช้ urlManager ที่ Yii Framework ออกแบบมาให้เราได้เรียกใช้งานอยู่แล้ว และยังมี Component อีกมาก ๆ ที่ Yii Framework เตรียมไว้ให้เราใช้ แต่เราจะไปศึกษาต่อในบทต่อ ๆ ไปละกันนะครับ

ต่อไปก่อนเราจะสร้างไฟล์ .htaccess เราต้องมีการเปิดใช้งาน Module Mod\_Rewrite จะก่อน

- Appserv ส่วนใหญ่จะอยู่ใน C:/AppServ/Apache2.2/conf/httpd.conf

- Xampp ส่วนใหญ่จะอยู่ใน C:/xampp/apache/conf/httpd.conf

เมื่อเจอกไฟล์ “httpd.conf” นี้แล้ว จากนั้นให้ทำการแก้ไขด้วยโปรแกรม EditPlus

ค้นหาคำว่า “#LoadModule rewrite\_module modules/mod\_rewrite.so”

```

190 #LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
191 #LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
192 #LoadModule proxy_connect_module modules/mod_proxy_connect.so
193 #LoadModule proxy_http_module modules/mod_proxy_http.so
194 #LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
195 #LoadModule negotiation_module modules/mod_negotiation.so
196 #LoadModule rewrite_module modules/mod_rewrite.so
197 #LoadModule setenvir_module modules/mod_setenvir.so
198 #LoadModule spelling_module modules/mod_speling.so
199 #LoadModule status_module modules/mod_status.so
200 #LoadModule unique_id_module modules/mod_unique_id.so
201 LoadModule userdir_module modules/mod_userdir.so
202 #LoadModule usertrack_module modules/mod_usertrack.so
203 #LoadModule vhost_alias_module modules/mod_vhost_alias.so
204 #LoadModule ssl_module modules/mod_ssl.so
205 LoadModule php5_module E:/AppServ/php5\php5apache2_2.dll

```

เมื่อเจอแล้ว ให้เอาเครื่องหมาย “#” หน้าสุดออก จากนั้นกดปุ่ม Save และวิธี Restart Server 1 ครั้ง

สร้างไฟล์ .htaccess || ดาวโหลดได้ที่ <http://goo.gl/SnZt9h>

เพื่อให้การทำงานของระบบ urlManager เกิดการทำงานเป็นประ โยชน์สูงสุด เราจำเป็นเพิ่มคำสั่งต่อไปนี้ลงในไฟล์ .htaccess และกด save ไฟล์ และนำเอาไฟล์ไปวางไว้ในไดเรกทอรี่ [skeleton]/.htaccess

คำสั่ง : PHP Config

ไดเรกทอรี่ : [skeleton].htaccess

```

<IfModule mod_rewrite.c>
    Options +FollowSymLinks
    IndexIgnore */
    RewriteEngine On
    RewriteCond %{REQUEST_FILENAME} !-
    RewriteCond %{REQUEST_FILENAME} !-
    RewriteRule . index.php
</IfModule>

```

โดยเมื่อสร้างไฟล์ .htaccess แล้ว จะเกิดความแตกต่างดังนี้

การเรียกใช้งาน Gii จากเดิม การเรียกใช้จะมีลักษณะดังนี้

[http://localhost/tutorial\\_yii/index.php?r=gii](http://localhost/tutorial_yii/index.php?r=gii)

เมื่อสร้างไฟล์ .htaccess แล้ว การเรียกใช้จะมีลักษณะดังนี้

[http://localhost/tutorial\\_yii/gii](http://localhost/tutorial_yii/gii)

## การตั้งค่าการเริ่มต่อฐานข้อมูล

การตั้งค่าฐานข้อมูลใน Yii Framework นั้นสามารถ มีได้หลาย ๆ ฐานข้อมูลได้ แต่ในด้านย่างนี้ อาจจะยกตัวอย่างมาแต่ 1 ฐานข้อมูลเท่านั้น โดยเราจะสร้างฐานข้อมูลชื่อว่า “db\_tutorial\_yii”

คำสั่ง : PHP

ไฟล์ config/main.php

```
return array(
```

```
.....
```

```
'db'=>array(
```

```
    'connectionString' => 'mysql:host=localhost;dbname=db_tutorial_yii', // dbTutorial_yii ฐานข้อมูล
```

```
    'emulatePrepare' => true,
```

```
    'username' => 'root', // Username เข้าสู่ระบบฐานข้อมูล
```

```
    'password' => '12345', // Password สำหรับเข้าสู่ระบบฐานข้อมูล
```

```
    'charset' => 'utf8', // กำหนด Encode ฐานข้อมูลเป็น "utf8"
```

```
),
```

```
.....
```

## วิธีการตั้งค่าฐานข้อมูล

- แก้ไขชื่อ Host (localhost) และชื่อฐานข้อมูล (db\_tutorial\_yii) ในบรรทัด connectionString
- แก้ไข Username (root) ที่ใช้ชื่อเดียวกับฐานข้อมูล ในบรรทัด username โดย Username ต้องตรงกับ Username ที่ใช้ Login เข้าใช้ฐานข้อมูล
- แก้ไข Password (12345) ที่ใช้ชื่อเดียวกับฐานข้อมูล ในบรรทัด password โดย password ต้องตรงกับ password ที่ใช้ Login เข้าใช้ฐานข้อมูล

## กำหนด default ภาษาที่ใช้แสดงเริ่มต้น

การกำหนด default ภาษาที่ใช้แสดงในเว็บแอปพลิเคชันของเรา สามารถกำหนดภาษาให้กับเว็บแอพฯ ได้ รองรับหลากหลายภาษา

คำสั่ง : PHP

ไฟล์ config/main.php

```
return array(
```

```
.....
```

```
    'name'=>'Tutorial Yii Framework',
```

```
    'language' => 'th', // เพิ่มเติม
```

```
.....
```

ภาษาที่ Yii Framework จัดทำไว้ให้เรา เพื่อรองรับการใช้งานนี้ มีอยู่มากมาก ยกตัวอย่างเช่น ‘en’, ‘zh\_cn’, ‘zh\_tw’, ‘ca’, ‘de’, ‘el’, ‘es’, ‘sv’, ‘he’, ‘nl’, ‘pt’, ‘pt\_br’, ‘ru’, ‘it’, ‘fr’, ‘ja’, ‘pl’, ‘hu’, ‘ro’, ‘id’, ‘vi’, ‘bg’, ‘lv’, ‘sk’, ‘uk’, ‘ko\_kr’, ‘kk’, ‘cs’ เป็นต้น

## กำหนด TimeZone ให้กับเว็บแอ��波单击ชั่ว

การกำหนด TimeZone ให้กับเว็บแอ��波单击ชั่วนี้ มีความสำคัญมาก ๆ เช่นกัน หากจะพูดให้เห็นภาพแล้วจะถือว่า “วันและเวลา” ส่วนใหญ่อาจจะไม่เกิดปัญหา คือ จำนวน “นาที” ไม่ตรงกับ “นาที” จริง จากประสบการณ์ที่ผมเจอก็คือ มันจะเร็วกว่าเวลาจริงอยู่ 60 นาที เพื่อไม่ให้ปัญหานี้เกิดขึ้นอีก เราจึงจำเป็นต้องกำหนด TimeZone ให้กับเว็บแอ��波单击ชั่วครับ

คำสั่ง : PHP

ไฟล์ config/main.php

```
return array(
    ...
    'name'=>'Tutorial Yii Framework',
    'language' => 'th',
    'timeZone'=>'Asia/Bangkok', // กำหนด TimeZone เป็น "Asia/Bangkok" (+7)
    ...
)
```

## ตั้งค่า Messages และแสดงข้อความเตือนเป็นภาษาไทย

ปกติแล้วข้อความเตือนจะเป็นภาษาอังกฤษถูกกำหนดเป็นค่าเริ่มต้น หากเราต้องการเปลี่ยน Messages ข้อความเตือนมาเป็นภาษาไทย เราสามารถทำได้ตามตัวอย่างด้านล่างนี้

ภาษาที่ Yii Framework จัดทำไว้ให้เรา เพื่อรับรองการใช้งานนี้ มีอยู่มากมาย ยกตัวอย่างเช่น 'fi', 'zh\_cn', 'zh\_tw', 'ca', 'de', 'el', 'es', 'sv', 'he', 'nl', 'pt', 'vi', 'bg', 'lv', 'sk', 'uk', 'ko\_kr', 'kk', 'cs' เป็นต้น

คำสั่ง : PHP

ไฟล์ config/main.php

```
return array(
    ...
    'components'=>array(
        ...
        'message'=>array(
            'basePath'=>Yii::getPathOfAlias('application.messages'),
        ),
    ),
    ...
)
```



សាខាអ៊ូរុប  
fenics128[at]gmail.com

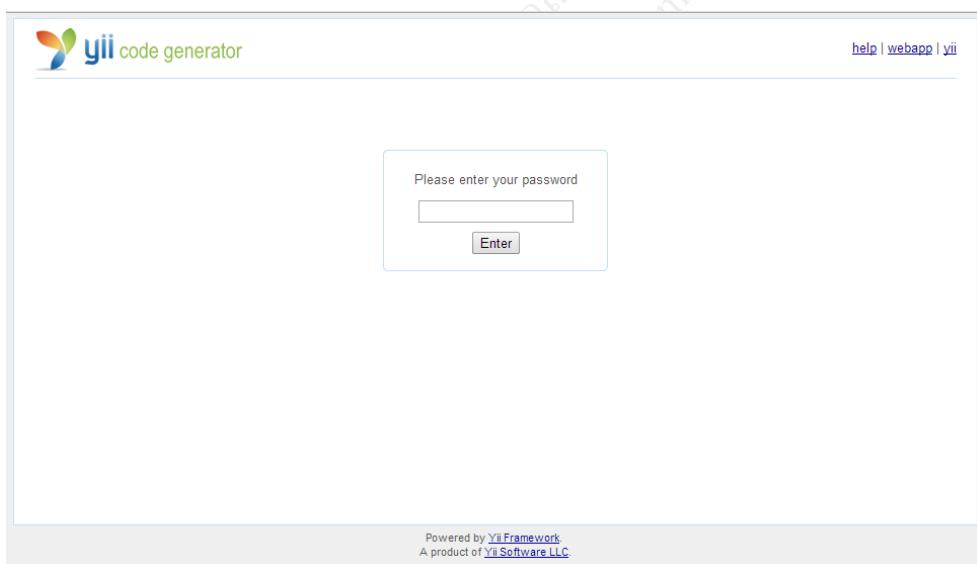
# การพัฒนาเว็บแอปพลิเคชัน ด้วย Yii Framework

## บทที่ 4 วิธีการใช้งาน Gii Code Generator

วิธีการใช้งาน Gii Code Generator ของ Yii Framework ที่ถูกออกแบบเพื่อช่วยให้เราสร้างเว็บแอปพลิเคชัน ด้วย Yii Framework ให้เสร็จได้อย่างรวดเร็วขึ้น ความสามารถของ Gii ยังมีอีกมากมาย อาทิ เช่น การสร้าง Controller Model Modules Form และที่สำคัญที่สุดคือ CRUD (Create-Read-Update-Delete) ไว้ให้เราใช้งาน ผสมจะอธิบายวิธีการใช้งาน Gii ของ Yii Framework แบบละเอียด

### ตรวจสอบระบบ Gii เมื่อต้น

1. เข้าไปที่ URL [http://localhost/tutorial\\_yii/gii](http://localhost/tutorial_yii/gii)
2. หน้าจอ “เข้าสู่ระบบ” ของระบบ Gii Code Generator
3. กรอก “Password” ที่ตั้งค่าไว้ในไฟล์ Config (main.php) ลงไปและกดปุ่ม Enter



4. เมื่อ “login” เป็นที่เรียบร้อยแล้ว จะเข้าสู่หน้า Code Generator ของ Gii System

## ส่วนที่ 1 : Controller Generator

การทำงานในส่วนนี้ จะเป็นการ Generator Controller ออกแบบรูปแบบ MVC (Model-View-Controller) โดยเราไม่ต้องไปสร้างไฟล์ Controller และไฟล์ View ด้วยตัวเองอีกเลย เพียงแค่เราเข้ามาใช้ Gui ของ Yii Framework และกรอกชื่อ Controller ID ที่เราต้องการสร้าง เสร็จแล้วก็กดปุ่ม Preview และปุ่ม Generate ตามลำดับ แค่นี้เราก็จะได้ไฟล์ Controller และไฟล์ View ตามรูปแบบ MVC เรียบร้อยแล้ว ซึ่งจะทำให้การพัฒนาเว็บแอ��เพล็กชันนั้น ทำสำเร็จได้อย่างรวดเร็วกว่าเดิม

The screenshot shows the Yii code generator interface. At the top, there's a navigation bar with links for help, webapp, yii, and logout. On the left, a sidebar titled 'Generators' lists several options: Controller Generator (which is selected and highlighted in blue), Crud Generator, Form Generator, Model Generator, and Module Generator. The main content area is titled 'Controller Generator'. It contains a brief description: 'This generator helps you to quickly generate a new controller class, one or several controller actions and their corresponding views.' Below this, a note says 'Fields with \* are required. Click on the highlighted fields to edit them.' There are three input fields: 'Controller ID \*' (containing 'index'), 'Base Class \*' (containing 'Controller'), and 'Action IDs' (containing 'index'). Below these is a 'Code Template \*' field with the value 'default (D:\xampp\htdocs\yii\framework\generators\controller\templates\default)'. At the bottom of the form is a 'Preview' button. At the very bottom of the page, there's a footer note: 'Powered by Yii Framework A product of Yii Software LLC'.

1. ในช่องของ Controller ID ให้เรากรอกชื่อ Controller ที่เราต้องการสร้าง (ตัวอักษรตัวแรกต้องขึ้นต้นด้วยตัวพิมพ์ใหญ่ๆ ท่านั้น)
2. เมื่อเรากรอกชื่อ Controller ID ที่เราต้องการแล้ว ขึ้นตอนต่อไปเราจะกดปุ่ม Preview เพื่อที่ระบบจะจำลองชื่อไฟล์ที่ระบบจะสร้างให้เราทราบก่อน เมื่อเราต้องการสร้างไฟล์ตามที่ปรากฏแล้ว
3. ขั้นต่อไป เราจะพิจารณาตัวเลือกปุ่ม Generate เพื่อยืนยันว่า “เราต้องการสร้างไฟล์ Controller” ตามชื่อ Controller ID ที่เราได้กรอกลงไว้ (สมมุติผมต้องการสร้างไฟล์ Controller ชื่อว่า Post และกดปุ่ม Generate) หากสร้างไฟล์สำเร็จ ระบบจะแสดงข้อความว่า “The controller has been generated successfully. You may try it now” เป็นอันว่าสร้างไฟล์สำเร็จแล้ว

4. ทดสอบการสร้าง Controller ได้ที่ลิงค์ URL [http://localhost/tutorial\\_yii/post](http://localhost/tutorial_yii/post) คำว่า post ก็อปปี้ ControllerID ที่เราสร้าง รูปแบบของ URL นี้ จะเป็นแบบ “tutorial\_yii/[controllerId]/[actionIds]/[id]”

5. กรณีที่เราต้องการสร้างไฟล์ Controller ไว้ใน Modules เราต้องสามารถทำได้ เพียงแค่ให้เราเพิ่มชื่อ Modules ที่เราต้องการสร้าง Controller ยกตัวอย่างเช่น ในเว็บแอพฯ ผมได้สร้าง Modules ชื่อว่า “admin” ไว้ แล้วผมต้องการสร้าง Controller เก็บไว้ใน Modules ผมก็แค่เปลี่ยนชื่อ ControllerID ที่เรากรอก จาก “Post” มาเป็น “admin/Post” (ModulesName/ControllerID) เพียงแค่นี้ Gii Generator Controller ก็จะสร้าง Controller เก็บไว้ใน Modules นั้นให้เราแล้วครับ

The screenshot shows a web browser displaying a Yii Framework application. The title bar says "Tutorial Yii Framework". The navigation bar includes links for Home, About, Contact, and Login. Below the navigation bar, the URL "Home > Post" is shown. The main content area has a heading "post/index". A note below the heading says: "You may change the content of this page by modifying the file E:\xampp\htdocs\tutorial\_yii2\protected\views\post\index.php." The background of the page has a watermark-like text "Tutorial" repeated diagonally.

### เพิ่มเติม : การกำหนดสิทธิ์การเข้าถึงภายในไฟล์ Controller

การกำหนด (permission) สิทธิ์การเข้าถึงข้อมูลในส่วนต่าง ๆ ภายใต้การทำงานของ Controllers การกำหนดค่า permission เพื่อใช้กำหนดสิทธิ์การเข้าถึงข้อมูลในแต่ละส่วน โดยจะกำหนดค่าบุคคลใหม่ เข้าใช้ส่วนใหม่ได้ดังนี้ ให้เราเข้าไปกำหนดค่าในตัวของ Controller โดยพึ่งก์ชั้นที่ใช้กำหนดค่ามีอยู่ 2 ตัว คือ filters() และ accessRules()

โดย พึ่งก์ชั้น filters() และ accessRules() นั้นจะถูกสร้างขึ้นอัตโนมัติ เมื่อมีการสร้าง Controller ด้วย Gii CRUD Generater แต่ถ้าสร้างด้วย Gii Controller Generater หรือสร้าง Controller เองให้เราสร้าง 2 พึ่งก์ชั้นนี้ ขึ้นมาด้วยทุกครั้ง ปล. หากเราไม่ต้องการจำกัดสิทธิ์ไม่จำเป็นต้องสร้างครับ

คำสั่ง : PHP (ตัวอย่าง)

ไฟล์存放位置 : [skeleton]/protected/controllers/PostController.php

```
// รูปแบบฟังก์ชัน : filters()
public function filters(){
    return array('accessControl','postOnly + delete');
}

// รูปแบบฟังก์ชัน : accessRules()
public function accessRules(){
    return array(
        array('allow','actions'=>array('index','view'), 'users'=>array('*')),
        array('allow','actions'=>array('create','update'), 'users'=>array('@')),
        array('allow','actions'=>array('admin','delete'), 'users'=>array('admin')),
        array('deny','users'=>array('*')),
    );
}
```

## อธิบายเพิ่มเติม : พื้นที่ชั้น accessRules()

คำสั่ง ‘users’=>array(“\*”)

1. ‘users’=>array(‘admin’) ก็อ ผู้ใช้งานที่เป็น “ผู้ดูแลระบบ”
2. ‘users’=>array(‘@’) ก็อ ผู้ใช้งานที่ “กำลังออนไลน์” (ล็อกอินอยู่)
3. ‘users’=>array(‘\*’) ก็อ ผู้ใช้งานทั่วไป หรือ ผู้ใช้งานไม่ได้ล็อกอิน

คำสั่ง ‘actions’=>array(‘index’,’view’)

array(‘index’,’view’) ก็อ Action ที่กำหนดสิทธิเข้าใช้งาน ซึ่งความสามารถกำหนดได้ตัวเองได้จากตัวอย่างจะเป็น actionIndex() และ actionView() เป็นสิทธิของ ผู้ใช้งานทั่วไป

ตัวอย่าง “การกำหนดสิทธิ์การเข้าใช้งาน Controller เป็นต้น”

ตัวอย่าง : ห้ามให้ทุกคนใช้งาน Controller นี้ ยกเว้น admin กับ user1

```
public function accessRules(){  
    return array(  
        array('allow','users'=>array('admin','user1')),  
        array('deny', 'users'=>array('*')),  
    );  
}
```

ตัวอย่าง : อนุญาตให้ทุกคน เข้าใช้งาน Controller นี้ได้

```
public function accessRules(){  
    return array(  
        array('allow','users'=>array('*')),  
        array('deny','users'=>array('*')),  
    );  
}
```

ตัวอย่าง : อนุญาตเฉพาะสมาชิกที่มีการ “ล็อกอินสำเร็จ” ถึงจะเข้าใช้งาน Controller นี้ได้

```
public function accessRules(){  
    return array(  
        array('allow','users'=>array('@')),  
        array('deny', 'users'=>array('*')),  
    );  
}
```

ตัวอย่าง : อนุญาตเฉพาะสมาชิก ใช้งาน action ชื่อ index, update และ delete ได้

```
public function accessRules(){
    return array(
        array('allow', 'actions'=>array('index','update','delete'), 'users'=>array('@')),
        array('deny', 'users'=>array('*')),
    );
}
```

ตัวอย่าง : คัดแปลง เช่น ให้อนุญาตเฉพาะ user ที่อยู่ใน group ที่ต้องการ (สมมุติ ผู้กำหนด group\_type = 1 เป็น Manager, group\_type = 0 เป็น User ธรรมดา) จากโค้ดด้านล่างก็จะได้ว่า เฉพาะ สมาชิกที่เป็น manager ท่านนี้ สามารถใช้งาน action delete ได้

```
public function accessRules(){
    return array(
        array('allow',
            'actions'=>array('delete'),
            'expression'=>'$user->group_type ==1',
        ),
        array('deny',
            'users'=>array('*'),
        ),
    );
}
```

\*\*\* หาก group\_type นี้ พม สมมุติขึ้นมา ควรจะทำแบบผມกีอ่ายลีม ไปสร้างส่วนที่ดึง group\_type จาก db มาด้วยนะครับ

\*\*\* แนะนำให้สร้างเป็น Component Class เช่น สร้างไฟล์ชื่อว่า DriveSoftCenter.php ไว้ในโฟล์เดอร์ [skeleton]/components/DriveSoftCenter.php

```
class DriveSoftCenter extends CApplicationComponent {
    public static function checkUserGroup(){
        $user = User::model()->findByPk(Yii::app()->user->id);
        return $user->group_type; // ในตารางต้องมี field "group_type" ด้วยนะครับ
    }
}
```

วิธีเรียกใช้งาน Component Class || ไฟล์ DriveSoftCenter.php

```
'expression'=>'DriveSoftCenter::checkUserGroup()',
```

## ส่วนที่ 2 : Crud Generator

Crud ย่อมาจาก (Create - Read - Update - Delete) คือ ส่วนที่ทำให้การ Generate โค้ดให้สามารถเพิ่ม ลบ แก้ไข ข้อมูลได้ โดยที่ไม่ต้องเขียนโค้ดเอง ซึ่งจะสามารถใช้ Crud Generate ที่ต้องการแล้วท่านนั้น ซึ่ง Crud Generate มีรูปแบบการแสดงผล

The screenshot shows the Yii code generator interface. At the top, there's a navigation bar with the Yii logo and links for help, webapp, yii, and logout. On the left, a sidebar titled 'Generators' lists 'Controller Generator', 'Crud Generator' (which is selected), 'Form Generator', 'Model Generator', and 'Module Generator'. The main area is titled 'Crud Generator' and contains the following fields:

- 'Model Class \*' (input field)
- 'Controller ID \*' (input field)
- 'Base Controller Class \*'
  - 'Controller' (highlighted in yellow)
- 'Code Template \*'
  - 'default (D:\xampp\htdocs\yii\framework\generators\crud\templates\default)' (highlighted in yellow)

At the bottom of the form are 'Preview' and 'Generate' buttons. Below the form, a footer notes 'Powered by [Yii Framework](#). A product of [Yiisoft LLC](#)'.

\*\*\* ก่อนทำการสร้าง Crud Generate ได้ เราต้องสร้าง Model ก่อน ศึกษาส่วนที่ 4 Model Generate

1. ให้พิมพ์ชื่อ Model Class ที่เราได้สร้างไว้ตามด้วยชื่อ Controller ID ที่เราต้องการสร้าง จากดัวอย่าง สมมุติว่าผมได้สร้าง Model Class ที่ชื่อว่า “Account” ไว้แล้ว และ Controller ID ที่ผมจะสร้างชื่อว่า “Account” จากนั้นให้กดปุ่ม Preview

2. ระบบจะแสดงข้อมูลไฟล์ที่ระบบต้องการสร้างให้เราทราบ เมื่อเรารับทราบแล้ว เรายกกดปุ่ม Generate เพื่อสร้างไฟล์นั้นออกมากจริง ๆ

3. หากสร้างไฟล์สำเร็จ ระบบจะแสดงข้อความว่า “The controller has been generated successfully. You may try it now” เป็นอันว่าสร้างไฟล์สำเร็จแล้ว

4. กรณีที่เราสร้าง Modules ชื่อว่า “admin” เราต้องกรอกข้อมูล Model Class จากเดิม คือ “Account” เป็น “application.models.Account” (application.models.[modelName]) และเปลี่ยนข้อมูล Controller ID จาก “Account” มาเป็น “admin/Account” ด้วย

### ส่วนที่ 3 : Form Generator

ส่วนนี้จะสามารถสร้างหน้าตาของแบบฟอร์มที่ใช้ในการกรอกข้อมูล โดยจะถูกเรียกใช้โดย Controller รูปแบบฟอร์มที่ Generate ออกมานั้น จะมีรูปแบบมาจาก Model ที่ได้สร้างไว้ซึ่ง Form Generator มีรูปแบบการแสดงผล ดังนี้

\*\*\* ก่อนทำการสร้าง Form Generate ได้ เราต้องสร้าง Model ก่อน ศึกษาส่วนที่ 4 Model Generate

1. ให้พิมพ์ชื่อ Model Class ที่เราได้สร้างไว้ ตามด้วยชื่อ View Name ที่เราต้องการสร้างด้วย จากตัวอย่างสมมุติว่าผมได้สร้าง Model Class ชื่อว่า “Account” ไว้แล้ว และ View Name ที่ผมจะสร้างชื่อว่า “account/addForm” จากนั้นให้กดปุ่ม Preview

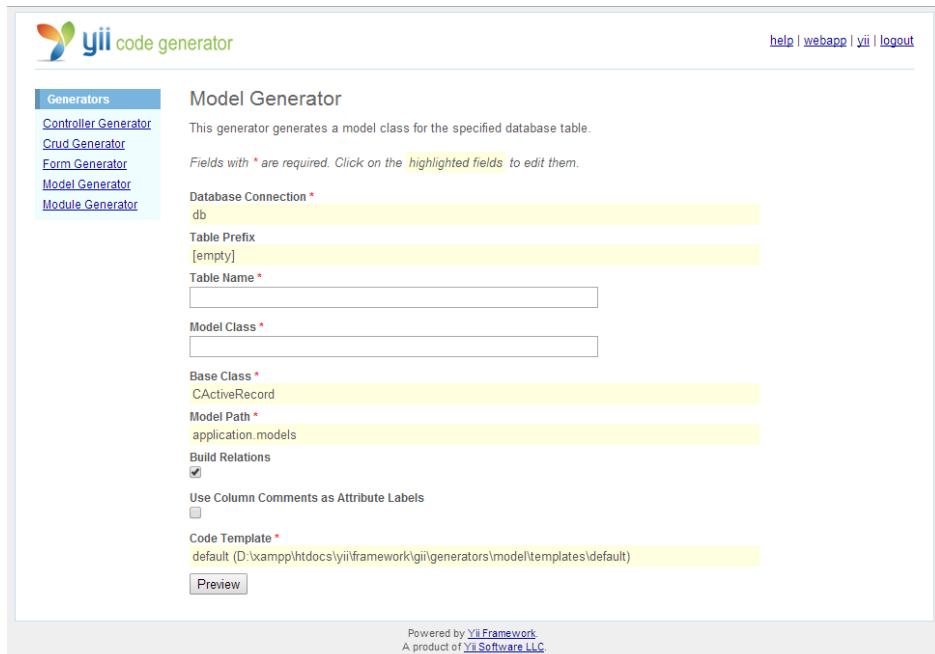
2. ระบบจะแสดงข้อมูลไฟล์ที่ระบบต้องการสร้างให้เราทราบ เมื่อเรารับทราบแล้ว เรายกกดปุ่ม Generate เพื่อสร้างไฟล์นั้นออกมายังริง ๆ

3. หากสร้างไฟล์สำเร็จแล้ว ระบบจะแสดงข้อความว่า “The form has been generated successfully” เป็นอันว่าสร้างไฟล์สำเร็จแล้ว

4. กรณีที่เราสร้าง Modules ชื่อว่า “admin” เรายกต้องกรอกข้อมูล Model Class จากเดิม คือ “Account” เปลี่ยนมาเป็น “application.models.Account” และเปลี่ยนชื่อ View Name จาก “addForm” มาเป็น “admin/account/addForm” ด้วย

5. พอยังจากสร้าง Form สำเร็จแล้ว ระบบจะแสดงคำสั่งของ actionAddForm() ออกมานะ ให้เรานำ actionAddForm() ที่ได้นำไปวางไว้ใน Controller ที่เราเรียกใช้งาน Form ดังกล่าว

ส่วนที่ 4 : Model Generator || ดาวโหลดไฟล์ SQL ตาราง Account ได้ที่นี่ <http://goo.gl/i41G1T>  
 ส่วนที่ทำการเชื่อมต่อกับฐานข้อมูล โดยการสร้าง Model นั้น จะมีการอ้างอิงถึงตารางที่อยู่ในฐานข้อมูลด้วย ซึ่ง Model Generator มีรูปแบบการแสดงผล ดังนี้



The screenshot shows the 'Model Generator' page of the Yii Framework. On the left, there's a sidebar with a 'Generators' tab selected, containing links to 'Controller Generator', 'Crud Generator', 'Form Generator', 'Model Generator', and 'Module Generator'. The main area has a title 'Model Generator' and a sub-instruction: 'This generator generates a model class for the specified database table.' Below this, there are several input fields:

- Database Connection \***: A dropdown menu set to 'db'.
- Table Prefix**: A dropdown menu set to '[empty]'.
- Table Name \***: An input field with no value.
- Model Class \***: An input field with no value.
- Base Class \***: A dropdown menu set to ' CActiveRecord'.
- Model Path \***: A dropdown menu set to 'application.models'.
- Build Relations**: A checked checkbox.
- Use Column Comments as Attribute Labels**: An unchecked checkbox.
- Code Template \***: A dropdown menu set to 'default (D:\xampp\htdocs\yii\framework\k\gii\generators\model\templates\default)'.

At the bottom left is a 'Preview' button, and at the bottom right is a note: 'Powered by [Yii Framework](#). A product of [Yiisoft LLC](#)'.

1. ให้พิมพ์ตาราง ตรงช่อง Table Name ที่เราต้องการสร้าง จากตัวอย่างสมมุติว่าผมได้สร้าง Table Name ที่ชื่อว่า “account” ไว้แล้ว กดปุ่ม Preview
2. ระบบจะแสดงข้อมูลไฟล์ที่ Model Generator ต้องการสร้างให้เราทราบ เมื่อเรารับทราบแล้ว เราจึงกดปุ่ม Generate! เพื่อสร้างไฟล์นั้นออกมากirring ๆ
3. หากสร้างไฟล์สำเร็จ ระบบจะแสดงข้อความว่า “The code has been generated successfully” เป็นอันว่าสร้างไฟล์ Model สำเร็จแล้ว

### กำหนด Attribute ให้กับเมธอด Rules ในโมเดล

Rules คือ การกำหนดค่าให้กับ attribute ใน model ว่ามีคุณสมบัติอย่างไรบ้าง อาทิเช่น ในการทำฟอร์มกำหนด username ต้อง require (ห้ามมีค่าว่าง), length (มีตัวอักษรไม่เกิน ?? ตัว), เบอร์โทรศัพท์ต้องเป็นตัวเลขเท่านั้น, ช่องเลือกเพศ ต้องเป็น radiobox เป็นต้น และ การกำหนด attribute ต่าง ๆ ในโมเดลนั้น เราสามารถกำหนดได้ใน เมธอดชื่อว่า rules() ใน Model เท่านั้น นั่นคือ

## รูปแบบของ Rules จะเป็นไปตามนี้

```
array('attribute list','validator name','on'=>'scenario name',...validation parameters...)
```

1. attribute list คือ ชื่อ attribute ที่มีหลาย attribute ให้ใช้เครื่องหมาย (,) เช่น 'name, tel, address'
2. validator name คือ ชื่อชนิด validator หรือ ตัวแปรการกำหนดคุณสมบัติต่าง ๆ
3. validation parameters คือ ค่าของ validator ที่ต้องการกำหนด

ตัวอย่าง : กำหนดให้ต้องใส่ค่า (ห้ามเป็นค่าว่าง)

```
array('id,name','required'), // ใน (Field) id, name ห้ามเป็นค่าว่าง
```

ตัวอย่าง : กำหนดให้ค่าต้องไม่มีช้ำ

```
array('id,name','unique'), // ใน (Field) id, name ค่าต้องไม่มีซ้ำกันกับค่าในฐานข้อมูล
```

ตัวอย่าง : กำหนดให้มีจำนวนตัวอักษร ไม่เกิน 128 ตัวอักษร

```
array('name','length','max'=>128), // ใน (Field) name มีจำนวนตัวอักษรไม่เกิน 128 ตัวอักษร
```

ตัวอย่าง : กำหนดให้ต้องมีจำนวนตัวอักษร 10 ตัวอักษร

```
array('name','length','is'=>10), // ใน (Field) name มีจำนวนตัวอักษร 10 ตัวอักษร
```

ตัวอย่าง : กำหนดให้ต้องมีตัวอักษร ไม่น้อยกว่า 8 ตัวอักษร

```
array('name','length','min'=>8), // ใน (Field) name ต้องมีจำนวนตัวอักษรไม่น้อยกว่า 8 ตัวอักษร
```

ตัวอย่าง : กำหนดให้ต้องมีค่าเหมือนกับ Field (ใช้เวลาสมัครสมาชิกแล้วให้ใส่ password 2 ครั้ง)

```
// ใน (Field) password1 มีค่าเหมือนกับ (Field) password2
```

```
array('password1','compare','compareAttribute'=>'password2'),
```

ตัวอย่าง : กำหนดให้ต้องเป็นไปตาม รูปแบบที่กำหนด

```
// ใน (Field) name กำหนดรูปแบบ pattern และ ระบุข้อความเตือน
```

```
array('name','match','pattern'=>'^[\\w\\s]+$', 'message'=>'คุณไม่ได้กรอกข้อมูลตามรูปแบบที่กำหนด')),
```

ตัวอย่าง : กำหนดให้ค่าต้องเป็น 1,2,3 เท่านั้น

```
array('name','in','range'=>array(1,2,3)), // ใน (Field) name ต้องมีค่าเป็น 1,2,3 เท่านั้น
```

ตัวอย่าง : กำหนดให้ค่าต้องเป็น ตัวเลข เท่านั้น

```
array('id','numerical','integerOnly'=>true), // ใน (Field) id ต้องมีค่าเป็น ตัวเลข เท่านั้น
```

ตัวอย่าง : กำหนดให้ค่าต้องเป็น true, false เท่านั้น

```
array('type','boolean'), // ให้ (Field) type ต้องมีค่าเป็น true,false เท่านั้น
```

ตัวอย่าง : กำหนดให้ค่าต้องเป็น วันที่หรือเวลา เท่านั้น

```
array('date','date','format'=>'yyyy-M-d H:m:s'), // ให้ (Field) date ต้องมีค่า วันที่หรือเวลาเท่านั้น
```

ตัวอย่าง : กำหนดให้ Field ที่เป็นไฟล์สำหรับอัปโหลด ต้องเป็นไปตามที่กำหนด

```
array(  
    'files','file','types'=>'jpg,gif,png', // ให้ (Field) files กำหนดประเภท file ชนิด jpg, gif, png  
    'wrongType'=>'Only jpg, gif, png allowed','allowEmpty'=>true, // ข้อความเตือน  
    'maxSize'=>1024*1024*5,'tooLarge'=>'ขนาดไฟล์ไม่เกิน 5MB' // ขนาดไฟล์รูป  
,
```

## ส่วนที่ 5 : Module Generator

หน่วยของเฟร์ร์ในตัวเองที่ประกอบด้วย models, views, controllers และส่วนประกอบที่สนับสนุนอื่นๆ ในหลาย ๆ ด้าน โดยคุณลักษณะกับโปรแกรม แม่กิจกรรมแตกต่างที่สำคัญ คือ โดยจะไม่สามารถใช้งานได้ ตัวคุณเดียวและมันจะต้องอยู่ภายใต้แอพพลิเคชั่น ผู้ใช้งานเข้าถึงการควบคุมโดยคุณเมื่อตอนที่พากษาทำกับ ตัวควบคุมในการใช้งานปกติ

โดยจะมีประโยชน์ในสถานการณ์ต่าง ๆ สำหรับโปรแกรมขนาดใหญ่ โดยอาจจะแบ่งโดยคุณออก เป็นโดยคุณต่าง ๆ แต่ละการพัฒนาและมีการเก็บรักษาไว้ต่างหาก คุณลักษณะบางอย่างที่ใช้กันทั่วไป เช่น การจัดการผู้ใช้ การจัดการความคิดเห็น อาจจะได้รับการในแบบของโดยคุณเพื่อให้สามารถนำกลับมาใช้งาน อย่างง่ายดายในโครงการในอนาคต Modules Generator มีรูปแบบการแสดงผล ดังนี้

The screenshot shows the Yii Code Generator interface. At the top, there's a navigation bar with links for help, webapp, yii, and logout. On the left, a sidebar titled "Generators" lists several options: Controller Generator, Crud Generator, Form Generator, Model Generator, and Module Generator. The "Module Generator" option is selected and highlighted in blue. The main content area has a title "Module Generator". Below it, a sub-instruction says "This generator helps you to generate the skeleton code needed by a Yii module." It also notes that fields marked with an asterisk (\*) are required and provides a link to edit them. A "Module ID" input field is present. Underneath, there's a "Code Template" section with a dropdown menu set to "default (D:\xampp\htdocs\yii\framework\yii\generators\module\templates\default)". At the bottom of this section is a "Preview" button. The footer of the page includes the text "Powered by Yii Framework" and "A product of Yii Software LLC".

1. ให้พิมพ์ชื่อ Modules ID ที่เราต้องการสร้าง จากตัวอย่างสมมุติว่าผมได้สร้าง Modules ID ที่ชื่อว่า “admin” และให้กดปุ่ม Preview

2. ระบบจะแสดงข้อมูลไฟล์ที่ระบบต้องการสร้างให้เราทราบ เมื่อเราấnทរานแล้ว เราจะกดปุ่ม Generate เพื่อสร้างไฟล์นั้นออกมากจริง ๆ

3. หากสร้างไฟล์สำเร็จ ระบบจะแสดงข้อความว่า “The module has been generated successfully” เป็นอันว่าสร้างไฟล์สำเร็จแล้ว

4. หลังจากสร้าง Modules สำเร็จ ระบบจะแสดงคำสั่ง ให้นำคำสั่งนั้นไปวางไว้ในส่วนของ modules ภายในไฟล์ Config (main.php)

คำสั่ง : PHP

ไดเรกทอรี่ : [skeleton]/protected/config/main.php

```
return array(
    .....
    'modules'=>array(
        'admin',
    ),
    .....
);
```

The module has been generated successfully.

To access the module, you need to modify the application configuration as follows:

```
<?php
return array(
    'modules'=>array(
        'admin',
    ),
    .....
);
```





ຄະດີທິດ ນກງານ ເລຂມ  
fenics128[at]gmail.com

## การพัฒนาเว็บแอปพลิเคชัน ด้วย Yii Framework

### บทที่ 5 การเชื่อมต่อฐานข้อมูล

ปัจจุบันการเชื่อมต่อฐานข้อมูล จะเป็นตัวกลางสำคัญที่จะช่วยให้เว็บแอปพลิเคชัน มีความน่าสนใจ นำไปใช้งาน เนื่องจากฐานข้อมูลจะเป็นศูนย์รวมการเก็บรวบรวมข้อมูลต่าง ๆ อาทิข้อมูลสมาชิก ข้อมูลการเข้าใช้งานเว็บไซต์ ข้อมูลหัวข้อมูลต่าง ๆ เป็นต้น ซึ่งข้อมูลเหล่านี้ มักจะเป็นตัวนำเสนอบริการให้เว็บแอปพลิเคชันนั้น ๆ คุณมีชีวิตชีวา นำไปใช้งาน เพราะมันสามารถตอบโต้กับผู้ใช้งานได้ทันที แต่เมื่อเทียบกับเว็บไซต์ธรรมดาที่สร้างขึ้นด้วยภาษา HTML เพียงอย่างเดียว หรือเว็บไซต์ที่นำเสนอบนเฉพาะในด้านของเนื้อหาเพียงอย่างเดียว นั่น ในโลกปัจจุบันนี้ เว็บไซต์ต่าง ๆ เหล่านี้ มีบทบาทก่อ起 น้ำหนัก พัฒนาเว็บแอปฯ เอาละครับ นำมาเข้าเรื่องกันเลยดีกว่า เริ่มต้นด้วยหัวข้อแรกกันเลยละกันนะครับ

#### การเชื่อมต่อฐานข้อมูล MySQL

การเชื่อมต่อฐานข้อมูล ด้วยความสามารถของ Yii Framework ที่ถูกออกแบบมาให้สามารถรองรับกับฐานข้อมูลหลาย ๆ แบบ หรือจะบอกว่า เกือบทั้งหมดของฐานข้อมูลเลยก็ว่าได้ แต่เราที่จะมาเรียนรู้วิธีการเชื่อมต่อฐานข้อมูลนั้น ในบทที่ 3 การตั้งค่า (Config) ผ่านไฟล์ config ที่มีการตั้งค่าการใช้งานฐานข้อมูลไปแล้ว ในหัวข้อนี้ อาจจะอธิบายเพิ่มเติมอีกเพียงนิดหน่อย อย่างเช่น บทที่ 3 ผ่านตั้งค่าการเชื่อมต่อกับฐานข้อมูล MySQL แต่ในหัวข้อนี้ อาจจะอธิบายการตั้งค่าเชื่อมต่อ กับฐานข้อมูลประเภทอื่น ๆ คุ้นหู เช่น Oracle MySQL PostgreSQL MySQLi SQLite และ MySQLi-PDO ที่จะมาอธิบายในบทถัดไป

เอาละครับ ผ่านมาเป็นตัวอย่างก่อนนะครับ (เพราะคิดว่าจะเป็นอะไรที่คุณนิยมใช้มากสุด) อันดับแรกเลย ขอให้คุณไปสร้างฐานข้อมูลไว้ตัวนึง (ใช้คำสั่ง SQL สร้างหรือจะสร้างด้วย phpMyAdmin หรืออะไรก็ได้แล้วแต่คุณ) เนื่องจากในบทที่ 3 การตั้งค่า ผ่านไฟล์ config ทำการตั้งค่าฐานข้อมูล เพื่อติดต่อฐานข้อมูลประเภท MySQL เอาไว้

#### การเชื่อมต่อฐานข้อมูลต่าง ๆ นอกเหนือจาก MySQL

หากเราไม่ใช้ฐานข้อมูลประเภทMySQL สามารถพิมพ์คำสั่งด้านล่างลงในไฟล์ Config (main.php) โดยเปลี่ยนคำสั่ง mysql:host=localhost;dbname=dbTutorial\_yii เป็นคำสั่งตามฐานข้อมูลที่เราต้องการเรียกใช้

```
SQLite: sqlite:/path/to/dbfile  
MySQL: mysql:host=localhost;dbname=testdb  
PostgreSQL: pgsql:host=localhost;port=5432;dbname=testdb  
SQL Server: mssql:host=localhost;dbname=testdb  
Oracle: oci:dbname=/localhost:1521/testdb
```

## การเขื่อมต่อฐานข้อมูลแบบ Multiple-database ที่สนับสนุนใน Yii Framework

โดยปกติแล้ว การใช้งาน Database แบบทุกเว็บจะมีการเขื่อมต่อ ฐานข้อมูลเพียงฐานข้อมูลเดียว แต่สำหรับผู้ผลิต เว็บแอพฯ ของผู้ต้องมีการเขื่อมต่อไปยังฐานข้อมูลอื่นในเครื่อง Server อื่นด้วย ผู้ซึ่งศึกษาค้นคว้าหาข้อมูลการตั้งค่าและในที่สุดผมก็เจอกัน ก็เลยกองทำดูปรากฏว่ามันสามารถใช้งานได้ดี ก็เลยมาเขียนบันทึกไว้ว่าวันนี้ขั้นตอนการตั้งค่าอะไรอย่างไรบ้าง

### 1. เข้าไปตั้งค่าไฟล์ Config (main.php) และเพิ่มฐานข้อมูลใหม่ลงไป

คำสั่ง : PHP

ไดเรกทอรี่ : [skeleton]/protected/config/main.php

```
// เพิ่มเข้าไปใหม่
'components'=>array(
    'db2' => array(
        'class' => 'CDbConnection',
        'connectionString'=>'mysql:host=localhost;dbname=db_tutorial_yii_2', // dbTutorial_yii_2
        'emulatePrepare'=gt;true, // อันนี้ sql บางตัวมันเรียกใช้
        'username'=>'root', // ชื่อผู้ใช้งานฐานข้อมูล
        'password'=>'12345', // รหัสผ่านฐานข้อมูล
        'charset'=>'utf8', // character set
    ),
),
```

### 2. สร้างไฟล์ Components ชื่อว่า “RActiveRecord.php” เพื่อใช้ติดต่อฐานข้อมูลตัวที่ 2 ของเรา

คำสั่ง : PHP

ไดเรกทอรี่ : [skeleton]/components/RActiveRecord.php

```
class RActiveRecord extends CActiveRecord {
    private static $db_subMain = null;
    protected static function getSubMainDbConnection(){
        if (self::$db_subMain !== null){
            return self::$db_subMain;
        }else{
            self::$db_subMain = Yii::app()->db2;
            if (self::$db_subMain instanceof CDbConnection){
                Yii::app()->db->setActive(false); // ปิด db
                Yii::app()->db2->setActive(true); // เปิด db2
                return self::$db_subMain;
            }else{
                throw new CDbException(Yii::t('yii','ไม่สามารถติดต่อ “ฐานข้อมูล” ได้'));
            }
        }
    }
}
```

3. แก้ไขไฟล์ Models ที่ต้องการเชื่อมต่อ กับฐานข้อมูลตัวที่ 2 เพียงแค่ทำการเปลี่ยนจาก CActiveRecord เป็น RActiveRecord และเพิ่ม Function getDbConnection() ตามตัวอย่างลงใน Class ModelName ด้านล่างนี้

คำสั่ง : PHP

ไฟล์คร่าวๆ : [skelton]/protected/models/ModelName.php

```
// เปลี่ยนจาก CActiveRecord มาเป็น R ActiveRecord
// class ModelName extends CActiveRecord {
class ModelName extends R ActiveRecord {
    // แทรคคำสั่งนี้ลงในไฟล์ Model
    public function getDbConnection(){
        return self::getSubMainDbConnection();
    }
}
```

## การดึงข้อมูลจากฐานข้อมูล ด้วยวิธีการแบบต่าง ๆ

วิธีการดึงข้อมูลจากตาราง (Table) ในฐานข้อมูลออกแบบเป็นตารางแบบ TbGridView (Widget) การที่เราจะเอาข้อมูล (Data) ในฐานข้อมูลของเราไปแสดงใน Widget TbGridView ได้นั้น เราจำเป็นที่จะต้องแปลงข้อมูล (Data) ของเรามาให้อยู่ในรูปแบบdataProvider อะก่อน ซึ่งในบทเรียนนี้ ผมขออธิบาย 2 อันที่เป็นที่นิยมใช้กัน เอาละเรามาดูกัน กับ คำสั่ง 2 ตัวนี้กันเดখครับ

1. Data Access Object (DAO) :: CSqldataProvider
2. Active Record (AR) :: CActiveDataProvider

\*\*\*\* ให้เราสร้าง Controller ชื่อว่า “member” เพื่อใช้สำหรับเขียนคำสั่งทดสอบการ “ดึงข้อมูล”

ดาวโหลดไฟล์ .SQL ตาราง Account และ Profile ประกอบ || ได้ที่นี่ <http://goo.gl/KSDnfs>

### Data Access Object (DAO) : CSqldataProvider

การใช้งานฐานข้อมูลแบบธรรมด้า สร้างอยู่บนพื้นฐาน PHP Data Object (PDO) ที่เป็นส่วนขยายสำหรับการเข้าถึงข้อมูลใน DBRMS เช่น MySQL เป็นต้น

#### DAO Class ประกอบด้วย

1. CDbConnection สำหรับจัดการ Database Connection
2. CDbCommand สำหรับประมวลผลของชุดคำสั่ง SQL
3. CDbDataReader สำหรับดึงข้อมูลจาก Result Set
4. CDbTransaction สำหรับจัดการ Transaction

\*\* หมายเหตุ DAO Class เวลาต้องการใช้งาน ในไฟล์ php.ini จะต้องเปิดใช้งาน Modules php\_mysql, php\_pdo.dll, php\_pdo\_mysql.dll ไว้ด้วย

## ตัวอย่าง : การใช้งานเบื้องต้น

```
/** ตั้งข้อมูลการเชื่อมต่อจากไฟล์ config/main.php **/
$connection = Yii::app()->db;
/** หรือ ระบุการเชื่อมต่อโดยตรง */
$connection = new CDbConnection (
    'mysql:host=localhost;dbname=db_turotial_yii', 'root','12345'
);
/** เปิด Connection */
$connection->active = true;
/** การตั้งข้อมูล */
$sql = 'SELECT * FROM tableName'; // คำสั่ง SQL
$command = $connection->createCommand($sql); // สร้าง SQL Command
$dataReader = $command->query(); // Query ข้อมูลออกมา
foreach ($dataReader as $row){
    echo $row['column_name'] . ' - ' . $row['column_name'];
}
/** ปิด Connection */
$connection->active = false;
```

## ตัวอย่าง : การจัดการ Transaction เบื้องต้น

```
$transaction=$connection->beginTransaction();
try{
    $connection->createCommand($sql1)->execute();
    $connection->createCommand($sql2)->execute();
    $transaction->commit();
} catch(Exception $e) {
    $transaction->rollBack();
}
```

## มาเริ่มกันเลย

ก่อนอื่นเราต้องมีการจัดเตรียมฐานข้อมูลไว้ทั้งหมด 2 ตาราง ได้แก่ ตาราง account และตาราง profile โดย 2 ตาราง ซึ่งสามารถ Import ไฟล์ .SQL (<http://goo.gl/KSDnfs>) ที่ดาวโหลดมาลงในฐานข้อมูลของเราได้เลย จะมีความสัมพันธ์ แบบ 1 ต่อ 1 (ONE TO ONE)

### สิ่งที่ต้องทำ

- สร้าง Model ให้กับตาราง “Account” และ ตาราง “Profile”
- สร้าง Controller ชื่อว่า “CSqlDataProvider” (CSqlDataProviderController.php)
- ถ้าสร้างตามข้อ 1-2 พร้อมแล้ว มาลูบกันได้เลย

ทดสอบ : ให้แก้ไขฟังก์ชัน actionIndex() ตามคำสั่งด้านล่างนี้

คำสั่ง : PHP

ไฟร์เรคทอรี่ : [skeleton]/protected/controllers/CSqlDataProviderController.php

```
class CSqlDataProviderController extends Controller {
    public function actionIndex() { // แก้ไขฟังก์ชันนี้
        $sql = "SELECT * FROM account ORDER BY id ASC"; // คำสั่ง SQL ที่ต้องการ
        $dataProvider = new CSqlDataProvider($sql,array( // เอ้า sql แปลงเป็น dataProvider
            'pagination'=>array(
                'pageSize'=gt;10, // กำหนดให้นานนึงแสดง 10 รายการ
            ),
        ));
        $this->render('index',array(
            'dataProvider'=>$dataProvider, // ส่งตัวแปรไปยัง view
        ));
    }
}
```

ทดสอบ : เพิ่มคำสั่ง GridView (Widget)

คำสั่ง : PHP

ไฟร์เรคทอรี่ : [skeleton]/protected/views/cSqlDataProviderController/index.php

```
$this->widget('zii.widgets.grid.CGridView', array(
    'id'=>'grid_1', // ชื่อ GridView
    'dataProvider'=>$dataProvider, // ตัวแปร data ที่มีข้อมูล
    'enablePagination'=>true, // กำหนดให้แสดงปุ่มเปลี่ยนหน้า
));
```

\*\*\* เอ้า Code ด้านบนไปใส่ตรงที่ต้องการแสดงตาราง สำหรับ Code นี้ เป็น Widget ที่รับค่าตัวแปร \$dataProvider และนำมาสร้างเป็นตาราง GridView

ผลลัพธ์ : GridView (Widget) || ให้เข้าไปที่ [http://localhost/tutorial\\_yii/CSqlDataProvider](http://localhost/tutorial_yii/CSqlDataProvider)

Displaying -3-0 of 1 result.

id	pid	username	password
1	1	user1	pass1
2	2	user2	pass2
3	3	user3	pass3
4	4	user4	pass4

## JOIN Table : Data Access Object (DAO)

การ Join Table เราจะใช้งานเมธอด CSqlDataProvider นี้ และเขียนคำสั่ง MySQL ในการเชื่อมต่อกับฐานข้อมูล คงไม่มีอะไรแปลกใหม่สำหรับคนที่เคยเขียน PHP + MySQL เชื่อมต่อกับฐานข้อมูล แต่คราวนี้ เคยเขียน PHP ธรรมดามาก่อนแล้ว (แนะนำไปศึกษาคำสั่ง MySQL มา ก่อน)

### คำสั่ง MySQL : Left Join

การ Join ตารางแบบ Left Join เป็นการค้นหาข้อมูลจาก 2 ตาราง โดยยึดตารางทางด้านซ้าย (account) เป็นหลัก ซึ่งข้อมูลในตาราง account มีข้อมูลทั้งหมด 4 แถวข้อมูล และตาราง profile มีข้อมูลทั้งหมด 5 แถวข้อมูล เรามาลองเปลี่ยนคำสั่ง MySQL มาเป็นดังนี้

```
"SELECT account.*, profile.* FROM account LEFT JOIN profile ON account.pid = profile.id ORDER BY account.id ASC"
```

### คำสั่ง MySQL : Right Join

การ Join ตารางแบบ Right Join เป็นการค้นหาข้อมูลจาก 2 ตาราง โดยยึดตาราง ทางขวา (profile) เป็นหลัก ซึ่งข้อมูลในตาราง account มีข้อมูลทั้งหมด 4 แถวข้อมูล และตาราง profile มีข้อมูลทั้งหมด 5 แถว ข้อมูล เรามาลองเปลี่ยนคำสั่ง MySQL มาเป็นดังนี้

```
"SELECT account.*, profile.* FROM account RIGHT JOIN profile ON account.pid = profile.id ORDER BY account.id ASC"
```

### คำสั่ง MySQL : Inner Join

การ Join ตารางแบบ Inner join ข้อมูลที่จะอุปกรณ์ต้องมีทั้งในตาราง ซ้ายและขวา แสดงข้อมูลอุปกรณ์ทั้งหมด ตามตัวอย่าง เราจะค้นหาข้อมูล account.pid = profile.id หากพบข้อมูลก็จะนำอุปกรณ์แสดง แต่ถ้าไม่พบข้อมูล ก็จะไม่แสดง ทดสอบคำสั่งดังนี้

```
"SELECT account.*, profile.* FROM account INNER JOIN profile ON account.pid = profile.id ORDER BY account.id ASC"
```

## Active Record (AR) : CActiveDataProvider

การใช้งานแบบใช้ฟังก์ชัน relation ใน AR และสร้าง relation ให้กับ Database ก่อนอื่นเราจะต้องไปสร้าง Model ของ account และ profile ให้ไปที่ Model Generator ใน Gii System ก่อน

ก่อนอื่นเราต้องมีการจัดเตรียมฐานข้อมูลไว้ทั้งหมด 2 ตาราง ได้แก่ ตาราง account และตาราง profile โดย 2 ตาราง ซึ่งสามารถ Import ไฟล์ .SQL (<http://goo.gl/KSDnfs>) ที่ดาวโหลดมาลงในฐานข้อมูลของเราได้เลย จะมีความสัมพันธ์ แบบ 1 ต่อ 1 (ONE TO ONE)

## สิ่งที่ต้องทำดังต่อไปนี้

1. สร้าง Model ให้กับตาราง “Account” และ ตาราง “Profile”
2. สร้าง Controller ชื่อว่า “CActiveDataProvider” (CActiveDataProviderController.php)
3. ถ้าสร้างตามข้อ 1-2 พร้อมแล้ว มาลุยกันได้เลย

ทดสอบ : ให้แก้ไขฟังก์ชั่น actionIndex ตามคำสั่งด้านล่างนี้

คำสั่ง : PHP

ไฟล์存放位置 : [skelton]/protected/controllers/CActiveDataProviderController.php

```
class CActiveDataProviderController extends Controller {
    public function actionIndex() { // แก้ไขฟังก์ชั่นนี้
        $dataProvider = new CActiveDataProvider('Profile', array( // แปลงเป็น dataProvider
            'pagination'=>array(
                'pageSize'=>10, // กำหนดให้หน้านึงแสดง 10 รายการ
            ),
        ));
        $this->render('index',array(
            'dataProvider'=>$dataProvider, // ส่งตัวแปรไปยัง view
        ));
    }
}
```

ทดสอบ : เพิ่มคำสั่ง GridView (Widget)

คำสั่ง : PHP

ไฟล์存放位置 : [skelton]/protected/views/cActiveDataProvider/index.php

```
$this->widget('zii.widgets.grid.CGridView', array(
    'id' => 'grid_1', // ชื่อ GridView
    'dataProvider' => $dataProvider, // ตัวแปร data ที่มีข้อมูล
    'enablePagination' => true, // กำหนดให้แสดงปุ่มเปลี่ยนหน้า
));
```

ผลลัพธ์ : GridView (Widget) || ให้เข้าไปที่ URL [http://localhost/tutorial\\_yii/CActiveDataProvider](http://localhost/tutorial_yii/CActiveDataProvider)

Displaying 1-5 of 1 result.		
ID	Firstname	Lastname
1	Peter	Wang
2	Sarah	Mary
3	Johan	Willie
4	Wane	Oper
5	Null	Null

## Join Model : Active Record (AR)

วิธีการใช้งาน Active Record ที่ Yii Framework ทำการออกแบบมาให้เราได้ใช้งานกัน ในส่วนนี้ AR ถูกออกแบบให้ใช้ร่วมกันกับ Model จะอธิบายให้เข้าใจง่าย ๆ หน่อยก็คือ หากเราต้องการใช้งาน AR เราต้องทำการสร้าง Model ออกแบบก่อน เรายังจะใช้งาน AR ได้

### เริ่มต้น : Join Model

จากตัวอย่าง คือ ผู้ต้องการใช้ตาราง “Account” เชื่อมต่อไปยังตาราง “Profile” โดยใช้ (Field) pid:FK ของตาราง “Account” อ้างอิงไปยัง (Field) id:PK ของตาราง “Profile” ดังนั้นจะเข้าไปแก้ไข ฟังก์ชัน relations() ในไฟล์ Model Account ตามคำสั่งดังต่อไปนี้

คำสั่ง : PHP

ไฟร์เซตอรี่ : [skeleton]/protected/models/Account.php

```
public function relations() { // ฟังก์ชัน relations() อยู่ใน Model Account
    return array(
        'myprofile'=>array(self::BELONGS_TO,'profile','pid'),
    );
}
```

### คำอธิบาย : Join Model

- ‘myprofile’ :: ชื่ออะไรก็ได้ (ที่เราต้องการตั้ง) ข้อควรระวัง อย่าให้ชื่อความสัมพันธ์นี้ ไปซ้ำกับชื่อแต่ละฟิลด์ของ table นะครับ เวลาเรียกใช้มันจะไม่ได้
- ‘self::BELONGS\_TO’ :: กำหนดรูปแบบการเชื่อมต่อ จากตัวอย่างจะเป็น relations 1:1
- ‘profile’ :: ชื่อ Model ที่เราต้องการเชื่อมต่อ
- ‘pid’ :: ชื่อ (Field) pid:FK ของตาราง “Account” อ้างอิงไปยัง PK ของตาราง “Profile”

### Relations : การสร้างความสัมพันธ์ (relations) ระหว่าง Model ด้วยกันเอง

1. **HAS\_ONE / HAS\_MANY** : ใช้ใน Model ที่มี Primary Key และมี Model อื่น ใช้ Foreign Key อ้างมาหา โดยฟิลด์ที่ใช้เป็นฟิลด์ที่เป็น Foreign Key เช่น จากตัวอย่าง Model Profile ใช้ id ซึ่งเป็น Foreign Key อ้างอิงมาหา id ของ Model Account ซึ่งเป็น Primary Key นั่นเอง และการคืนค่า (return) HAS\_ONE จะคืนค่าออกแบบแค่ 1 ค่าเท่านั้น แต่ HAS\_MANY จะคืนค่าออกแบบเป็น Array เท่านั้น

2. **BELONGS\_TO** : ใช้ใน Model ที่มี Foreign Key อ้างไปที่ Primary Key ของ Model อื่น เช่น Model Account ที่มี id ซึ่งเป็น Foreign Key ต้องการอ้างอิงไปยัง id ซึ่งเป็น Primary Key ของ Model Profile

ทดสอบ : ให้แก้ไขฟังก์ชัน actionIndex ตามคำสั่งด้านล่างนี้

คำสั่ง : PHP

ไดเรคทอรี่ : [skeleton]/protected/controllers/CActiveDataProviderController.php

```
public function actionIndex() {
    // Account คือ Model ที่ใช้ Gii สร้าง
    $dataProvider = new CActiveDataProvider('Account',array(
        'criteria'=>array(
            'with'=>'myprofile' // เรียก relation ชื่อ myprofile ใน Model Account
        ),
        'pagination'=>array(
            'pageSize'=gt;10, // จำนวนรายการที่แสดงต่อหน้า
        ),
    )
);
$this->render('index',array(
    'dataProvider'=>$dataProvider, // ส่งข้อมูลไปยัง View
));
}
```

ทดสอบ : เพิ่มคำสั่ง GridView (Widget)

คำสั่ง : PHP

ไดเรคทอรี่ : [skeleton]/protected/views/cActiveDataProvider/index.php

```
$this->widget('zii.widgets.grid.CGridView', array(
    'id'=>'grid_1', // ชื่อตาราง
    'dataProvider'=>$dataProvider, // ตัวแปร data ที่มีข้อมูล
    'enablePagination'=>true, // กำหนดให้แสดงปุ่มเปลี่ยนหน้า
    'columns'=>array( // กำหนด column ที่จะแสดง
        'id','pid','username','password',// (Field) id, pid ,username, password ของตาราง Account
        'myprofile.firstname', // (Field) firstname' ของตาราง Profile
        'myprofile.lastname', // (Field) lastname' ของตาราง Profile
    ),
));
```

ผลลัพธ์ : GridView (Widget) || ให้เข้าไปที่ URL [http://localhost/tutorial\\_yii/CActiveDataProvider](http://localhost/tutorial_yii/CActiveDataProvider)

ID	Pid	Username	Password	Firstname	Lastname
1	1	user1	pass1	Peter	Wang
2	2	user2	pass2	Sarah	Mary
3	3	user3	pass3	Johan	Willie
4	4	user4	pass4	Wane	Oper

Displaying 1-4 of 1 result.

**เพิ่มเติม :** การเขียนความสัมพันธ์ เพื่อเชื่อมโยง โมเดล 2 โมเดล เข้าด้วยกัน

จัดรูปแบบความสัมพันธ์ของฐานข้อมูล ระหว่าง Model กับ Model ด้วยกันให้อยู่ในรูปแบบของ Active Record (AR) :: CActiveDataProvider นั้นจะมีการเขียนความสัมพันธ์ เพื่อเชื่อมโยง โมเดล 2 โมเดล เข้าหากัน โดยรูปแบบโครงสร้าง จะเขียนไว้ใน model ที่ต้องการ

ตัวอย่าง : รูปแบบ Active Record

```
'VarName'=>array('RelationType', 'ClassName', 'ForeignKey', ...additional options)
```

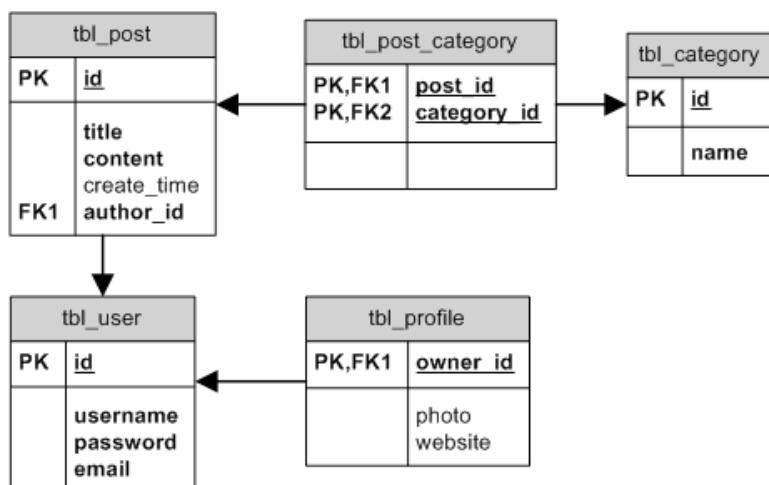
คำอธิบายเพิ่มเติม :

1. VarName เป็นชื่อ อะไรก็ได้ ตั้งมาเพื่อเวลาเราเอาไปเรียกความสัมพันธ์นี้
2. RelationType เป็น ชนิดของ ความสัมพันธ์
3. ClassName เป็นชื่อ class model ที่จะไปสัมพันธ์ด้วย
4. ForeignKey เป็นชื่อ field ที่ใช้เชื่อมโยงหรืออ้างอิง 2 ตารางเข้าด้วยกัน

โดยชนิดของความสัมพันธ์

1. BELONGS\_TO = เป็นของ....
2. HAS\_MANY = มีหลายอัน (1 to many)
3. HAS\_ONE = มีแค่หนึ่ง (1 to 1)
4. MANY\_MANY = 1 อัน มีหลายอย่าง แต่ละอย่างอาจเป็นของหลายอัน (many to many)

**ER-Diagram :** ตัวอย่างจากเว็บ Yii



### เพิ่มเติม : อธิบาย รูปภาพ ER

1. tbl\_user = เก็บข้อมูลผู้ใช้ -> สร้าง Model (โมเดล) ชื่อ User
2. tbl\_profile = เก็บรายละเอียด profile ของแต่ละคน -> สร้าง Model (โมเดล) ชื่อ Profile
3. tbl\_post = เก็บข้อมูลการโพส -> สร้าง Model (โมเดล) ชื่อ Post
4. tbl\_category = เก็บหมวดหมู่ของโพสนั้น -> สร้าง Model (โมเดล) ชื่อ Category
5. tbl\_post\_category = เป็นตารางที่เอาไว้เชื่อมความสัมพันธ์แบบ many to many เท่านั้นครับ

### ความสัมพันธ์แบบ HAS\_ONE

จาก ER แล้วจะเห็นได้ว่า 1 user จะมีแค่ 1 profile เท่านั้น ดังนั้น ความสัมพันธ์จะเป็น 1 to 1 เราจะใช้ HAS\_ONE ให้เราแก้ไขพงก์ชั้น relations() ดังนี้

```
public function relations() {
    return array(
        'profile' => array(
            self::HAS_ONE, 'Profile', 'owner_id',
        ),
    );
}
```

เมื่อเราเขียนอยู่ในโมเดล ของ User จะเป็นการอ้างว่า โมเดล User นี้จะไปสัมพันธ์กับ โมเดล Profile แบบ HAS\_ONE โดยจะใช้ owner\_id (PK/FK) ของ Profile เป็นตัวอ้างอิงมาซึ่ง id (PK) ของ User

ตัวอย่าง : เวลาเรียกใช้งาน

```
$user = User::model()->findByPk(1); // เรียก user ที่มี id = 1
$profile = $user->profile;
echo $profile->website; // หรือ $user->profile->website; ก็ได้
```

### ความสัมพันธ์แบบ BELONGS\_TO

ในการนี้นี้ เราจะมองกลับกัน โดยที่ 1 post จะเป็นของ 1 user เท่านั้น ดังนั้น ความสัมพันธ์เมื่อมอง ส่วนทางกลับ เราจะใช้ BELONGS\_TO

```
public function relations() {
    return array(
        'user' => array(
            self::BELONGS_TO, 'User', 'author_id'
        ),
    );
}
```

เมื่อเราเขียนอยู่ในโมเดล ของ Post จะเป็นการอ้างว่า โมเดล Post นี้จะไปสัมพันธ์กับ โมเดล User แบบ BELONGS\_TO โดยจะใช้ author\_id (PK/FK) ของ Post เป็นตัวอ้างอิงไปยัง id (PK) ของ User

ตัวอย่าง : เวลาเรียกใช้งาน

```
$post = Post::model()->findByPk(10); // เรียก Post ที่มี id = 10  
$user = $post->user;  
echo $user->username; // หรือ $user->user->username; ก็ได้
```

ความสัมพันธ์แบบ HAS\_MANY

1 user จะสามารถโพสได้หลายโพส ดังนั้น ความสัมพันธ์จะเป็น 1 to many เราจะใช้ HAS\_MANY ในการเชื่อมความสัมพันธ์

```
public function relations() {  
    return array(  
        'posts' => array(  
            self::HAS_MANY, 'Post', 'author_id',  
            ),  
    );  
}
```

เมื่อเราเขียนอยู่ในโมเดล ของ User จะเป็นการอ้างว่า โมเดล User นี้จะไปสัมพันธ์กับ โมเดล Post แบบ HAS\_MANY โดยจะใช้ author\_id (FK) ของ Post เป็นตัวอ้างอิงมาหา id (PK) ของ User

ตัวอย่าง : เวลาเรียกใช้งาน

```
$user = User::model()->findByPk(1); // เรียก User ที่มี id = 1  
$posts = $user->posts;  
foreach($posts as $item){  
    echo $item->title;  
}
```

\*\*\* ข้อสังเกต \$posts ที่ได้นั้นจะมีอยู่หลายโพสดังนั้น จึงต้องใช้คำสั่ง foreach() เข้ามาช่วยในการดึงข้อมูลออกมาแสดง

ความสัมพันธ์แบบ MANY\_MANY

ในหัวข้อนี้ 1 post จะมีได้หลาย category และในแต่ละ 1 category อาจจะมีได้หลาย post ที่ได้ จึงเป็นความสัมพันธ์แบบ many to many เราจะใช้ MANY\_MANY ความสัมพันธ์ แบบ many to many นั้นจำเป็นต้องเพิ่มตาราง เพื่อใช้ในการเชื่อมโยง อ้างอิงของ 2 ตารางด้วย นั้นก็คือ ตาราง tbl\_post\_category นั้นเอง

```
public function relations() {
    return array(
        'categories' => array(
            self::MANY_MANY,'Category', 'tbl_post_category(post_id, category_id)'
        ),
    );
}
```

เมื่อเราเขียนอยู่ในโมเดล ของ Post จะเป็นการอ้างว่า โมเดล Post นี้จะไปสัมพันธ์กับ โมเดล Category แบบ MANY\_MANY โดยจะใช้ตาราง tbl\_post\_category เป็นตารางอ้างอิง มี post\_id (FK) อ้างอิง id (PK) ของโมเดล Post และ category\_id (FK) อ้างอิง id (PK) ของโมเดล Category

ตัวอย่าง : เวลาเรียกใช้งาน

```
$post = Post::model()->findPk(10); // โพส id ที่ 10
$categories = $post->categories;
foreach($categories as $item){
    echo $item->name;
}
```

การเขียน relation เชื่อม 2 โมเดลนั้น เราอาจเขียนแค่โมเดลเดียวที่ใช้ได้ครับ ไม่จำเป็นต้องเขียนความสัมพันธ์ทั้ง 2 โมเดล ถ้าเราไม่ได้เรียกใช้

**Active Record (AR)** : การดึงข้อมูลแบบธรรมชาติ

การดึงข้อมูลแบบนี้ เป็นแบบ พื้นฐาน Yii Framework กันเลยที่เดียว ก่อนจะเริ่ม เราต้องทำการสร้าง Model ไว้ซึ่งก่อนหลังจากเราสร้าง Model เดริยมพร้อมแล้ว เราสามารถกันเลยครับ

ตัวอย่าง : การดึงข้อมูลจากฐานข้อมูล

```
// Modelname ชื่อ Model ที่เราต้องการดึงข้อมูล
$dataModel = Modelname::model()->findAll(); // ดึงข้อมูลออกมากทั้งหมด
$dataModel = Modelname::model()->find(); // ดึงข้อมูล เฉพาะข้อมูล Column แรก ออกมา
$dataModel = Modelname::model()->findByPk($id); // ดึงข้อมูล เฉพาะข้อมูล $id นี้ ออกมา
$dataModel = Modelname::model()->findByAttributes($attributes); // ดึงข้อมูล $attributes นี้ ออกมา
```

ตัวอย่าง : การ Insert ข้อมูล

```
$add = new Model();
$add->..... = xxxxxxxx; // กำหนดข้อมูลที่ต้องการเพิ่มลงใน
// ตรง .... ใส่ชื่อ Attribute ของ Model จะสามารถเก็บค่าเข้า หรือดึงค่าออกได้ตามสะดวก
$add->save(); // บันทึกข้อมูลลงฐานข้อมูล
```

ตัวอย่าง : การ Update ข้อมูล

```
// Modelname คือ ชื่อ Model  
$update = Modelname::model()->findByPk($id); // $id ที่ต้องการแก้ไขข้อมูล  
$update->..... = xxxxxxx; // เวลา find ให้ใช้ตัวที่ไม่ใช่ findAll เพราะมัน update ได้ทีละ record  
// ตรง ..... ใส่ชื่อ Attribute ของ Model จะสามารถเก็บค่าเข้า หรือดึงค่าออกได้ตามสะดวก  
$update->save(); // บันทึกข้อมูลลงฐานข้อมูล
```

ตัวอย่าง : การ Delete ข้อมูล

```
// Modelname คือ ชื่อ Model  
$data = Modelname::model()->findByPk($id); // $id ที่ต้องการลบข้อมูล  
$data->delete(); // เวลา find ให้ใช้ตัวที่ไม่ใช่ findAll เพราะมัน Delete ได้ทีละ record
```

**Array Data (AD) : CArrayDataProvider**

การดึงข้อมูลจาก Array ผ่านเมธอด CArrayDataProvider ให้เราเข้าไปสร้าง Controller ชื่อว่า “CArrayDataProvider” เพื่อใช้ในการทดสอบ

ตัวอย่าง : การดึงข้อมูลจาก Array ผ่านเมธอด CArrayDataProvider ออกมาแสดงใน GridView

```
$rawData = array( // ข้อมูล Array  
    array('id'=>1,'username'=>'from','email'=>'array'),  
    array('id'=>2,'username'=>'test 2','email'=>'hello@example.com'),  
);  
// สร้าง Data Provider  
$dataProvider = new CArrayDataProvider($rawData, array(  
    'pagination'=>array(  
        'pageSize'=>10,  
    ),  
));  
// แสดงผล GridView  
$this->widget('zii.widgets.grid.CGridView', array(  
    'dataProvider'=>$dataProvider,  
));
```

แสดงผล : ให้เข้าไปที่ URL [http://localhost/tutorial\\_yii/CArrayDataProvider](http://localhost/tutorial_yii/CArrayDataProvider)

Displaying 1-2 of 1 result.

ID	Username	Email
1	from	array
2	test 2	hello@example.com

## การพัฒนาเว็บแอปพลิเคชัน ด้วย Yii Framework

### บทที่ 6 การใช้งาน Active Form

#### Form และ Widget CActiveForm (รูปแบบมาตรฐานของ Yii Framework ที่ใช้กัน)

รูปแบบ Form และ Widget CActiveForm จะทำงานร่วมกับ Model (โมเดล) ที่เราสร้างไว้ และจะทำงานระหว่างคำสั่ง \$this->beginWidget('CActiveForm') กับ \$this->endWidget() ซึ่งหากเราต้องการสร้างฟอร์มขึ้นมา เราสามารถเขียนคำสั่งการเรียกใช้ ฟิลด์ ในรูปแบบต่าง ๆ ได้ เช่น textField, textArea, passwordField เป็นต้น โดยให้เราสร้าง Controller ซึ่งว่า CActiveForm ก่อนและสร้างไฟล์ Model จากตาราง active\_form (<http://goo.gl/Zzo0zm>) ด้วยนะครับ

ตัวอย่าง : รูปแบบคำสั่ง Form และ CActiveForm || ดาวโหลดไฟล์ (.SQL) <http://goo.gl/Zzo0zm>

```
<?php $model = new ActiveForm(); ?>
<?php $form = $this->beginWidget('CActiveForm', array(
    'id'=>'active-form', // ชื่อฟอร์ม
    'enableAjaxValidation'=>true, // เปิดการทำงาน ส่งข้อมูลแบบ Ajax
    'enableClientValidation'=>true,
)); ?>
<!-- แทรกคำสั่งการเรียกใช้งาน ฟิลด์ ในรูปแบบต่าง ๆ (ระหว่างส่วนนี้เท่านั้น) -->
<?php $this->endWidget(); ?>
```

คำสั่ง : errorSummary

```
// รูปแบบ errorSummary($model);
<?php echo $form->errorSummary($model); ?>
```

ใช้แสดง “ข้อผิดพลาด” (Error) ของโปรแกรมหรือการป้อนข้อมูลลงในฟิลด์ ไม่ถูกต้อง



คำสั่ง : error

```
// รูปแบบ error($model,$AttributeName,$htmlOptions);  
<?php echo $form->error($model,'textField'); ?>
```

Text Field \*

Text Field ไม่ควรเป็นค่าว่าง

คำสั่ง : labelEx

```
// รูปแบบ labelEx($model,$AttributeName,$htmlOptions);  
<?php echo $form->labelEx($model,'textField'); ?>
```

Text Field \*

```
// รูปแบบ textField($model,$AttributeName,$htmlOptions);  
<?php echo $form->labelEx($model,'textField'); ?>  
<?php echo $form->textField($model,'textField'); ?>  
<?php echo $form->error($model,'textField'); ?>
```

Text Field \*

คำสั่ง : textArea

```
// รูปแบบ textArea($model,$AttributeName,$htmlOptions);  
<?php echo $form->labelEx($model,'textarea'); ?>  
<?php echo $form->textArea($model,'textarea'); ?>  
<?php echo $form->error($model,'textarea'); ?>
```

Textarea \*

คำสั่ง : passwordField

```
// รูปแบบ passwordField($model,$AttributeName,$htmlOptions);
<?php echo $form->labelEx($model,'passwordField'); ?>
<?php echo $form->passwordField($model,'passwordField'); ?>
<?php echo $form->error($model,'passwordField'); ?>
```

คำสั่ง : numberField

```
// รูปแบบ numberField($model,$AttributeName,$htmlOptions);
<?php echo $form->labelEx($model,'numberField'); ?>
<?php echo $form->numberField($model,'numberField'); ?>
<?php echo $form->error($model,'numberField'); ?>
```

คำสั่ง : telField

```
// รูปแบบ telField($model,$AttributeName,$htmlOptions);
<?php echo $form->labelEx($model,'telField'); ?>
<?php echo $form->telField($model,'telField'); ?>
<?php echo $form->error($model,'telField'); ?>
```

คำสั่ง : fileField

```
// รูปแบบ fileField($model,$AttributeName,$htmlOptions);
<?php echo $form->labelEx($model,'fileField'); ?>
<?php echo $form->fileField($model,'fileField'); ?>
<?php echo $form->error($model,'fileField'); ?>
```

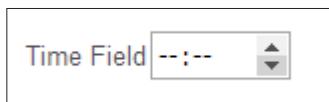
คำสั่ง : dateField

```
// รูปแบบ dateField($model,$AttributeName,$htmlOptions);
<?php echo $form->labelEx($model,'dateField'); ?>
<?php echo $form->dateField($model,'dateField'); ?>
<?php echo $form->error($model,'dateField'); ?>
```



คำสั่ง : timeField

```
// รูปแบบ timeField($model,$AttributeName,$htmlOptions);
<?php echo $form->labelEx($model,'timeField'); ?>
<?php echo $form->timeField($model,'timepicker'); ?>
<?php echo $form->error($model,'timepicker'); ?>
```



คำสั่ง : dropDownList

```
// รูปแบบ dropDownList($model,$AttributeName,$value,$htmlOptions);
<?php echo $form->labelEx($model,'dropDownList'); ?>
<?php echo $form->dropDownList($model,'dropdown',array('1'=>1,'2'=>2)); ?>
<?php echo $form->error($model,'dropdown'); ?>
```



คำสั่ง : emailField

```
// รูปแบบ emailField($model,$AttributeName,$htmlOptions);
<?php echo $form->labelEx($model,'emailField'); ?>
<?php echo $form->emailField($model,'emailField'); ?>
<?php echo $form->error($model,'emailField'); ?>
```



คำสั่ง : urlField

```
// รูปแบบ urlField($model,$AttributeName,$htmlOptions);
<?php echo $form->labelEx($model,'urlField'); ?>
<?php echo $form->urlField($model,'urlField'); ?>
<?php echo $form->error($model,'urlField'); ?>
```

คำสั่ง : rangeField

```
// รูปแบบ rangeField($model,$AttributeName,$htmlOptions);
<?php echo $form->labelEx($model,'rangeField'); ?>
<?php echo $form->rangeField($model,'rangeField'); ?>
<?php echo $form->error($model,'rangeField'); ?>
```

คำสั่ง : hiddenField

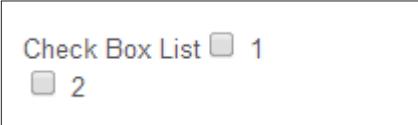
```
// รูปแบบ hiddenField($model,$AttributeName,$htmlOptions);
<?php echo $form->labelEx($model,'textField'); ?>
<?php echo $form->hiddenField($model, 'textField'); ?>
<?php echo $form->error($model, 'textField'); ?>
```

คำสั่ง : checkBox

```
// รูปแบบ checkBox($model,$AttributeName,$htmlOptions);
<?php echo $form->labelEx($model,'checkbox'); ?>
<?php echo $form->checkBox($model,'checkboxes'); ?>
<?php echo $form->error($model,'checkboxes'); ?>
```

คำสั่ง : checkBoxList

```
// รูปแบบ checkBoxList($model,$AttributeName,$htmlOptions);
<?php echo $form->labelEx($model,'checkBoxList'); ?>
<?php echo $form->checkBoxList($model,'checkboxes',array(1,2)); ?>
<?php echo $form->error($model,'checkboxes'); ?>
```



Check Box List  1  
 2

คำสั่ง : radioButton

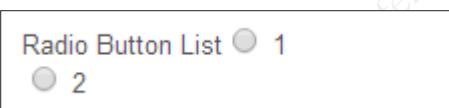
```
// รูปแบบ radioButton($model,$AttributeName,$htmlOptions);
<?php echo $form->labelEx($model,'radioButton'); ?>
<?php echo $form->radioButton($model,'radioButton'); ?>
<?php echo $form->error($model,'radioButton'); ?>
```



Radio Button \*

คำสั่ง : radioButtonList

```
// รูปแบบ radioButtonList($model,$AttributeName,$htmlOptions);
<?php echo $form->labelEx($model,'radioButtonList'); ?>
<?php echo $form->radioButtonList($model,'radioButtons',array(1,2)); ?>
<?php echo $form->error($model,'radioButtons'); ?>
```



Radio Button List  1  
 2

คำสั่ง : submitButton

```
// รูปแบบ submitButton($label,$htmlOptions);
<?php echo CHtml::submitButton('submit'); ?>
```



submit

คำสั่ง : listBox

```
// รูปแบบ listBox($model,$AttributeName,$value,$htmlOptions);
<?php echo $form->labelEx($model,'radioButtonList'); ?>
<?php echo $form->listBox($model,'radioButtons',array(1,2)); ?>
<?php echo $form->error($model,'radioButtons'); ?>
```



คำสั่ง : resetButton

```
// รูปแบบ resetButton($label,$htmlOptions);
<?php echo CHtml::resetButton('reset'); ?>
```



คำสั่ง : ajaxButton

```
// รูปแบบ ajaxButton($name,$url,$ajaxOptions,$htmlOptions);
<?php echo CHtml::ajaxButton('ajaxButton',array('myController/MyAction'),
array(
    'dataType'=>'json',
    'type'=>'post',
    'beforeSend'=>'function(){alert(\'beforeSend\');}',
    'success'=>'function(data) {alert(\\'success\');}',
    'error'=>'js:function(){alert(\\'error\');}',
    'complete'=>'js:function(){alert(\\'complete\');}',
),
array('id'=>'mybtn1','class'=>'class1 class2')
); ?>
```



คำสั่ง : ajaxSubmitButton

```
// รูปแบบ ajaxSubmitButton($name,$url,$ajaxOptions,$htmlOptions);
<?php echo CHtml::ajaxSubmitButton('ajaxSubmitButton',array('myController/MyAction'),
array(
'dataType'=>'json','type'=>'post',
'beforeSend'=>'function(){alert(\\'beforeSend\\');}',
'success'=>'function(data) {alert(\\'success\\');}',
'error'=>'js:function(){alert(\\'error\\');}',
'complete'=>'js:function(){alert(\\'complete\\');}',
),
array('id'=>'mybtn1','class'=>'class1 class2'
); ?>
```



## สรุปการสร้างฟอร์มแบบ CActiveForm

ฟอร์มส่วนใหญ่แล้ว เราจะเป็นการสร้างฟอร์ม โดยเรียกใช้งาน Widget CActiveForm ซะกว่าเนื่องจากการตอบสนองที่ดีกว่าและการใช้งานง่ายกว่าแบบ ActiveForm ธรรมดามาก และหากเราติดตั้ง Yii Booster Extension ด้วยแล้ว เราอีกจะสามารถเรียกใช้งาน ActiveForm ที่หลากหลายมากยิ่งขึ้น



## Introducing YiiBooster

By Yii developers, for Yii developers: YiiBooster is Yii Bootstrap on 'roids.

## การพัฒนาเว็บแอปพลิเคชัน ด้วย Yii Framework

### บทที่ 7 ติดตั้งและใช้งาน Extension

Extensions (ส่วนขยาย) คือ ซอฟแวร์, ระบบ หรือ application ที่ได้ถูกพัฒนาจากบุคคลภายนอกที่ไม่ใช่ผู้พัฒนา เช่นระบบ ล็อกอิน ระบบสมัครสมาชิก ระบบบันลือก ระบบเว็บนอร์ด ระบบตัวกรองคำ เป็นต้น ซึ่งระบบเหล่านี้ เราสามารถดาวน์โหลดและทำการติดตั้งเพิ่มเติมลงในตัวระบบของเราที่กำลังพัฒนาอยู่ได้ โดยที่เราไม่ต้องไปเสียเวลาพัฒนาระบบดังกล่าวเหล่านี้เอง

ประโยชน์ของ Extensions นักจากทำให้การพัฒนาระบบของเรา นั้น เสริจเร็วขึ้นแล้ว extension นั้น มักจะมีประสิทธิภาพมากกว่าที่เราเขียนเอง เพราะ extension แต่ละตัว ผู้พัฒนามักจะพัฒนา กันมาอย่างต่อเนื่อง และเมื่อ extension นั้นมีอีก version ใหม่ เรา ก็สามารถ อัพเดท extension ใหม่ ตัวนั้นทันตัวก่อได้

สำหรับ Yii Framework และ Extension จะมีความสำคัญเป็นอย่างมาก เพราะถือว่า การนำ extension มาติดตั้งและใช้งานร่วมกับระบบของเรา จะช่วยให้ระบบมีประสิทธิภาพมากยิ่งขึ้น พร้อมทั้งทำให้การพัฒนาระบบของเราเสร็จเร็วกว่าปกติ แต่ใน Extensions ส่วนใหญ่แล้ว จะเป็นการรวมเอา Widget ในรูปแบบต่าง ๆ รวมกันไว้เป็นหนึ่ง หน้างานนี้ Extension ทำให้สะดวกในการเรียกใช้งานมากขึ้น

Widget คือ คำที่อ่านว่า “วิดเจ็ต” ครับ สำหรับ Yii “วิดเจ็ต” เป็นส่วนขยาย (Extension) แบบหนึ่ง ข้อดีของ “วิดเจ็ต” คือ สามารถใช้ช้าได้หมายความว่า กับระบบเว็บแอปพลิเคชันขนาดใหญ่ ๆ ที่มีองค์ประกอบอนามากมาย “วิดเจ็ต” เป็นอีกส่วนหนึ่งที่ผมชอบมากใน Yii เพราะมีการออกแบบจัดการได้ดีมาก โดยสามารถพนวกทรัพยากรพากศรีปและรูปภาพเอาไว้ใน “วิดเจ็ต” ได้เลย

ในบทนี้ เราจะมาเรียนรู้วิธีการดาวน์โหลด ติดตั้ง ตั้งค่าระบบ และการใช้งาน Extension พร้อมกับแนะนำ Extension ดี ๆ ไว้ให้เราใช้งานให้กับเว็บแอปพลิเคชันของเรา

#### การดาวน์โหลด Extension

การดาวน์โหลด Extensions เพื่อนำมาใช้งานร่วมกับระบบของเรา นั้น เราสามารถดาวน์โหลดได้ตามเว็บไซต์สอน Yii Framework ทั่วไปได้ หรือ ดาวน์โหลดจากแหล่งรวม Extensions สำหรับ Yii Framework ได้ที่ลิงก์ <http://www.yiiframework.com/extensions/>

#### การติดตั้ง Extension

การติดตั้ง Extension นั้น เราสามารถทำได้่ายมาก ๆ แค่เรา นำไฟล์เดอร์ ที่ได้จากการแตกไฟล์ หลังการดาวน์โหลด Extension ไปเก็บไว้ในไฟล์เดอร์ protected/extensions ที่ Yii Framework ได้เตรียมไว้เพียงแค่นี้ เรา ก็สามารถติดตั้ง Extension สำเร็จเรียบร้อยแล้ว แต่ในบาง Extension ต้องมีการตั้งค่าระบบ Extensions เพื่อนำไปใช้งาน ในไฟล์ Config (main.php)

## การตั้งค่าระบบ (หลังติดตั้ง Extensions)

ในบาง Extensions นั้น เราต้องมีการตั้งค่าระบบก่อนการเรียกใช้งาน Extensions ส่วนใหญ่แล้ว การตั้งค่าระบบจะสามารถเข้าไปตั้งค่าระบบได้จากไฟล์ Config (main.php) ในส่วนของ preload, import, modules, components

### ตัวอย่าง : ไฟล์ Config (main.php)

```
return array(
    'preload' => array('log'),
    'import'=>array(
        'application.models.*',
        'application.components.*'
    ),
    'components'=>array(
        'user'=>array( /*.... */ ),
        'urlManager'=>array( /*.... */ ),
        'db'=>array( /*.... */ ),
        'errorHandler'=>array( /*.... */ ),
        'log'=>array( /*.... */ ),
    ),
)
```

## การเรียกใช้งาน Extension

การเรียกใช้งาน Extension ส่วนใหญ่จะเป็น Widget และการเรียกใช้งาน Widget นั้น เราสามารถเขียนคำสั่งดังนี้ \$this->widget(['Widget Name', array([Widget Options ]));

### ตัวอย่าง : การเรียกใช้งาน Extension Bootstrap (Widget TbGridView)

```
$this->widget('bootstrap.widgets.TbGridView', array(
    'id' => 'grid_1', // ชื่อตาราง
    'type'=>'striped bordered',
    'dataProvider' => $dataProvider, // ตัวแปร data ที่มีข้อมูล
    'enablePagination' => true, // กำหนดให้แสดงปุ่มเปลี่ยนหน้า,
    'columns' => array( // กำหนด column ที่จะแสดง
        'id', 'pid', // (Field) 'id' และ 'pid' ของตาราง Account
        'username', 'password', // (Field) 'username' และ 'password' ของตาราง Account
    ),
));
```

## แนะนำ Extension ที่จำเป็นและน่าสนใจ

Extension ที่จะมาแนะนำให้ใช้กันมีทั้งหมด 3 Extension ด้วยกัน จะประกอบไปด้วย Yii Booster v.4.0.1, Yii Rights Modules, Yii User Moduls ที่จำเป็นและน่าสนใจที่สุด

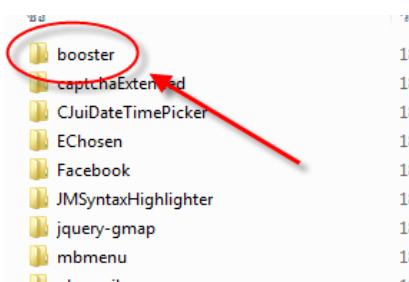
### Yii Booster version 4.0.1

Extension เกี่ยวกับการแสดงผล UI ให้กับผู้ใช้งาน ด้วยการรวมฟังก์ชัน รูปแบบการแสดงของ Twitter Bootstrap v.3.1.1 อาทิเช่น การวางแผนสร้างเว็บให้รองรับการแสดงผลหลากหลายอุปกรณ์ มือถือ คอมพิวเตอร์ เป็นต้น



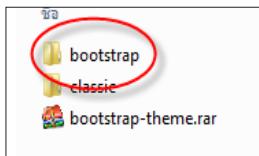
ดาวน์โหลด Yii Booster v.4.0.1 || ดาวน์โหลดที่นี่ <http://goo.gl/wL098g>

แตกไฟล์ yiibooster-4.0.1.zip และแก้ไขชื่อไฟล์เดอร์เป็น “booster” ไว้ในไฟล์เดอร์ extensions ตรงไดเรกทอรี่ [skeleton]/protected/extensions/



ดาวโหลด Twitter Bootstrap Theme v.3.1.1 || ดาวโหลดที่นี่ <http://goo.gl/w6e6Xe>

แตกไฟล์ bootstrap-theme.rar และเปลี่ยนชื่อไฟล์เดอร์เป็น “bootstrap” ไว้ในไฟล์เดอร์ themes ตรงไดเรกทอร์ [skeleton]/themes/



ตัวอย่าง : Theme Bootstrap v.3

Congratulations! You have successfully created your Yii application.  
You may change the content of this page by modifying the following two files:  
• View file: E:\xampp\htdocs\PhpProject1\themes\bootstrap\views\site\index.php  
• Layout file: E:\xampp\htdocs\PhpProject1\themes\bootstrap\views\layouts\main.php  
For more details on how to further develop this application, please read the documentation. Feel free to ask in the [forum](#), should you have any questions.  
Copyright © 2014 by DriveSoftCenter.Net  
All Rights Reserved.  
Powered by [Yii Framework](#).

## ตั้งค่าการเข้มต่อ Yii Booster v.4.0.1

การตั้งค่า เพื่อเรียกใช้งาน Yii Booster v.4.0.1 สามารถทำได้ด้วย การแก้ไขไฟล์ Config (main.php) ทั้งหมด 4 ส่วนคือ ส่วนของ theme, preload, aliases, components

ตัวอย่าง : ตั้งค่าการเข้มต่อ Yii Booster v.4.0.1

คำสั่ง : PHP

ไดเรกทอร์ : [skeleton]/protected/config/main.php

```
return array(  
    'theme'=>'bootstrap', // เพิ่มเติม Theme Bootstrap  
    'preload' => array('log', 'booster'), // เพิ่มเติมคำสั่ง 'booster' นี้ลงไป  
    'aliases'=>array('booster'=>'application.extensions.booster'), // เพิ่มเติมคำสั่ง  
    'components'=>array(  
        'booster'=>array('class'=>'booster.components.Booster'), // เพิ่มเติมคำสั่ง  
    ),  
)
```

## คำแนะนำ

Widget ประเภทต่าง ๆ ภายใน Yii Booster Extension version 4.0.1 มีมากกว่า 20 ประเภทที่เตรียมไว้ให้เราเรียกใช้งาน แต่พมขอกตัวอ่อนมาเพียงแค่ส่วนหนึ่งของ Widget ทั้งหมด

### ตัวอย่าง : การใช้งาน Widget TbLabel

```
$this->widget('booster\widgets\TbLabel',array(
    'context' => 'success',
    // 'default', 'primary', 'success', 'info', 'warning', 'danger'
    'label' => 'Success',
));
```

Example

Success

### ตัวอย่าง : การใช้งาน Widget TbHighCharts

```
$this->widget('booster\widgets\TbHighCharts',array(
    'options'=>array(
        'series'=>array(['data'=>[1, 2, 3, 4, 5, 1, 2, 1, 4, 3, 1, 5]])
    )
));
```

Example

Chart title



Highcharts.com

ติดตาม ภาคผนวก คู่มือการใช้งาน Yii Booster v.4.0.1 (ท้ายเล่ม)

Chapter  
7

## Yii User Modules v.0.3.61

ผมขอแนะนำเป็นพิเศษเลย เพราะ Yii User Modules ตัวนี้ถือว่า เป็นตัวที่ยอดนิยมที่สุด ความสามารถเพื่อใช้จัดการกับ User เช่น การล็อกอิน, การสมัครสมาชิก, การเปลี่ยนรหัสผ่าน, การจัดการข้อมูลสมาชิกฯลฯ

Yii User Modules ได้จัดเตรียมฟังก์ชันและระบบที่จำเป็นต้องมีในเว็บแอปพลิเคชันในปัจจุบันไว้ให้เราได้เรียกใช้ได้ตามสะดวก เมื่อติดตั้ง Yii User Modules แล้ว เว็บแอปพลิเคชันของเราจะมีประสิทธิภาพในการทำงานได้ดีขึ้นมาก ไหน เรามาติดตามกันต่อ กันเลย

ดาวโหลด Yii User Modules v.0.3.61 || ดาวโหลดได้ที่นี่ <http://goo.gl/kTgUpD>

หลังการดาวโหลดให้แตกไฟล์ yii-user-0.3-61-gfc69518.zip และเปลี่ยนชื่อไฟล์เดอร์เป็น “user” เก็บไว้ในไฟล์เดอร์ modules ตรงไดเรกทอรี่ [skeleton]/protected/modules/

ตัวอย่าง : ตั้งค่า Yii User Modules ในไฟล์ Config (main.php)

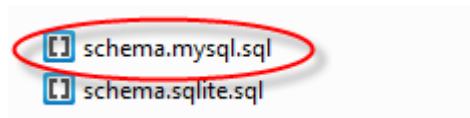
คำสั่ง : PHP

ไดเรกทอรี่ : [skeleton]/protected/config/main.php

```
return array(
    'import'=>array(
        .....
        'application.modules.user.models.*', // เพิ่มเติม
        'application.modules.user.components.*', // เพิ่มเติม
    ),
    'modules' =>array(
        // เพิ่มเติม
        'user'=>array(
            'tableUsers'=>'tbl_users',
            'tableProfiles'=>'tbl_profiles',
            'tableProfileFields'=>'tbl_profiles_fields',
        ),
    ),
    'components' => array(
        'user' => array(
            'allowAutoLogin'=>true,'class'=>'CWebUser', 'loginUrl'=>array('/user/login'), // เพิ่มเติม
        ),
    ),
)
)
```

## นำเข้า Database (.SQL)

Import ไฟล์ฐานข้อมูล จากไฟล์เครื่อง data (schema.mysql.sql) ที่อยู่ในไฟล์เครื่อง [skeleton]/protected/modules/user/data/schema.mysql.sql ลงในฐานข้อมูลที่เราที่ได้สร้างไว้



ทดสอบ : ระบบ Yii User Modules v.0.3.61 | ลิ้งค์ [http://localhost/tutorial\\_yii/user](http://localhost/tutorial_yii/user)

ERROR : หากเจอ Error เหนืออนในรูปนี้

### CDbException

ข้อมูลที่ใช้งาน "User" ไม่สามารถเลือกฟิลด์ "user.lastvisit\_at" ได้ ด้วย因为, ชื่อฟิลด์จะต้องมีอยู่ในตารางหรือระบุเป็นชื่อเสื่อมของฟิลด์เดิม

D:\xampp\htdocs\yii\framework\db\ar\ CActiveFinder.php(960)

```
948 }  
949 elseif(preg_match('/^(.*)\s+AS\s+(\w+)/im', $name, $matches)) // if the column is already aliased  
950 {
```

วิธีแก้ไข : ให้เราเข้าไปยังฐานข้อมูล ตาราง “users” และเพิ่งแก้ไขชื่อ ฟิลด์ จาก “lastvisit” มาเป็น “lastvisit\_at” แทนนะครับ

ตัวอย่าง : ระบบ Yii User Modules v.0.3.61 สำเร็จ

## List User

Displaying 1-2 of 1 result.

username	Registration date	Last visit
admin	2014-07-27 14:26:51	0000-00-00 00:00:00
demo	2014-07-27 14:26:51	0000-00-00 00:00:00

## การใช้งาน Yii User Extension

ระบบล็อกอิน : Yii User ทดลอง Login ด้วยกรอก (username :: admin) และ (password :: admin) ซึ่ง Yii User ได้เตรียม Username, Password ไว้ให้เราทดสอบแล้ว

ระบบล็อกอิน : เข้าไปที่ลิ้งค์ [http://localhost/tutorial\\_yii/user/login](http://localhost/tutorial_yii/user/login)

# Login

Please fill out the following form with your login credentials:

*Fields with \* are required.*

username or email \*

password \*

[Register](#) | [Lost Password?](#)

Remember me next time

แสดงข้อมูลส่วนตัว : เข้าไปที่ลิ้งค์ [http://localhost/tutorial\\_yii/user/profile](http://localhost/tutorial_yii/user/profile)

## Your profile

username	admin	Operations
First Name	Administrator	<a href="#">Manage Users</a>
Last Name	Admin	<a href="#">List User</a>
E-mail	webmaster@example.com	<a href="#">Edit</a>
Registration date	2014-07-27 14:26:51	<a href="#">Change password</a>
Last visit	2014-07-27 14:50:31	<a href="#">Logout</a>
Status	Active	

\*\*\* เราสามารถจัดการข้อมูล Profile ส่วนตัวของเราได้ เช่น แก้ไขข้อมูล, เปลี่ยนรหัสผ่าน เป็นต้น

ระบบจัดการข้อมูลส่วนตัว : เข้าไปที่ลิงก์ [http://localhost/tutorial\\_yii/user/admin](http://localhost/tutorial_yii/user/admin)

## Manage Users

You may optionally enter a comparison operator (<, <=, >, >=, <> or =) at the beginning of each of your search values to specify how the comparison should be done.

[Advanced Search](#)

Displaying 1-2 of 1 result.

ID	username	E-mail	Registration date	Last visit	Superuser	Status	Operations
1	admin	webmaster@example.com	2014-07-27 14:26:51	2014-07-27 14:50:31	Yes	Active	
2	demo	demo@example.com	2014-07-27 14:26:51	0000-00-00 00:00:00	No	Active	

## Update User 2

Fields with \* are required.

username \*

password

E-mail \*

Superuser \*

Status \*

## Operations

- [Create User](#)
- [View User](#)
- [Manage Users](#)
- [Manage Profile Field](#)
- [List User](#)

\*\*\* เราสามารถ “สร้าง”, “ค้นหา”, “ลบ”, “แก้ไข”, “รายละเอียด”, “รายการทั้งหมด” ของข้อมูล Users ได้ตามรายการที่เราเลือก

แสดงข้อมูลส่วนตัว “ADMIN” : เข้าไปที่ลิงก์ [http://localhost/tutorial\\_yii/user/admin/view/id/2](http://localhost/tutorial_yii/user/admin/view/id/2)

## View User "demo"

Id	2
username	demo
First Name	Demo
Last Name	Demo
password	fe01ce2a7fbac8fafaed7c982a04e229
E-mail	demo@example.com
activation key	099f825543f7850cc038b90aaff39fac
Registration date	2014-07-27 14:26:51
Last visit	0000-00-00 00:00:00
Superuser	No
Status	Active

## Operations

- [Create User](#)
- [Update User](#)
- [Delete User](#)
- [Manage Users](#)
- [Manage Profile Field](#)
- [List User](#)

ระบบจัดการข้อมูล ProfileField : เข้าไปที่ลิงค์ [http://localhost/tutorial\\_yii/user/profileField/admin](http://localhost/tutorial_yii/user/profileField/admin)

The screenshot shows a table titled "Manage Profile Fields" with the following data:

ID	Variable name	Title	Field Type	Field Size	Required	Position	Visible	Operations
2	firstname	First Name	VARCHAR	50	Yes and show on registration form	0	For all	
1	lastname	Last Name	VARCHAR	50	Yes and show on registration form	1	For all	

เราสามารถ “สร้าง”, “คืนหา”, “ลบ”, “แก้ไข”, “รายละเอียด”, “รายการทั้งหมด” ของข้อมูล Users ได้ตามรายการที่เราเลือก

Profile Field คือ Field ที่เราต้องการให้สมาชิก กรอก เพิ่มเติม ซึ่ง เมื่อong ต้น Yii-User ได้กำหนด Field “firstname” และ “lastname” ไว้ให้เราแล้ว แต่หากเราต้องการเพิ่ม Field “facebook” หรือ Filed “twitter” เรา ก็สามารถทำได้

The screenshot shows the "Create Profile Field" form with the following fields:

- Variable name \***:
- Title \***:
- Field Type \***:
- Field Size**:
- Field Size min**:
- Range**:
- Predefined values (example: 1;2;3;4;5 or 1==One;2==Two;3==Three;4==Four;5==Five)**:
- Error Message**:

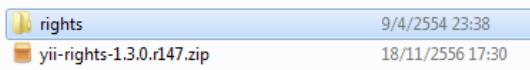
## Yii Rights Modules v.1.3.0

ผลงานนำ Modules ตัวที่ 2 เป็นพิเศษ Yii Rights Modules ความสามารถของมันจะเพื่อช่วยในการขัดการกับ Access Control ต่าง ๆ ให้เป็นอย่างดี

เนื่องจาก Yii Rights Modules มีหน้าที่ไว้จัดการกับ Access Control ต่าง ๆ เช่น การกำหนดคุณสมบัติการเข้าถึง Controller ของ User แต่ละระดับ (ผู้ใช้ทั่วไป, สมาชิก, ผู้ดูแลระบบ) เป็นต้น

ดาวโหลด **Yii Rights Modules v.1.3.0** || ได้ที่นี่ <http://goo.gl/CZXOnw>

แตกไฟล์ yii-rights-1.3.0.r147.zip และเปลี่ยนชื่อโฟลเดอร์เป็น “rights” ไปไว้ในโฟลเดอร์ modules ไดเรกทอรี่ [skeleton]/protected/modules



ตัวอย่าง : ตั้งค่า Yii Rights Modules ในไฟล์ Config (main.php)

คำสั่ง : PHP

ไดเรกทอรี่ : [skeleton]/protected/config/main.php

```
return array(
    'import'=>array(
        ....
        'application.modules.rights.*', // เพิ่มเติม
        'application.modules.rights.components.*', // เพิ่มเติม
    ),
    'modules' => array(
        // เพิ่มเติม
        'rights' => array(
            'superuserName'=>'admin',
            'appLayout'=>'//layouts/main', // เปลี่ยน Theme Bootstrap
            'layout'=>'rights.views.layouts.main',
            'install'=>false,
        ),
    ),
    'components' => array(
        'user' => array(
            'allowAutoLogin'=>true,'class'=>'RWebUser', 'loginUrl'=>array('/user/login'), // เพิ่มเติม
        ),
        'authManager' => array(
            'class' => 'RDbAuthManager', 'connectionID'=>'db', 'defaultRoles'=>array('Guest'), // เพิ่มเติม
        ),
    ),
)
```

นำเข้า Database (.SQL) || ดาวน์โหลดที่นี่ <http://goo.gl/Fg61Pe>

Import ไฟล์ฐานข้อมูล จากไฟล์ที่เราดาวน์โหลดมา (schema-yii-rights-edit.sql) ลงในฐานข้อมูลที่เราตั้งไว้

Name	Date modified
schema-yii-rights-edit.sql	27/7/2557 15:17

ทดสอบ : การ Login (เข้าสู่ระบบ) || เข้าไปที่ลิงค์ [http://localhost/tutorial\\_yii/user/login](http://localhost/tutorial_yii/user/login)

**Login**

Please fill out the following form with your login credentials:

Fields with \* are required.

Username *	Username : admin
<input type="text" value="admin"/>	Password : admin
Password *	<input type="password" value="*****"/>
Hint: You may login with demo/demo or admin/admin.	
<input type="checkbox"/> Remember me next time	
<input type="button" value="Login"/>	

หน้าแรก Yii Rights : เข้าไปที่ลิงค์ [http://localhost/tutorial\\_yii/rights](http://localhost/tutorial_yii/rights)

Assignments	Permissions	Roles	Tasks	Operations
<b>Assignments</b>				
Here you can view which permissions has been assigned to each user.				
Name	Roles	Tasks	Operations	
admin	Super Admin			
demo	Member			

แก้ไขไฟล์ : Components (Controller.php)

คำสั่ง : PHP

ไดเรกทอรี่ : [skeleton]/protected/components/Controller.php

// แบบใหม่ (เปลี่ยน extends จาก CController มาเป็น RController)

class Controller extends RController {...}

หน้าแรก : Yii Rights Modules || เข้าไปที่ลิ้งค์ [http://localhost/tutorial\\_yii/rights](http://localhost/tutorial_yii/rights)

Name	Roles	Tasks	Operations
admin	Super Admin		
demo	Member		

## เมนูหลักของ Yii Rights คือ

1. Assignments
2. Permissions
3. Roles
4. Tasks
5. Operations



Assignments หรือ การมอบหมายงาน จะเป็นการกำหนดความสามารถแต่ละ user ว่า user อยู่ใน Role ไหน สามารถ access เข้าระบบอันไหนได้บ้าง โดยการกำหนดใน Assignments จะมีผลแต่ละ user ท่านนั้น user นั่ง จะไม่ส่งผล ต่ออีก 1 user

Permissions หรือ กำหนดศิทธิ์ จะเป็นการกำหนดศิทธิ์ ในแต่ละ Role ให้สามารถเข้าไปใช้งาน Controller หรือ Action ต่าง ๆ

Roles หรือ บทบาท จะเป็นการกำหนดบทบาทหน้าที่ ว่าระบบเรา มี Role อะไรบ้าง โดยแรกเริ่มพื้นฐาน Right จะสร้างไว้อยู่ 3 อัน คือ Admin, Authenticated, Guest โดย account แรกสุดของเราที่สร้างตอน install right จะอยู่ใน role ของ Admin โดยอัตโนมัติ และไม่สามารถลบ role Admin ออกได้

## วิธีการใช้งาน : Yii Rights Modules ร่วมกับ Yii User Modules

ก่อนอื่นให้เราสร้าง Account User ขึ้นมา 1 User โดยใช้ Yii User Modulse สร้างและกำหนดให้ค่าดังนี้ Name : Test Username : test, Password : text ลิ้งค์ [http://localhost/tutorial\\_yii/user/admin/create](http://localhost/tutorial_yii/user/admin/create) เมื่อสร้าง User เสร็จแล้ว เราจะมาเรียนรู้วิธีการกำหนด Access Control ของ Yii Rights Modules กันเลย

ตัวอย่าง : เข้าไปที่ “หน้าแรก” ของ Yii Rights Modules || ลิ้งค์ที่นี่ [http://localhost/tutorial\\_yii/rights](http://localhost/tutorial_yii/rights)

Name	Roles	Tasks	Operations
admin	Super Admin		
demo	Member		
test			

\*\*\* เราจะพบกับ Account ใหม่ที่เราสร้างชื่อว่า “test” ดังรูปด้านบน

ตัวอย่าง : คลิกที่ลิ้งค์คำว่า “test” เพื่อกำหนด Role

Name	Roles	Tasks	Operations
admin	Super Admin		
demo	Member		
test			

ตัวอย่าง : กำหนด User “test” ให้อยู่ใน Role ชื่อ Member และ กดปุ่ม Assign

Assignments for test

This user has not been assigned any items.

Assign item

Member ▾

- Roles
- Super Admin
- Member**

ตัวอย่าง : ผลลัพธ์การกำหนด Role ให้กับ User “test”

Assignments for test

Member	Role	Revoke
--------	------	--------

Assign item

Super Admin ▾

Assign

ตัวอย่าง : กลับไปหน้าแรกของ Yii Rights Modules || ลิ้งค์ที่นี่ [http://localhost/tutorial\\_yii/rights](http://localhost/tutorial_yii/rights)

Name	Roles	Tasks	Operations
admin	Super Admin		
demo	Member		
test	Member		

\*\*\* เราจะเห็น User “test” เช็ต role เป็น “Member” ได้

ตัวอย่าง : กำหนด Permissions || ลิ้งค์ที่นี่ [http://localhost/tutorial\\_yii/authItem/permissions](http://localhost/tutorial_yii/authItem/permissions)

Item	Member
No authorization items found.	

\*) Hover to see from where the permission is inherited.

\*\*\* คลิกที่ลิ้งค์ “Generate items for controller actions”

ตัวอย่าง : เพิ่มฟังก์ชัน filters() ใน SiteController เพื่อเรียกใช้งาน Yii Rights Modules

คำสั่ง : PHP ไฟล์เดิมอยู่ที่ : [skeleton]/protected/controllers/SiteController.php

```
class SiteController extends Controller {
    public function filters(){
        return array(
            'rights', // perform access control for CRUD operations
    );
}
.....
```

## กำหนด Permissions ให้กับ User “test” ในส่วนของ SiteController เท่านั้น

ในส่วนของ Controller ชื่อว่า “Site” และ Action (Index,Error,Contact,Login,Logout) จากนั้น ให้กดปุ่ม Generate ด้านล่างสุด

<input type="checkbox"/>	Site.*	controllers\SiteController.php
<input checked="" type="checkbox"/>	Index	controllers\SiteController.php:28
<input checked="" type="checkbox"/>	Error	controllers\SiteController.php:38
<input checked="" type="checkbox"/>	Contact	controllers\SiteController.php:52
<input checked="" type="checkbox"/>	Login	controllers\SiteController.php:78
<input checked="" type="checkbox"/>	Logout	controllers\SiteController.php:104

ตัวอย่าง : ผลลัพธ์ กำหนด Permissions ให้กับ User “test” ในส่วนของ SiteController เท่านั้น

**Permissions**

Here you can view and manage the permissions assigned to each role.  
Authorization items can be managed under Roles, Tasks and Operations.

Generate items for controller actions

Item	Member
Site.Contact	Assign
Site.Error	Assign
Site.Index	Assign
Site.Login	Assign
Site.Logout	Assign

\*\*\* สังเกต เราจะกำหนดให้ User “test” เข้าใช้งานของ SiteController สามารถจัดการ Action (Contact, Error, Index, Login, Logout) พากนี้ได้

ตัวอย่าง : การเรซต Role ให้กับชื่อ Member

ให้เราคลิกที่ลิ้งค์ Assign และแก้ไขเป็น Revoke ตามวงกลมสีแดง เลยครับ โดยผู้จะกำหนดให้ User “test” หรือ Member ทั้งหมด สามารถ เข้าใช้งาน Controller Action : Index และ Login ได้เท่านั้น

ให้เราคลิก คำว่า Assign ของ Site.Index และ Site.Login อย่างละ 1 ครั้ง ให้มันเปลี่ยนจากคำว่า Assign เป็น Revoke ตามภาพ

Item	Member
Site.Contact	Assign
Site.Error	Assign
Site.Index	Revoke
Site.Login	Revoke
Site.Logout	Assign

\*\*\* SiteController ที่เป็น “member” สามารถใช้งาน เฉพาะ action ชื่อ index กับ login ได้เท่านั้น และในส่วนของ role ของ Admin สามารถ access ใช้งานได้ทุกอันอยู่แล้วครับ จึงไม่มีให้กด Assign

ตัวอย่าง : การกำหนด Rule ของ User “test” ที่มี Role ชื่อว่า Member

ให้ล็อกอินเข้าสู่ระบบด้วย User “test” ที่ล็อก [http://localhost/tutorial\\_yii/user/login](http://localhost/tutorial_yii/user/login) และทดสอบ ให้ User “test” เข้าไปปั๊ง action “contact” | [http://localhost/tutorial\\_yii/site/contact](http://localhost/tutorial_yii/site/contact) จะไม่สามารถเข้าถึง ข้อมูลได้ แต่หากเข้าไปปั๊ง action “index” จะเข้าถึงข้อมูลได้

ตัวอย่าง : User “test” เข้าใช้ action “contact” ไม่ได้

## CHttpException

You are not authorized to perform this action. (D:\xampp\htdocs\tutorial\_yii\protected\modules\rights\components\RController.php:62)

```
#0 D:\xampp\htdocs\tutorial_yii\protected\modules\rights\components\RController.php(57): RController->accessDenied()
#1 D:\xampp\htdocs\yii\framework\web\filters\CFilter.php(38): RightsFilter->preFilter(Object(CFilterChain))
#2 D:\xampp\htdocs\tutorial_yii\protected\modules\rights\components\RController.php(36): CFilter->filter(Object(CFilterChain))
#3 D:\xampp\htdocs\yii\framework\web\filters\CInlineFilter.php(58): RController->filterRights(Object(CFilterChain))
#4 D:\xampp\htdocs\yii\framework\web\filters\CFilterChain.php(130): CInlineFilter->filter(Object(CFilterChain))
#5 D:\xampp\htdocs\yii\framework\web\CCController.php(291): CFilterChain->run()
#6 D:\xampp\htdocs\yii\framework\web\CCController.php(265): CCController->runActionWithFilters(Object(CInlineAction), Array)
#7 D:\xampp\htdocs\yii\framework\web\CWebApplication.php(282): CCController->run('error')
#8 D:\xampp\htdocs\yii\framework\base\CErrorHandler.php(331): CWebApplication->runController('site/error')
#9 D:\xampp\htdocs\yii\framework\base\CErrorHandler.php(204): CErrorHandler->render('error', Array)
#10 D:\xampp\htdocs\yii\framework\base\CErrorHandler.php(129): CErrorHandler->handleException(Object(CHttpException))
#11 D:\xampp\htdocs\yii\framework\base\CAplication.php(732): CErrorHandler->handle(Object(CExceptionEvent))
#12 {internal function}: CApplication->handleException(Object(CHttpException))
#13 {main}
```

แต่หากจะกำหนดให้ User “test” สามารถเข้าถึง action “contact” ได้ ให้เราเข้าไปเปลี่ยนค่า จาก Assign เป็น Revoke ของ角色 Site.Contact ได้เลย

การเข้าใช้งานระบบจัดการ Control Access ของ Yii Rights Modules เราต้องล็อกอินเป็น “admin” เท่านั้นนะครับ



សាខាអ៊ីនិក  
fenics128[at]gmail.com

# การพัฒนาเว็บแอ�플ิเคชัน ด้วย Yii Framework

## บทที่ 8 รวมเทคนิคต่าง ๆ ที่น่าสนใจ

แนะนำเทคนิคต่าง ๆ ที่น่าสนใจให้รู้จัก เช่น การใช้งาน Template การแทรกคำสั่ง CSS JavaScript และการใช้งาน Captcha Widget เป็นต้น

### การแทรกคำสั่ง CSS JavaScript

การแทรกคำสั่งหรือไฟล์ CSS และ JavaScript ลงในเว็บแอ�플ิเคชัน นั้น สำหรับ Yii Framework เราสามารถแทรกคำสั่งหรือนำเข้าไฟล์ CSS และ JavaScript จากภายนอกมาใช้งาน ได้เฉพาะ

ตัวอย่าง : การแทรกคำสั่ง CSS, JavaScript

```
$cs = Yii::app()->clientScript; // ประกาศตัวแปร $cs จาก Object :: clientScript  
$css = "คำสั่ง CSS"; // คำสั่ง CSS  
$js = "คำสั่ง JavaScript"; // คำสั่ง JavaScript  
$cs->registerCss("Css Name",$css,CClientScript::POS_HEAD); // แทรกคำสั่ง CSS  
$cs->registerScript("JS Name",$js,CClientScript::POS_END); // แทรกคำสั่ง JavaScript
```

ตัวอย่าง : การนำไฟล์ CSS, JavaScript

```
$cs = Yii::app()->clientScript; // ประกาศตัวแปร $cs จาก Object :: clientScript  
$path = Yii::app()->baseUrl; // ตั้งค่า Path ที่เรากำลังใช้งานอยู่  
$cs->registerCssFile($path."/css/main.css",CClientScript::POS_HEAD); // นำเข้าไฟล์ CSS  
$cs->registerScriptFile($path."/js/main.js",CClientScript::POS_END); // นำเข้าไฟล์ JavaScript
```

ตำแหน่งไฟล์ : ที่ต้องการนำไฟล์หรือแทรกคำสั่ง CSS, JavaScript ไปแสดงไว้

- CClientScript::POS\_HEAD ตำแหน่ง บนสุด
- CClientScript::POS\_BEGIN ตำแหน่ง เริ่มต้น/ตั้งต้น
- CClientScript::POS\_END ตำแหน่ง ท้ายสุด
- CClientScript::POS\_LOAD ตำแหน่ง ระหว่างการโหลด
- CClientScript::POS\_READY ตำแหน่ง ทันที/รวดเร็ว

ตัวอย่าง : นำเข้าไฟล์ JavaScript ของ Yii Framework

```
Yii::app()->clientScript->registerCoreScript('jquery'); // file jquery.js  
Yii::app()->clientScript->registerCoreScript('jquery.ui'); // file jquery-ui.js
```

ตัวอย่าง : นำเข้าไฟล์ CSS, JavaScript จาก Extension ตัวอย่าง Yii booster Extension

```
// booster assets path :: /protected/extensions/booster/assets
$assetsUrl = Yii::app()->getAssetManager()->publish(Yii::getPathOfAlias('ext.booster.assets'));
Yii::app()->clientScript->registerScriptFile($assetsUrl.'bootstrap/js/bootstrap.min.js');
Yii::app()->clientScript->registerCssFile($assetsUrl.'bootstrap/css/bootstrap.min.css');
```

ตัวอย่าง : นำเข้าไฟล์ CSS, JavaScript จาก Widget ตัวอย่าง Zii Widget

```
$url = Yii::app()->getAssetManager()->publish(Yii::getPathOfAlias('zii.widgets.assets'));
Yii::app()->clientScript->registerScriptFile($url.'listview/jquery.yiilistview.js',CClientScript::POS_END);
Yii::app()->clientScript->registerCssFile($url.'listview/styles.css',CClientScript::POS_END);
```

ตัวอย่าง : ระบุ Path ของ CSS, JavaScript ในส่วนของ Component ไฟล์ Config (main.php)

คำสั่ง : PHP

ไดเรกทอรี่ : [skelton]/protected/config/main.php

```
'components'=>array(
    'clientScript' => array(
        'packages'=>array(
            'test.jquery'=>array(
                'baseUrl'=>'//ajax.googleapis.com/ajax/libs/jquery/1/',
                'js'=>array('jquery.min.js'),
            ),
            'test.jquery.ui'=>array(
                'baseUrl'=>'//ajax.googleapis.com/ajax/libs/jqueryui/1/',
                'js'=>array('jquery-ui.min.js'),
            ),
        ),
    ),
),
```

การเรียกใช้ : Path ของ CSS, JavaScript

```
Yii::app()->clientScript->registerCoreScript('test.jquery');
// file jquery-min.js
```

```
Yii::app()->clientScript->registerCoreScript('test.jquery.ui');
// file jquery.ui.js
```

## การแทรกคำสั่ง MetaTag

การแทรกคำสั่ง MetaTag ลงในเว็บแอปพลิเคชัน นั้น สำหรับ Yii Framework เราสามารถแทรกคำสั่ง MetaTag ได้แบบง่าย ๆ

ตัวอย่าง : คำสั่ง MetaTag “คำสั่งเดียว”

```
Yii::app()->clientScript->registerMetaTag('This is customized for the site keys list', 'description');
/* output */
// <meta name="description" content="This is customized for the site keys list" />
```

ตัวอย่าง : คำสั่ง MetaTag “หลายคำสั่ง”

```
$metaTags = array(
    'description'=>'This is customized for the site keys list', // Meta tags Search Engine
    'keywords'=>'keyword one, keyword two, keyword three', // Meta tags Search Engine
    'language'=>'en', // Meta tags สำหรับออกภาษาที่ใช้ในเว็บ
    'author'=>'Drive Software Center', // Meta tags สำหรับบอกชื่อผู้เขียนหน้านี้ และลิขสิทธิ์
    'generator'=>'Dreamweaver', // Meta tags สำหรับบอกเครื่องมือที่ใช้สร้างเว็บเพจนี้
    'contact_addr'=>'demo@gmail.com', // Meta tags สำหรับใส่อีเมลสำหรับให้คนติดต่อกลับ
);
foreach(array_keys($metaTags) as $metaTag){
    Yii::app()->clientScript->registerMetaTag($metaTags[$metaTag],$metaTag);
}

/* output */
// <meta name="description" content="This is customized for the site keys list" />
// <meta name="keywords" content="keyword one, keyword two, keyword three" />
// <meta name="language" content="en" />
// <meta name="author" content="Drive Software Center" />
// <meta name="generator" content="Dreamweaver" />
// <meta name="contact_addr" content="demo@gmail.com" />
```

ตัวอย่าง : การแสดงผล “รูปภาพ” ด้วยคำสั่ง CHtml::image()

```
// รูปแบบ image($imgPath);
CHtml::image(Yii::app()->request->baseUrl.'./images/yii.jpg', 'this is alt', array(
    "width"=>"50","height"=>"50"
));
// 
```

ตัวอย่าง : การแสดงผล “ลิงค์” ด้วยคำสั่ง CHtml::link()

```
echo CHtml::link('Link Text',array('controller/action','param1=>'value1'));
// <a href="/tutorial_yii/controller/action?param1=value1">Link Text</a>

echo CHtml::link('Link Text',array('controller/action'), array('target'=>'_blank'));
// <a target="_blank" href="/tutorial_yii/controller/action">Link Text</a>

echo CHtml::link('Link Text',array('action'));
// <a href="/tutorial_yii/chtml/action">Link Text</a>

echo CHtml::link('Link Text',array('/controller/action'));
// <a href="/tutorial_yii/controller/action">Link Text</a>

$imgPath = 'img/photo.jpg'; // Path ที่เก็บไฟล์รูปภาพ
$image = CHtml::image($imgPath); // สร้างคำสั่งแสดงรูปภาพ
echo CHtml::link($image, $imgPath);
// <a href="img/photo.jpg"></a>
```

ตัวอย่าง : การแสดงผล “ลิงค์” ด้วยคำสั่ง CHtml::ajaxLink()

```
// in View
echo CHtml::ajaxLink('Test request',array('ajax/reqTest'),array('update'=>'#req_res'));
echo '<div id="req_res">...</div>';

// in Controller
public function actionReqTest() {
    echo date('H:i:s');
    Yii::app()->end();
}

// <a href="#" id="yt0">Test request</a>
// <div id="req_res">...</div>

/* <script type="text/javascript">
jQuery('body').on('click','#yt0',function(){
    jQuery.ajax({url:'/tutorial_yii/link/reqTest','cache':false,'success':function(html){
        jQuery("#req_res").html(html)
    });
    return false
});
/* </script>
```

ตัวอย่าง : การแสดงผล “ปุ่ม” ด้วยคำสั่ง CHtml::button()

```

CHtml::button('Button Text', array('submit' => array('controller/action')));
// <input name="yt0" type="button" value="Button Text" id="yt0" />
/* <script type="text/javascript">
jQuery('body').on('click','#yt0',function(){
    jQuery yii.submitForm(this,'/tutorial_yii/controller/action',{});
    return false;
});
*/
/* </script>

CHtml::button('Delete', array(
    'name'=>'btnDelete',
    'submit' => array('user/delete','id'=>1),
    'confirm'=>'Are you sure you want to delete?',
    'class'=>'btn btn-large btn-danger',
    'style'=>'width:160px;';
));
// <input name="btnDelete" class="btn btn-large btn-danger" style="width:160px;" type="button" value="Delete" id="btnDelete" />
/* <script type="text/javascript">
jQuery('body').on('click','#btnDelete',function(){
    if(confirm('Are you sure you want to delete?')){
        jQuery yii.submitForm(this,'/tutorial_yii/user/delete/1',{});
        return false;
    }else{
        return false;
    }
});
*/
/* </script>

```

ตัวอย่าง : การแสดงผล “รายการข้อมูล” ด้วยคำสั่ง CHtml::listData()

```

$models = categories::model()->findAll();
$list = CHtml::listData($models,'category_id','category_name');
// output array("1"=>"Arts", "2"=>"Science","3"=>"Culture");

$models = categories::model()->findAll(array('order'=>'category_name'));
$list = CHtml::listData($models,'category_id','category_name');
// output array("1"=>"Arts","3"=>"Culture", "2"=>"Science");

```

## การใช้งาน CCaptcha Widget

การใช้งาน Widget CCaptcha ต้องเรียกใช้งานผ่านเมธอด CActiveForm เท่านั้น ซึ่งเราจะมาเรียนรู้วิธีการใช้งาน ดังต่อไปนี้

1. ต้องมีชื่อฟิลด์ verifyCode อยู่ใน “Model” ที่ต้องการสร้าง Captcha

2. แก้ไขไฟล์โมเดล (Model) และเพิ่มตัวแปร \$verifyCode; ฟังก์ชัน rule(), attributeLabels() ดังนี้

```
public $verifyCode; // เพิ่มเติม
public function rules(){
    return array(
        .....
        array('verifyCode', 'captcha', 'allowEmpty'=>!CCaptcha::checkRequirements()), // เพิ่มเติม
    );
}
public function attributeLabels(){
    return array(
        .....
        'verifyCode' => 'Verification Code', // เพิ่มเติม
    );
}
```

3. แก้ไขไฟล์ Controller ที่ต้องการแสดง Captcha และ เพิ่มเติมหรือแก้ไขเมธอด actions() ดังนี้

```
public function actions(){
    return array(
        // เพิ่มคำสั่ง 'captcha'
        'captcha'=>array(
            'class'=>'CCaptchaAction',
            'backColor'=>0xFFFFFF,
        ),
    );
}
```

4. เพิ่มคำสั่งในการแสดงผล View ของ Captcha

```
<?php $form = $this->beginWidget('CActiveForm'); ?>
<?php if(CCaptcha::checkRequirements()): ?>
    <?php echo $form->labelEx($model,'verifyCode'); ?>
    <?php $this->widget('CCaptcha', array('captchaAction'=>'site/captcha'))); ?>
    <?php echo $form->textField($model,'verifyCode'); ?>
    <?php echo $form->error($model,'verifyCode'); ?>
<?php endif; ?>
<?php $this->endWidget(); ?>
```

ตัวอย่าง : การแสดงผล Captcha



### การใช้งาน Session

เป็นการสาธิต การนำ Session เข้ามาใช้งานหลังจากที่มีการเก็บ Session ไว้แล้ว โดยการนำ Session มาใช้งาน ซึ่งจะมีวิธีการดังต่อไปนี้

ตัวอย่าง : การประกาศตัวแปร Session แบบที่ 1

```
$session = new CHttpSession();
=session->open();
$session['key'] = 'ทดสอบ Session แบบที่ 1';
echo $session['key'];
=session->close();
// output : ทดสอบ Session แบบที่ 1'
```

ตัวอย่าง : การประกาศตัวแปร Session แบบที่ 2

```
Yii::app()->session['key'] = 'ทดสอบ Session แบบที่ 2';
$value = Yii::app()->session['key'];
echo $value;
// output : ทดสอบ Session แบบที่ 2
```

ตัวอย่าง : เก็บค่า Array ลงใน Session

```
Yii::app()->session['key3'] = array(
    'key1'=>'value1',
    'key2'=>'value2',
    'key3'=>'value3',
);
```

ตัวอย่าง : การลบข้อมูล Session

```
$session = new CHttpSession();
=session->open();
$session['key'] = 'ทดสอบ Session';
unset($session['key']); // ลบ Session วิธีที่ 1
=session->close();
Yii::app()->session->remove('key'); // ลบ Session วิธีที่ 2
```

ตัวอย่าง : การประภาคตัวแปร GET บน URL

```
// รูปแบบตัวแปร GET  
// URL :: <controller>/<action>/<id>/<name1>/<value1>/<name2>/<value2>/  
URL :: post/save/1/user/root/pass/1234  
// $_GET['user'] = root  
// $_GET['pass'] = 1234
```

### การใช้งาน Cookie

เป็นการสำคัญ การนำ Cookie เข้ามาใช้งานหลังจากที่มีการเก็บ Cookie ไว้แล้ว โดยการนำ Cookie มาใช้งาน ซึ่งจะมีวิธีการดังต่อไปนี้

ตัวอย่าง : การประภาคตัวแปร Cookie

```
$cookie = new CHttpCookie('key1','value1');  
$cookie->expire = time() + 3600; // an hour: 3600 seconds,  
Yii::app()->request->cookies['key1'] = $cookie;
```

### ระยะเวลา “อายุการใช้งาน”

- 1 ชั่วโมง : 3600 seconds.
- 1 วัน : 86400 seconds.
- 1 สัปดาห์ : 604800 seconds.
- 1 เดือน : 2592000 seconds.
- 1 ปี : 31104000 seconds.

ตัวอย่าง : ดึงข้อมูล Cookie

```
Yii::app()->request->cookies['key1'];
```

ตัวอย่าง : ล้างข้อมูล Cookie เฉพาะ Key1

```
unset(Yii::app()->request->cookies['key1']);
```

ตัวอย่าง : ล้างข้อมูล Cookie ทั้งหมด

```
Yii::app()->request->cookies->clear();
```

## สร้างฟังก์ชันสำหรับใช้งาน Cookie

เราจะสร้างฟังก์ชันสำหรับเรียกใช้งาน Cookie โดยเราจะสร้างไฟล์ชื่อว่า “CookiesHelper.php” ไว้ในโฟลเดอร์ components

คำสั่ง : PHP

ไฟล์存放位置 : [skelton]/protected/component/CookiesHelper.php

```
class CookiesHelper extends CApplicationComponent {
    public function putCMsg($name, $value) {
        if (is_array($value))
            Yii::app()->request->cookies[$name] = new CHttpCookie($name, json_encode($value));
        if (is_string($value))
            Yii::app()->request->cookies[$name] = new CHttpCookie($name, $value);
        return true;
    }
    public function getCMmsg($name) {
        if (!empty(Yii::app()->request->cookies[$name])) {
            $return = json_decode(Yii::app()->request->cookies[$name]);
            if (json_last_error() == JSON_ERROR_NONE && !is_string($return))
                return $return;
            else
                return Yii::app()->request->cookies[$name]->value;
        }else
            return 'Cookie not found';
    }
    public function updateCMmsg($name, $value) {
        if (!empty(Yii::app()->request->cookies[$name])) {
            $return = json_decode(Yii::app()->request->cookies[$name]);
            if (json_last_error() == JSON_ERROR_NONE && !is_string($return)) {
                array_push($return, $value);
                Yii::app()->request->cookies[$name] = new CHttpCookie($name, json_encode($return));
                return true;
            } else {
                Yii::app()->request->cookies[$name] = new CHttpCookie($name, $value);
                return true;
            }
        }else
            return 'Cookie not found';
    }
    public function delCMmsg($name = NULL) {
        unset(Yii::app()->request->cookies[$name]);
        return true;
    }
}
```

ตัวอย่าง : กำหนดค่าในไฟล์ Config (main.php)

คำสั่ง : PHP

ไดเรกทอรี่ : [skelton]/protected/config/main.php

```
'components'=>array(  
    ....  
    'Cookies'=>array(  
        'class'=>'application.components.CookiesHelper'  
    ),  
,
```

ตัวอย่าง : เรียกใช้งาน “ฟังก์ชัน Cookie” ที่เราสร้าง

```
// สร้าง Cookie "cookie_name1"  
Yii::app()->Cookies->putCMsg('cookie_name1',array('myCookieVal1','val2'));  
  
// ดึงข้อมูล Cookie "cookie_name1"  
$cookie = Yii::app()->Cookies->getCMsG('cookie_name1');  
print_r($cookie); // แสดงข้อมูล  
// array('myCookieVal1','val2')  
  
// เพิ่มเดินข้อมูล Cookie "cookie_name1"  
Yii::app()->Cookies->updateCMsG('cookie_name1','orange');  
// array('myCookieVal1','val2','orange')  
  
// ลบข้อมูล Cookie "cookie_name1"  
Yii::app()->Cookies->delCMsG('cookie_name1');
```

### การ redirect สำหรับ Yii Framework

การ redirect คือการเปลี่ยนหน้า page ปัจจุบันของเรามาไปยัง page อื่น ได้ทั้ง link ไป page กายในและ page กายนอกเว็บไซต์ของเรา

ตัวอย่าง : รูปแบบคำสั่ง redirect

```
redirect(mixed $url, boolean $terminate=true, integer $statusCode=302)
```

ตัวอย่าง : redirect ไปยังเว็บไซต์ภายนอก

```
$this->redirect("http://www.yii.in.th");
```

ตัวอย่าง : redirect ไปยังไฟล์ HTML ในเว็บของเรา

```
$this->redirect("/path/page.html");
```

ตัวอย่าง : redirect ไปยัง controller และ action ที่ต้องการ

```
$this->redirect("controller/action");
```

ตัวอย่าง : redirect โดยส่งค่าไปทาง get ด้วย

```
$this->redirect(array('controller/action','id'=>3, 'title'=>'Hello'));
// หรือ จะเท่ากับ index.php?r=controller/action&id=3&title=Hello
```

ตัวอย่าง : redirect แบบกำหนดไว้ในไฟล์ config/main.php

```
'components'=>array(
    'urlManager'=>array(
        .....
        'rules'=>array(
            'backoffice' => '/site/backoffice', // ทดสอบเพิ่ม
        .....
    )
)
```

ทดสอบ : หากเราเข้าไปที่ลิงค์ [http://localhost/tutorial\\_yii/backoffice](http://localhost/tutorial_yii/backoffice)

yii จะไปเรียกไฟล์ที่ชื่อว่า backoffice ที่อยู่ในไฟล์เครื่อง views/site/backoffice.php

### การ render สำหรับ Yii Framework

การกำหนดค่าว่าจะให้นำไฟล์ไหนขึ้นมาแสดงผลและสามารถการแนบตัวแปรข้อมูลลงไปในไฟล์ที่เราดำเนินการแสดงผลนั้น ๆ ได้อีกด้วย

ตัวอย่าง : รูปแบบคำสั่ง render

```
render('ชื่อไฟล์',array('ชื่อตัวแปร1'=>'ค่าตัวแปร1', 'ชื่อตัวแปร...'=>'ค่าตัวแปร..'))
```

ตัวอย่าง : ต้องการแสดงไฟล์ index.php ในไฟล์เครื่อง protected/views/site/index.php

```
$this->render('index');
```

ตัวอย่าง : ต้องการแสดงไฟล์ index.php และแนบตัวแปร model,dataProvider เข้าไปด้วย

```
$this->render('index', array('model' => $model, 'dataProvider' => $dataProvider));
```

ตัวอย่าง : ต้องการแสดงไฟล์ index.php ในไฟล์เครื่อง protected/views/site/content/index.php

```
$this->render('content/index');
```

ตัวอย่าง : ต้องการแสดงไฟล์ index.php ในไฟล์เครื่อง protected/views/site/page/content/index.php

```
$this->render('page/content/index');
```

## ระบบ Log ของ Yii Framework

Log คือ รูปแบบที่ใช้ในการเก็บข้อมูลสถานะต่าง ๆ แสดงข้อผิดพลาดที่เกิดขึ้น เพื่อตรวจสอบความถูกต้องของระบบที่เราพัฒนาขึ้น

Yii นั้นได้มีระบบ Log มาให้เราใช้งานมันอย่างง่ายดาย วิธีการใช้งาน อันดับแรกเราต้องไปเปิด config การใช้งาน log ก่อนครับ (โดยปกติระบบจะเปิดมาให้อยู่แล้ว)

ตัวอย่าง : เข้าไปที่ไฟล์ [skeleton]/protected/config/main.php

```
'components'=>array(
    'log'=>array(
        'class'=>'CLogRouter',
        'routes'=>array(
            array(
                'class'=>'CFileLogRoute',
                'levels'=>'error, warning',
            ),
            ),
        ),
),
```

สิ่งที่ต้องสนใจ “จุดแรก”

```
'class'=>'CFileLogRoute'
```

อธิบายเพิ่มเติม

1. CFileLogRoute เป็นการกำหนดให้ log ที่เกิดขึ้น ทำการเก็บไว้ในไฟล์ runtime/application.log
2. CDbLogRoute เก็บ log ไว้ใน Database
3. CEmailLogRoute ส่ง log ไปทาง email ที่กำหนด
4. CWebLogRoute แสดง log ที่ด้านล่างของหน้าเว็บ
5. CProfileLogRoute จะแสดง log แบบด้านล่างของหน้าเว็บ มีการกำหนดพื้นที่ตรวจสอบ log ได้

สิ่งที่ต้องสนใจ “จุดสอง”

```
'levels'=>'error, warning',
```

อธิบายเพิ่มเติม

1. ส่วนนี้เป็นการ บอกว่า log ที่เก็บนั้น เก็บเฉพาะ log ที่ level เป็นแบบ error และ warning เท่านั้น
2. level นี้จะมีอยู่ 5 แบบ คือ trace , info , profile , warning , error
3. เรื่อง error ของ Yii มันจะไม่ใช่ error ของคำสั่งภาษา php นะครับ แต่มันจะเป็น error ของระบบ การทำงานของ Yii Framework ที่ถูกกำหนดรูปแบบไว้

เพิ่มเติม : ในส่วนที่ config เก็บ log ใน db นั้นรับ แค่เพิ่ม config เข้าไป

```
array(
    'class' => 'CDbLogRoute',
    'levels' => 'trace, info, warning, error', //อยากเก็บอะไรก็เพิ่มลดลงได้ครับ
    'connectionID' => 'db',
    'autoCreateLogTable' => true, //ให้มันสร้างตารางให้อัตโนมัติ
),
```

## ทดสอบระบบ Log ของ Yii Framework

ก่อนอื่นให้เราสร้างไฟล์ Controller ชื่อว่า log ก่อน โดยใช้ Gii Controller Generator เป็นตัวสร้าง Controller หลังจาก config ตามข้างบนแล้ว ทดลองเขียน action ให้ผิดๆ แล้วลองรันหน้าเว็บ เพื่อให้ yii มัน error ตามตัวอย่างด้านล่างนี้

```
public function actionIndex()
{
    $this->renderxxx('index'); // ทดลองเขียนคำสั่ง render แบบผิด ๆ
}
```

### CException

LogController and its behaviors do not have a method or closure named "renderxxx".

```
D:\xampp\htdocs\yii\framework\base\CComponent.php(266)

254     public function __call($name,$parameters)
255     {
256         if($this->_m!==null)
257         {
258             foreach($this->_m as $object)
259             {
260                 if($object->getEnabled() && method_exists($object,$name))
261                     return call_user_func_array(array($object,$name),$parameters);
262             }
263         }
264         if(class_exists('Closure', false) && $this->canGetProperty($name) && $this->$name instanceof Closure)
265             return call_user_func_array($this->$name,$parameters);
266         throw new CException(Yii::t('yii','{class} and its behaviors do not have a method or closure named "{name}"',
267             array('{class}'=>get_class($this), '{name}'=>$name)));
268     }
269
270     /**
271      * Returns the named behavior object.
272      * The name 'asa' stands for 'as a'.
273      * @param string $behavior the behavior name
274      * @return IBehavior the behavior object, or null if the behavior does not exist
275      */
276     public function asa($behavior)
277     {
278         return isset($this->_m[$behavior]) ? $this->_m[$behavior] : null;
279     }
280 }
```

\*\*\* ตรวจสอบไฟล์ Log ให้เข้าไปที่ :: [skeleton]/protected/runtime/application.log แล้วลองเปิดไฟล์นี้คุณจะเจอ error และในกรณีที่มี error ใหม่ มันจะมาต่อท้ายบรรทัดนี้ไปเรื่อยๆ ครับ ทำให้เราสามารถดูข้อมูลหลังได้ และมีเวลากำหนดไว้ชัดเจนด้วย

สร้างคลาส (Class) ไว้ใช้งาน แบบเรียกผ่าน คำสั่ง Yiiapp();

จะมานำเสนอวิธีการสร้าง Class ใช้เองแบบเรียกใช้ผ่าน Yii::app()->ชื่อคลาส ครับ ซึ่งวิธีนี้คลาสของเรายังไม่เรียกใช้จะเป็น lazy load ด้วยครับ

### ขั้นตอนการสร้าง Class

1. เตรียมตั้งชื่อ Class และชื่อไฟล์ Class
2. สร้างไฟล์ชื่อ MyClass.php ไว้ในโฟลเดอร์ components
3. ตั้งค่าเลือกใช้งาน Class ที่เราสร้างไว้ในไฟล์ Config ในส่วนของ Components
4. เรียกใช้งาน Class ที่เราสร้าง

เตรียมตั้งชื่อ Class และชื่อไฟล์ Class

1. CMyClass = ชื่อคลาสที่เราจะใช้เรียก
2. MyClass = ชื่อคลาสจริง ๆ ของเรา
3. MyVar = ชื่อตัวแปร ตั้งต้นที่ใช้กับคลาสเรา

สร้างไฟล์ชื่อ MyClass.php ไว้ในโฟลเดอร์ components

สร้างไฟล์ชื่อว่า MyClass.php ไว้ในไดเรกทอร์ [skeleton]/protected/components/MyClass.php

```
class MyClass extends CApplicationComponent {  
    public $MyVar = array();  
    public function getVersion(){  
        return 'CMyClass v.0.0.1'; // คืนค่าที่ต้องการแสดงออก  
    }  
}
```

ตั้งค่าเลือกใช้งาน Class ที่เราสร้างไว้ในไฟล์ Config ในส่วนของ Components

```
return array(  
    'components'=>array(  
        'CMyClass'=>array(  
            'class'=>'MyClass',  
            'MyVar'=>array('test'=>'value'),  
        ),  
        .....  
    ),
```

เรียกใช้งาน Class ที่เราสร้าง

```
Yii::app()->CMyClass; // คำสั่งเรียกใช้งาน คลาส MyClass
```

```
Yii::app()->CMyClass->getVersion(); // คำสั่งเรียกใช้งาน ฟังก์ชัน getVersion ภายใน คลาส MyClass
```

## สร้างเว็บแอพพลิเคชั่น รองรับหลายภาษา (ไทย, อังกฤษ, จีน เป็นต้น)

การทำเว็บหลายภาษา เป็นสิ่งที่คนทำเว็บมักจะเจอกันบ่อย ๆ ในบทนี้เราจะมาทดลองทำเว็บ 2 ภาษา คือ ภาษาไทย และ ภาษาอังกฤษ แบบง่าย ๆ กัน เพื่อให้เห็นถึงการทำงานของมัน

### เตรียมไฟล์เดอร์สำหรับเก็บภาษาไว้ก่อน

ริ่มแรก เราต้องไปสร้างไฟล์เดอร์ภาษาที่เรามี เตรียมไว้ก่อนนะครับ เช่นไปสร้างไฟล์เดอร์ “en” และ “th” ในไดเรกทอร์ [skeleton]/protected/messages/ เราจะได้ตามรูป 2 ไฟล์เดอร์ ด้านล่างนี้



จากนั้น เช่นไปในไฟล์เดอร์ “en” และสร้างไฟล์ภาษา โดยจะสร้างไฟล์ ไฟล์คือภาษาสำหรับหน้า home (บางคนถ้าอยากรู้ว่าเป็นอะไรก็ตามเรียบร้อย ก็แยกเป็นไฟล์ภาษาเป็นแต่ละหน้า หรือบางคนจะใช้รวมไฟล์เดียวเลยก็ได้) ในไดเรกทอร์ [skeleton]/protected/messages เราจะได้ตามรูป 2 ไฟล์เดอร์ ด้านล่างนี้

สร้างไฟล์ : home.php เก็บภาษา en, th ไว้ในไฟล์เดอร์ en, th

```
return array(
    'key1' => 'message 1',
    'key2' => 'message 2',
);
```

แก้ไขไฟล์ : ข้อว่า home.php ในไฟล์เดอร์ en

```
return array(
    'welcome' => 'Welcome to my site',
    'what_lang' => 'This is English language',
);
```

แก้ไขไฟล์ : ข้อว่า home.php ในไฟล์เดอร์ th

```
return array(
    'welcome' => 'ยินดีต้อนรับสู่เว็บของฉัน',
    'what_lang' => 'คุณกำลังใช้ภาษาไทย',
);
```

ตัวอย่าง : การเรียกใช้งาน ภาษา en, th

```
// Yii::t('category', 'key'); category ก็คือ ชื่อไฟล์ key ก็คือ ค่าในไฟล์ภาษา ที่มันจะนำไปหา message
<h1><?php echo Yii::t('home', 'welcome');?></h1>
<?php echo Yii::t('home', 'what_lang');?>
```

สร้างไฟล์ : Controller ชื่อว่า “LangController” เพื่อนำทดสอบ “การเปลี่ยนภาษา”

คำสั่ง : PHP

ไฟร์เครื่องท่อรี่ : [skeleton]/protected/controllers/LangController.php

```
class LangController extends Controller {  
    public function actionIndex(){  
        Yii::app()->language = 'en';  
        $this->render('index');  
    }  
}
```

คำสั่ง : PHP

ไฟร์เครื่องท่อรี่ : [skeleton]/protected/views/lang/index.php

```
<h1><?php echo Yii::t('home', 'welcome');?></h1>  
<?php echo Yii::t('home', 'what_lang');?>
```

ตัวอย่าง : ‘en’ คือ อังกฤษ

## Tutorial Yii Framework

Home About Contact Login

Home » Lang

### Welcome to my site

This is English language

Copyright © 2014 by My Company.  
All Rights Reserved.  
Powered by [Yii Framework](#).

ตัวอย่าง : ‘th’ คือ ภาษาไทย

## Tutorial Yii Framework

Home About Contact Login

Home » Lang

### ยินดีต้อนรับสู่เว็บของฉัน

คุณกำลังใช้ ภาษาไทย

Copyright © 2014 by My Company.  
All Rights Reserved.  
Powered by [Yii Framework](#).

## สร้างเมนู “สลับภาษา”

เราจะมาสร้างเมนู ไว้ใช้ในการ สลับภาษา เช่น การเปลี่ยนภาษา ให้ได้ตามเรากำหนดที่เราต้องการ ให้แสดงผลลัพธ์ ออกมา โดยเราจะเข้าไปเพิ่มฟังก์ชัน actionSetlanguage() ในไฟล์ LangController เพื่อใช้สลับภาษาใน การแสดงผล

ตัวอย่าง : สร้างฟังก์ชัน actionSetlanguage() ใน LangController.php

คำสั่ง : PHP

ไฟล์เดิมที่อยู่ : [skeleton]/protected/controllers/LangController.php

```
class LangController extends Controller {
    .....
    public function actionSetlanguage() {
        $session = new CHttpSession;
        $session->open();
        $session['sess_lang'] = strtolower($_GET['lang']);
        $this->redirect(array('lang/index'));
    }
}
```

ตัวอย่าง : แก้ไขฟังก์ชัน actionIndex() ใน LangController.php

คำสั่ง : PHP

ไฟล์เดิมที่อยู่ : [skeleton]/protected/controllers/LangController.php

```
class LangController extends Controller {
    public function actionIndex() {
        $session = new CHttpSession;
        $session->open();
        Yii::app()->setLanguage(isset($session['sess_lang'])? $session['sess_lang']:'th');
        // กำหนด default ภาษา เป็น th
        $this->render('index');
    }
    .....
}
```

ตัวอย่าง : แก้ไขไฟล์ View (index.php)

คำสั่ง : PHP

ไฟล์เดิมที่อยู่ : [skeleton]/protected/views/lang/index.php

```
<?php
echo 'Select Language: ';
echo CHtml::link('TH', array('lang/setlanguage','lang'=>'th')); echo ' | ';
echo CHtml::link('EN', array('lang/setlanguage','lang'=>'en'));
?>
<br /><br /><hr />
<h1><?php echo Yii::t('home', 'welcome');?></h1>
<?php echo Yii::t('home', 'what_lang');?>
```

ตัวอย่าง : ทดสอบกดลิ้งค์คำว่า “EN” เพื่อเปลี่ยนเป็นภาษา “อังกฤษ”

## Tutorial Yii Framework

Home About Contact Login

Select Language: [TH](#) | [EN](#)

### Welcome to my site

This is English language

Copyright © 2014 by My Company.  
All Rights Reserved.  
Powered by [Yii Framework](#)

ตัวอย่าง : ทดสอบกดลิ้งค์ “TH” เพื่อเปลี่ยนเป็นภาษา “ภาษาไทย”

## Tutorial Yii Framework

Home About Contact Login

Select Language: [TH](#) | [EN](#)

### ยินดีต้อนรับสู่เว็บของฉัน

คุณกำลังใช้ ภาษาไทย

Copyright © 2014 by My Company.  
All Rights Reserved.  
Powered by [Yii Framework](#).



## สร้าง Module เพื่อแยกระบบการทำงานออกจากระบบหลัก

การสร้าง Modules ชี้อว่า Wiki เพื่อสร้างระบบงานย่อย ๆ เพื่อใช้สำหรับแยกการทำงานที่เป็นระบบย่อย ๆ ให้การทำงานออกจากระบบหลัก ของเว็บแอปพลิเคชัน เราชรับ ความสามารถประยุกต์ใช้งาน

## สร้าง Modules ด้วย Gii Module Generator

### ตั้งค่าเรียกใช้งาน Modules ในไฟล์ Config (main.php)

คำสั่ง : PHP

ไดเรกทอรี่ : [skeleton]/protected/config/main.php

```
return array(
    .....
    'modules'=>array(
        .....
        'wiki', // เพิ่มเติม Modules ใหม่
    ),
    .....
)
```

สร้างฐานข้อมูลใหม่ ตาราง ชื่อว่า “wiki” และสร้าง Model || ดาวโหลดที่นี่ <http://goo.gl/r0oZEy>

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1 <b>id</b>	int(6)			No	<i>None</i>	
<input type="checkbox"/>	2 <b>text</b>	text	utf8_general_ci		Yes	<i>NULL</i>	

แก้ไขไฟล์ DefaultController.php และเพิ่ม Action ตามคำสั่งดังต่อไปนี้

เพิ่มเติม : พื้นที่ชั้น “แสดงข้อมูล GridView”

คำสั่ง : PHP

ไฟล์存放位置 : [skeleton]/protected/modules/wiki/controllers/DefaultController.php

```
public function actionIndex($id=''){
    if(!empty($id)){
        $this->actionView($id);
    }else{
        $dataProvider = new CActiveDataProvider('Wiki',array(
            'pagination'=>array(
                'pageSize'=gt;10, // กำหนดให้หน้าบังคับ 10 รายการ
            ),
        )));
        $this->render('index',array(
            'dataProvider'=>$dataProvider
        ));
    }
}
```

เพิ่มเติม : พื้นที่ชั้น “เพิ่มข้อมูล”

คำสั่ง : PHP

ไฟล์存放位置 : [skeleton]/protected/modules/wiki/controllers/DefaultController.php

```
public function actionCreate(){
    if(empty($_POST['Wiki'])){
        $model = new Wiki();
        $this->render('create',array(
            'model' => $model
        ));
    }else{
        $model = new Wiki();
        $this->performAjaxValidation($model);
        $model->text = $_POST['Wiki']['text'];
        $model->save();
        echo '<hr>เพิ่มข้อมูลแล้ว :: '.$_POST['Wiki']['text'];
        echo '<br>['.CHtml::link('แก้ไข',array(
            'edit',
            'id'=>$model->id
        )).]';
    }
}
```

**เพิ่มเติม : พังก์ชัน “แสดงข้อมูล View”**

คำสั่ง : PHP

ไฟล์ controller : [skelton]/protected/modules/wiki/controllers/DefaultController.php

```
public function actionView($id){
    if(!empty($id)){
        $model = Wiki::model()->findByPk($id);
        if(!$model){
            $this->actionEdit($id);
            Yii::app()->end();
        }
        $this->render('view',array(
            'model' => $model
        ));
    }
}
```

**เพิ่มเติม : พังก์ชัน “แก้ไขข้อมูล”**

คำสั่ง : PHP

ไฟล์ controller : [skelton]/protected/modules/wiki/controllers/DefaultController.php

```
public function actionEdit($id=""){
    $model = Wiki::model()->findByPk($id);
    if(!$model){
        $model = new Wiki();
        $model->id = $id;
    }
    if(!empty($_POST['Wiki'])){
        if(!empty($_POST['Wiki']['text'])){
            $model->text = $_POST['Wiki']['text'];
            if($model->save()){
                $this->redirect(array(
                    'view','id' => $id
                ));
            }else{
                Wiki::model()->deleteByPk($id);
            }
        }
    }
    $this->render('edit',array(
        'model'=> $model
    ));
}
```

เพิ่มเติม : พังก์ชั่น “ลบข้อมูล”

คำสั่ง : PHP

ไฟร์แคร็ฟต์ : [skeleton]/protected/modules/wiki/controllers/DefaultController.php

```
public function actionDelete($id=''){
    $model = Wiki::model()->findByPk($id);
    if($model){
        $model->delete();
    }
    $this->redirect(array(
        'index'
    ));
}
```

เพิ่มเติม : พังก์ชั่น “AjaxValidation”

คำสั่ง : PHP

ไฟร์แคร็ฟต์ : [skeleton]/protected/modules/wiki/controllers/DefaultController.php

```
protected function performAjaxValidation($model){
    if(isset($_POST['ajax']) && $_POST['ajax']=='comment-form'){
        echo CActiveForm::validate($model);
        Yii::app()->end();
    }
}
```

สร้างไฟล์ Views (view.php)

คำสั่ง : PHP

ไฟร์แคร็ฟต์ : [skeleton]/protected/modules/wiki/views/default/view.php

```
<h3>
ID :: #<?php echo CHtml::encode($model->id);?>
[<?php echo CHtml::link('แก้ไข',array('edit','id' => $model->id));?>] |
[<?php echo CHtml::link('ลบ',array('delete','id' => $model->id));?>]
</h3>
เนื้อหา :: <?php echo $model->text; ?><br>
```

สร้างไฟล์ Views (create.php)

คำสั่ง : PHP

ไฟร์แคร็ฟต์ : [skeleton]/protected/modules/wiki/views/default/create.php

```
<h2>สร้างข้อมูลใหม่</h2>
<div class="form">
<?php echo CHtml::beginForm();?>
<?php echo CHtml::activeLabelEx($model,'text'); ?>
<?php echo CHtml::activeTextArea($model, 'text',array(
    'cols' => 100, 'rows' => 10,'style' => 'width:98%;'));;
?>
```

```
<?php echo CHtml::ajaxSubmitButton(
    'บันทึกข้อมูล',array('save'=>array('id'=>$model->id)),
    array(
        'update'=>'#req_result',
        'beforeSend' => 'function(call,settings){
            $("#req_result").html("กำลัง ประมวลผล กรุณารอ..");
        }',
    )
);
?>
<div id="req_result"></div>
<?php echo CHtml::endForm(); ?>
</div><hr>
<h3><?php echo CHtml::link('กลับ..หน้าหลัก',array('index'));></h3>
```

### สร้างไฟล์ Views (edit.php)

คำสั่ง : PHP

ไดเรกทอรี่ : [skeleton]/protected/modules/wiki/views/default/edit.php

```
<h2>แก้ไข ID :: #<?php echo CHtml::encode($model->id); ?></h2>
<?php echo CHtml::beginForm();?>
    <?php echo CHtml::activeTextArea($model, 'text',
        array('cols'=>100,'rows'=>10,'style'=>'width:98%;')) ?><hr>
    <?php echo CHtml::submitButton('บันทึกข้อมูล'); ?>
<?php echo CHtml::endForm(); ?>
```

### แก้ไขไฟล์ Views (index.php)

คำสั่ง : PHP

ไดเรกทอรี่ : [skeleton]/protected/modules/wiki/views/default/index.php

```
echo '<hr><h4>'.CHtml::link('เพิ่มข้อมูล',array('create')).'</h4>';
$this->widget('zii.widgets.grid.CGridView', array(
    'id' => 'grid_1', // ชื่อตาราง
    'dataProvider' => $dataProvider, // ตัวแปร data ที่มีข้อมูล
    'enablePagination' => true, // กำหนดให้แสดงปุ่มเบลี่ยนหน้า
    'columns' => array(), // กำหนด column ที่จะแสดง
));
```

ทดสอบ : เข้าไปที่ URL [http://localhost/tutorial\\_yii/wiki/](http://localhost/tutorial_yii/wiki/)

ทดสอบ : กด “เพิ่มข้อมูล” || เข้าไปที่ [http://localhost/tutorial\\_yii/wiki/default/create](http://localhost/tutorial_yii/wiki/default/create)

สร้างข้อมูลใหม่

Text  
ข้อมูลใหม่

[กลับ..หน้าหลัก](#)

ทดสอบ : หลักกรอกข้อมูลเสร็จ ให้กดปุ่ม “บันทึกข้อมูล” หากเพิ่มข้อมูลสำเร็จ จะแสดงผลลัพธ์

สร้างข้อมูลใหม่

Text  
ข้อมูลใหม่

[เพิ่มข้อมูลแล้ว :: ข้อมูลใหม่  
แก้ไข]

[กลับ..หน้าหลัก](#)

ทดสอบ : จากนั้นให้เราคลิกที่ลิงค์คำว่า “แก้ไข” เพื่อทดสอบแก้ไขข้อมูล

แก้ไข ID :: #1

ข้อมูลใหม่ - แก้ไข

ทดสอบ : แก้ไขข้อมูล “สำเร็จ” ระบบจะแสดงข้อมูล “อัปเดต” ค้างรูปภาพ

ID :: #1 [แก้ไข](#) | [ลบ](#)

เนื้อหา :: ข้อมูลใหม่ - แก้ไข

## เพิ่ม,ลบ,แก้ไขไฟล์รูปภาพ ด้วยฟังก์ชัน CUploadFile

บทนี้ เราจะมาเรียนรู้ “วิธีการอัพโหลดไฟล์รูปภาพใหม่”, “การลบไฟล์รูปภาพเดิมและอัพโหลดไฟล์รูปภาพใหม่แทน” ด้วยฟังก์ชันที่ Yii Framework สร้างไว้ให้เราได้เลือกใช้งาน คือ “ฟังก์ชัน CUploadFile” และ Yii ฟังก์ชัน saveAd(); มาเริ่มต้นกันเลยครับ

**สร้างฐานข้อมูล :** เพื่อใช้ทดสอบ “การอัพโหลดไฟล์” || ดาวน์โหลดที่ [http://goo.gl/4BL9tv \(.SQL\)](http://goo.gl/4BL9tv)

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	<b>id</b>	int(5)			No	None		
2	<b>name</b>	varchar(50)	utf8_unicode_ci		No	None		
3	<b>picture</b>	varchar(50)	utf8_unicode_ci		No	None		

**สร้างโมเดล :** ชื่อว่า “Picture” จากฐานข้อมูลด้วย Gii Model Generator System

**Model Generator**

This generator generates a model class for the specified database table.

Fields with \* are required. Click on the highlighted fields to edit them.

Database Connection \*

db

Table Prefix

[empty]

Table Name \*

picture

Model Class \*

Picture

Base Class \*

CActiveRecord

**สร้าง CRUD :** ชื่อ Picture ด้วย Gii CRUD Generator System

**Crud Generator**

This generator generates a controller and views that implement CRUD operations for the specified data model.

Fields with \* are required. Click on the highlighted fields to edit them.

Model Class \*

Picture

Controller ID \*

picture

Base Controller Class \*

Controller

Code Template \*

default (D:\xampp\htdocs\yii\framework\generators\crud\templates\default)

**Preview**

ແກ້ໄຂໂຟຣົມ : ເພື່ອໃຊ້ສໍາຮັບ “ອັພໂຫລດໄຟລ໌” ຮູບປາພ

ຄໍາສັ່ງ : PHP

ໄດ້ຮັບອ່ານຸ່າງ : [skeleton]/protected/views/picture/\_form.php

```
<?php
$form = $this->beginWidget('CActiveForm', array(
    'id'=>'picture-form',
    'enableAjaxValidation'=>false,
    'htmlOptions'=>array('enctype'=>'multipart/form-data'), /* ເພີ່ມເຕີມຄໍາສັ່ງລົງໄປ */
));
?>
.....
<?php echo $form->labelEx($model,'picture'); ?>
<?php echo $form->fileField($model,'picture'); ?> <!-- ປັບປຸງ -->
<?php echo $form->error($model,'picture'); ?>
.....
```

The screenshot shows a 'Create Picture' form. It includes fields for 'ID' (marked with a red asterisk), 'Name' (marked with a red asterisk), and 'Picture' (marked with a red asterisk). The 'Picture' field contains a file input button labeled 'ເລືອກໄຟລ໌' (Select File) and a placeholder text 'ໄມ່ໄດ້ເລືອກໄຟລ໌ໄດ້' (No file selected).

ຄໍາສັ່ງ : ໃຫ້ແກ້ໄຂ “PictureController.php” ໃຫ້ສາມາດຮັບ “ອັພແຫລດໄຟລ໌ຮູບປາພ” ໄດ້

ຄໍາສັ່ງ : PHP

ໄດ້ຮັບອ່ານຸ່າງ : [skeleton]/protected/controllers/PictureController.php

```
public function actionCreate(){
    $model = new Picture;
    if(isset($_POST['Picture'])){
        $model->attributes=$_POST['Picture'];
        // $model->picture :: picture ດີວ່າມີ “ຝຶກຕົກທີ່ໃຫ້ເກີນຂໍ້ໄຟລ໌ຮູບປາພ”
        $model->picture = CUploadedFile::getInstance($model, 'picture'); /* ເພີ່ມເຕີມ */
        if($model->save()){
            $path = Yii::app()->basePath.'..../images/'.$model->picture; /* ເພີ່ມເຕີມ */
            $model->picture->saveAs($path); /* ເພີ່ມເຕີມ */
            $this->redirect(array('view','id'=>$model->id));
        }
        $this->render('create',array('model'=>$model));
    }
}
```

คำสั่ง : ให้แก้ไข “PictureController.php” ให้สามารถ “อัพโหลดไฟล์รูปภาพ” ได้

คำสั่ง : PHP

ไฟล์เดิม : [skeleton]/protected/controllers/PictureController.php

```
public function actionUpdate($id){
    $model = $this->loadModel($id);
    if(isset($_POST['Picture'])){
        $oldfilename = $model->picture; /* เพิ่มเติม */
        $model->setAttributes($_POST['Picture']); /* แก้ไขเป็น */
        $file = CUploadedFile::getInstance($model,'picture'); /* เพิ่มเติม */
        if (is_object($file) && get_class($file) === 'CUploadedFile') { /* เพิ่มเติม */
            $model->picture = $file; /* เพิ่มเติม */
        }else{
            $model->picture = $oldfilename; /* เพิ่มเติม */
        }
        if($model->save()){
            if (is_object($file) && get_class($file) === 'CUploadedFile') { /* เพิ่มเติม */
                $path = Yii::app()->basePath '/../images/'.$model->picture; /* เพิ่มเติม */
                $model->picture->saveAs($path); /* เพิ่มเติม */
                if ($oldfilename != $model->picture->name) { /* เพิ่มเติม */
                    $path = Yii::app()->basePath '/../images/'.$oldfilename; /* เพิ่มเติม */
                    if(file_exists($path)){ /* เพิ่มเติม */
                        unlink($path); // ลบไฟล์รูปภาพเดิมทิ้ง
                    }
                }
            }
            $this->redirect(array('view','id'=>$model->id));
        }
    }
    $this->render('update',array(
        'model'=>$model,
    ));
}
```

ทดสอบ : อัพโหลดไฟล์รูปภาพใหม่ || URL [http://localhost/tutorial\\_yii/picture/create](http://localhost/tutorial_yii/picture/create)

Create Picture

Fields with \* are required.

ID \*

Name \*

Picture \*

## รู้ก่อน ป้องกันก่อน ขั้นตอนการอัพโหลดไฟล์ขึ้น Hosting

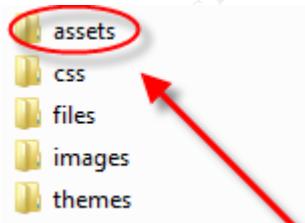
สวัสดีครับ ขอต้อนรับเข้าสู่ห้องเรียนรู้ “เคล็ดลับและเทคนิค” เกี่ยวกับ Yii PHP Framework เราจะไม่พูดอะไรมาเริ่มดันกันเลย ด้วย “เคล็ดลับ” แรกก็คือการ กำหนดค่าหรือการตั้งค่าต่าง ๆ ในขั้นตอนการนำไฟล์เว็บแอพพลิเคชัน อัพโหลดขึ้นไปยัง Server หรือ Web Hosting ครับ ผມเห็นสมาชิกหลายคนท่านสอบถามบ่อยมากว่า “ทำไมเว็บไม่ทำงาน ทั้งๆ ที่อัพโหลดไฟล์ขึ้น server ทั้งหมดแล้ว”

ก่อนอื่นเราต้องทำความเข้าใจก่อนนะว่า Server (localhost) กับ Server (Hosting) มันต่างกันยังไง เริ่มต้นด้วย Server (localhost) server ประเภทนี้ มีการติดตั้งลงในคอมพิวเตอร์ ส่วนบุคคลของเรา ซึ่งมันก็จะหมายถึง เรายังสิทธิ์ในการเข้าใช้ข้อมูลต่าง ๆ และสามารถจัดการ เพิ่ม, ลบ, แก้ไข ข้อมูล/ชื่อไฟล์เดอร์/ชื่อไฟล์ เป็นต้น แต่สำหรับ Server (Hosting) ประเภทนี้ เราได้นำไฟล์และไฟล์เดอร์ต่าง ๆ อัพโหลด ฝากไว้กับ Server ของบุคคลอื่น ซึ่งต้องมีการกำหนดสิทธิ์การเข้าใช้งานข้อมูล ซึ่งถ้า Web Hosting ประเภทแบ่งทรัพยากรให้ร่วมกันแล้ว ละก็ ต้องมีการกำหนดสิทธิ์ การเข้าถึงข้อมูลได้เฉพาะสมาชิก เพื่อป้องกันการเข้าไปคุหรือข้อมูลต่าง ๆ ของบุคคลอื่น ๆ ได้

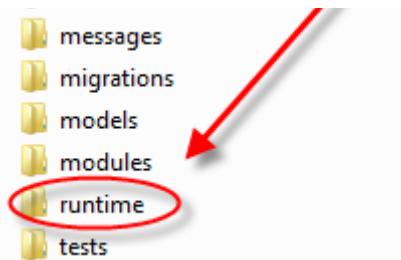
สำหรับน้องๆ ที่สนใจนี่ แนะนำสอนให้สมาชิก รู้วิธีการแบ่งสิทธิ์การเข้าถึงข้อมูล ทั้งสิทธิ์เข้าถึงไฟล์ และสิทธิ์การเข้าถึงไฟล์เดอร์ (ส่วนใหญ่จะกำหนดแค่สิทธิ์ในการเข้าถึงไฟล์เดอร์ เท่านั้น) ขั้นตอนก็ไม่มีอะไรมาก “เรามาลงมือปฏิบัติ” กันเลยดีกว่า...

### ไฟล์เดอร์เบื้องต้นที่ต้อง “กำหนดสิทธิ์” การเข้าถึงข้อมูลภายใน

- ไฟล์เดอร์ assets เก็บไฟล์ CSS, JavaScript, image, font เป็นต้น ของ Extensions, Yii ต่าง ๆ



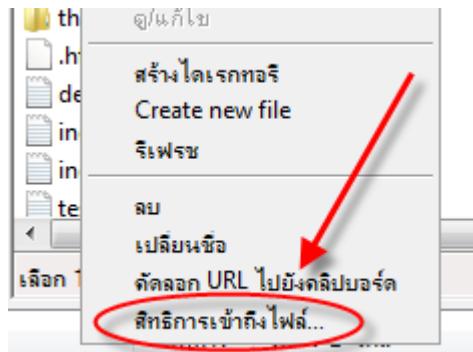
- ไฟล์เดอร์ runtime เก็บไฟล์ข้อมูล log ของระบบ Yii ทั้งหมด



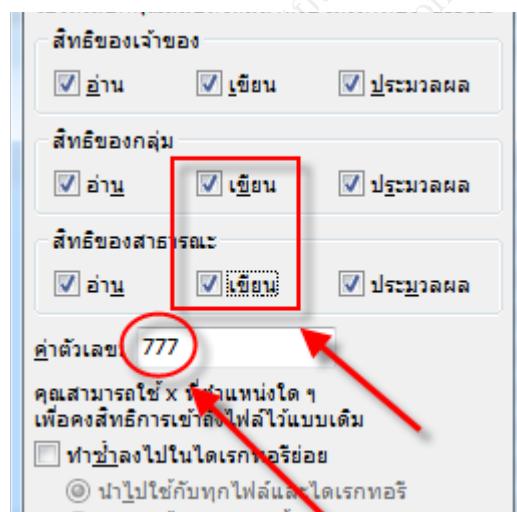
### ขั้นตอนการ “กำหนดสิทธิ์” การเข้าถึงข้อมูลภายใน

โปรแกรมที่ผมใช้อัพโหลดไฟล์ขึ้น Server ในตัวอย่างใช้โปรแกรม FileZilla FTP Client v.3.7.3

- คลิก “ขวา” ที่ไฟล์เดอร์ที่เราต้องการกำหนดสิทธิ์การเข้าถึงข้อมูล (ตามรูปภาพ)



- ติก “เครื่องหมายถูก” ให้ ครบทั้งหมด ดัง รูปภาพนี้ (สังเกต ค่าตัวเลขจาก 755 มาเป็น 777)



- กำหนดสิทธิ์ ให้ครบ ทั้ง 2 ไฟล์เดอร์ (assets, runtime) หรือไฟล์เดอร์ที่คุณต้องการอื่น ๆ ได้.
- เพียงเท่านี้ก็สามารถทำให้ “เว็บแอ��พพลิเคชัน” ของเราทำงานได้บน Server จริง ๆ แล้วครับ..

## สิ่งที่ต้องทำหลังการอัพโหลดไฟล์ขึ้น Server เรียบร้อยแล้ว

ต้องบอกก่อนนะว่า หลังการอัพโหลดไฟล์ขึ้นสู่ Server (Hosting) เรียบร้อยแล้ว นั้น ผมอยากจะแนะนำ เพื่อความปลอดภัยของเว็บแอปพลิเคชัน เพิ่มขึ้นอีก 1 ระดับ คือ เราไม่ควรนำไฟล์เดอร์ framework เก็บไว้ในไฟล์เดอร์เดียวกันกับเว็บของเรา ให้เราแยกไฟล์เดอร์ framework ออกมายไว้ด้านนอก อีก 1 ระดับครับ ดังรูปภาพ ดัวอย่าง ดังนี้ครับ



## การพัฒนาเว็บแอปพลิเคชัน ด้วย Yii Framework

### บทที่ 9 การใช้งาน Zii Widget

Widget คืออีกส่วนหนึ่งที่เพิ่มช่องทางในการออกแบบเว็บไซต์ให้มีความหลากหลาย (Extension) แบบหนึ่ง ข้อดีของ “วิดเจ็ต” คือ สามารถใช้ได้กับระบบเว็บแอปพลิเคชันขนาดใหญ่ ๆ ที่มีองค์ประกอบมากมาย

“วิดเจ็ต” เป็นอีกส่วนหนึ่งที่เพิ่มช่องทางในการออกแบบเว็บไซต์ให้มีความหลากหลาย โดยสามารถผนวกทรัพยากรพากศรีปและรูปภาพเอาไว้ใน “วิดเจ็ต” ได้เลย (ใน Yii เรียกทรัพยากรพากนี้ว่า asset ครับ)

ในบทนี้เราจะมาเรียนรู้วิธีการใช้งาน Widget ต่าง ๆ ที่สำคัญและจำเป็นต่อเว็บแอปพลิเคชันของเรา โดย Yii ได้จัดไว้ให้เราได้เรียกใช้ได้เลย ไม่ต้องไปดาวน์โหลดเข้ามาติดตั้งเพิ่มเติมอีก

#### ตัวอย่าง : Widget CBreadcrumbs

```
$this->widget('zii.widgets.CBreadcrumbs', array(  
    'links'=>array(  
        'Widget'=>array('widget/view', 'id'=>12),  
        'CBreadcrumbs',  
    ),  
));
```

[Home](#) » [Widget](#) » [CBreadcrumbs](#)

#### ตัวอย่าง : Widget CDetailView

```
$model = Profile::model()->findByPk(1);  
$this->widget('zii.widgets.CDetailView', array(  
    'data' => $model,  
    'attributes'=>array(  
        'firstname','lastname', // ข้อมูลที่ต้องการแสดง (field)  
    ),  
));
```

First Name	Administrator
Last Name	Admin

### ตัวอย่าง : Widget CMenu

```
$css = '#my-menu ul{list-style-type:none; border: 1px #000 solid;}  
#my-menu li {float:left; display:block; margin-right:20px; padding:8px;}  
#my-menu li.active a{color:#f00}; // คำสั่ง CSS ตกแต่ง เมนู  
Yii::app()->clientScript->registerCss('css-place-1',$css); // แทรกคำสั่ง CSS  
  
$this->widget('zii.widgets.CMenu', array(  
    'id'=>'my-menu',  
    'items'=>array(  
        array('label'=>'หน้าหลัก', 'url'=>array('site/index'), 'active'=>true),  
        array('label'=>'สินค้า', 'url'=>array('site/index')),  
        array('label'=>'ติดต่อเรา', 'url'=>array('site/index')),  
    ),  
));
```



### ตัวอย่าง : Widget CGridView

```
$dataProvider = new CActiveDataProvider('Profile', array(  
    'pagination' => array(  
        'pageSize' => 10 // กำหนดให้หน้านี้แสดง 10 รายการ  
    ),  
));  
$this->widget('zii.widgets.grid.CGridView', array(  
    'dataProvider'=>$dataProvider,  
    'columns'=>array(  
        'user_id',  
        'firstname',  
        'lastname',  
    ),  
));
```

Displaying 1-3 of 1 result.			
User ID	First Name	Last Name	
1	Administrator	Admin	
2	Demo	Demo	
3	Test	Test	

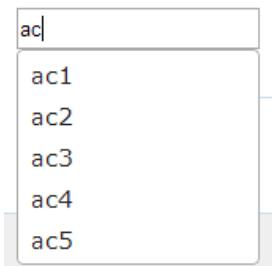
### ตัวอย่าง : Widget CJuiAccordion

```
$this->widget('zii.widgets.jui.CJuiAccordion',array(
    'panels'=>array(
        'panel 1'=>'content for panel 1',
        'panel 2'=>'content for panel 2',
        'panel 3'=>'content for panel 3',
    ),
    'options'=>array(
        'animated'=>'bounceslide',
        'style'=>array('minHeight'=>'100'),
    ),
));
```



### ตัวอย่าง : Widget CJuiAutoComplete

```
// data
$datalist = array('ac1','ac2','ac3', 'ac4', 'ac5');
$this->widget('zii.widgets.jui.CJuiAutoComplete', array(
    'name'=>'normal',
    'source'=>$datalist, //source จะเป็น Array
    'options'=>array('minLength'=>'2'),
    'htmlOptions'=>array(
        'style'=>'height:20px;width:150px;',
    ),
));
```



\*\*\* ทดสอบ กรอกชื่อ muc คำว่า “ac” และจะมีรายการขึ้นมาให้เราเลือก ดังรูป

ตัวอย่าง : Widget CJuiButton || ใช้งานร่วมกับ CActiveForm

```
if(isset($_POST['user'])){  
    echo 'User Name :: '.$_POST['user'].'<br /><br />';  
}  
echo 'User Name : <input type="text" value="" id="user" name="user" />';  
$this->widget('zii.widgets.jui.CJuiButton',array(  
    'name'=>'btnSubmit',  
    'buttonType'=>'submit',  
    'caption'=>'บันทึกข้อมูล',  
    'htmlOptions'=>array(  
        'style'=>'background:#0064cd;color:#ffffff;',  
    ),  
));
```

User Name :  บันทึกข้อมูล

```
$this->widget('zii.widgets.jui.CJuiButton',array(  
    'name'=>'ratio-button',  
    'buttonType'=>'radio',  
    'caption'=>'Single Radio',  
));
```

Single Radio

```
$this->widget('zii.widgets.jui.CJuiButton',array(  
    'name'=>'checkbox-button',  
    'buttonType'=>'checkbox',  
    'caption'=>'Single Checkbox',  
));
```

Single Checkbox

```
$this->widget('zii.widgets.jui.CJuiButton',array(  
    'name'=>'link',  
    'caption'=>'Link To drivesoftcenter.net',  
    'buttonType'=>'link',  
    'url'=>'http://drivesoftcenter.net',  
));
```

Link To drivesoftcenter.net

### ตัวอย่าง : Widget CJuiDatePicker

```
$this->widget('zii.widgets.jui.CJuiDatePicker',array(
    'name'=>'datepicker-showButtonPanel',
    'value'=>date('d-m-Y'),
    'options'=>array(
        'showAnim'=>'slide', //'slide', 'fold', 'slideDown', 'fadeIn', 'blind', 'bounce', 'clip', 'drop'
        'showButtonPanel'=>true,
    ),
    'htmlOptions'=>array('style'=>"),
));
});
```



### ตัวอย่าง : Widget CJuiDialog

```
$this->beginWidget('zii.widgets.jui.CJuiDialog',array(
    'id'=>'mydialog',
    'options'=>array( 'title'=>'Dialog box','autoOpen'=>false),
));
echo 'dialog content here';
$this->endWidget('zii.widgets.jui.CJuiDialog');
echo CHtml::link('Open Dialog', '#', array(
    'onclick'=>$('#mydialog').dialog("open");
    'return false;',
));
});
```



### ตัวอย่าง : Widget CJuiProgressBar

```
$this->widget('zii.widgets.jui.CJuiProgressBar', array(  
    'value'=>80, //value in percent  
    'htmlOptions'=>array(  
        'style'=>'height:20px;'  
    ),  
));
```



### ตัวอย่าง : Widget CJuiSlider

```
$this->widget('zii.widgets.jui.CJuiSlider', array(  
    'value'=>37,  
    'id'=>'amtSlider',  
    'options'=>array(  
        'min'=>0, 'max'=>100,  
        'slide'=>'js:function(event, ui) { $("#amt").val(ui.value);}',  
    ),  
    'htmlOptions'=>array( 'style'=>'height:12px;',  
));
```



### ตัวอย่าง : Widget CJuiSliderInput

```
$this->widget('zii.widgets.jui.CJuiSliderInput', array(  
    'name'=>'slider_basic',  
    'value'=>50, 'event'=>'change',  
    'options'=>array(  
        'min'=>0, 'max'=>100,  
        'slide'=>'js:function(event,ui){$("#amount_basic").val(ui.value);}',  
    ),  
    'htmlOptions'=>array('style'=>'width:200px;',  
));
```

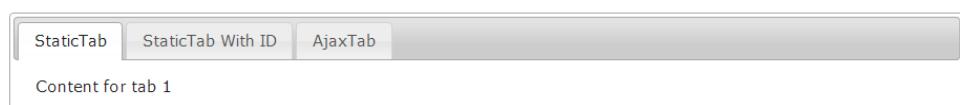


### ตัวอย่าง : Widget CJuiTabs

```
$this->widget('zii.widgets.jui.CJuiTabs', array(
    'tabs'=>array(
        'StaticTab 1'=>'Content for tab 1',
        'StaticTab 2'=>array(
            'content'=>'Content for tab 2',
            'id'=>'tab2'
        ),
    ),
    'options'=>array(
        'collapsible'=>true
    ),
));
});
```



```
$this->widget('zii.widgets.jui.CJuiTabs',array(
    'tabs'=>array(
        'StaticTab '=>'Content for tab 1',
        'StaticTab With ID'=>array('content'=>'Content for tab 2 With Id' , 'id'=>'tab2'),
        // panel 3 contains the content rendered by a partial view
        'AjaxTab'=>array('ajax'=>$this->createUrl('ajax')),
    ),
    // additional javascript options for the tabs plugin
    'options'=>array(
        'collapsible'=>true,
    ),
    'id'=>'MyTab-Menu',
));
});
```





fenics128[at]gmail.com

# การพัฒนาเว็บแอ��泲ิเคชัน ด้วย Yii Framework

## บทที่ 10 ตัวอย่างกรณีศึกษาระบบซื้อขายสินค้าออนไลน์พื้นฐาน

ระบบซื้อขายสินค้า ผ่านทางเว็บไซต์ หรือ e-Commerce ขึ้นพื้นฐาน คำว่า “พื้นฐาน” เนื่องจากระบบซื้อขายสินค้า ตามแบบจริง เป็นระบบที่ใหญ่มาก ๆ แต่สำหรับเนื้อหาต่อไปนี้ จะเป็นขั้นพื้นฐานที่ระบบซื้อขายสินค้า ต้องมีอยู่ในตัวระบบ ซึ่งจะประกอบไปด้วย ระบบจัดการข้อมูลสินค้าและข้อมูลสมาชิก ระบบสั่งซื้อสินค้า ระบบจัดซื้อสินค้า ระบบลดต้นทุนสินค้า และระบบจัดพิมพ์รายงาน

### อธิบายขั้นตอนการทำงานของระบบ

ก่อนเริ่มการทำงานของระบบ ในฐานข้อมูลต้องมีข้อมูลรายการสินค้าอยู่ก่อน ซึ่งขั้นตอนแรก เราต้องเตรียมข้อมูลรายการสินค้าไว้ให้พร้อม ตัวอย่างประมาณ 10 ชิ้น ข้อมูลต่อไปที่ต้องมีก็คือข้อมูลสมาชิก อาจจะได้จากการสมัครสมาชิกหรือเราจำลองตัวอย่างสมาชิกมาประมาณ 3 คน เมื่อเราเตรียมข้อมูลพื้นฐาน ทั้ง 2 ข้อมูลเสร็จสิ้น เรายังมาเริ่มกระบวนการสร้างสั่งซื้อสินค้า โดยการสั่งซื้อสินค้าได้นั้น ระบบต้องให้สามารถเลือกอินเทอร์เฟซระบบ เสิร์ฟเวอร์อย่างก่อน ถึงจะสามารถเข้าใช้งานระบบต่าง ๆ ได้ หลังจากเลือกอินเทอร์เฟซ สมาชิกถึงจะสามารถเข้าไปใช้งานระบบ การจัดการข้อมูลสมาชิก เช่น แก้ไขข้อมูลส่วนตัว เปลี่ยนรหัสผ่าน และระบบสั่งซื้อสินค้า

กระบวนการจัดการข้อมูลสินค้า อนุญาตให้เฉพาะผู้ดูแลระบบเท่านั้น ที่สามารถจัดการข้อมูลลูกค้าได้ ดังนี้ เพิ่มข้อมูลสินค้า แก้ไขข้อมูล ลบข้อมูล บันทึกข้อมูล และค้นหาข้อมูลสินค้า

กระบวนการทำงานของระบบสั่งซื้อสินค้า มีขั้นตอนดังนี้ 1) สมาชิกเลือกสินค้าที่ต้องการลง ตะกร้าสินค้า 2) ตรวจสอบรายการสินค้าในตะกร้าสินค้าให้เรียบร้อย 3) ยืนยันการสั่งซื้อสินค้า 4) สมาชิกแจ้งการชำระเงิน 6) รออีเมลยืนยันการจัดส่งสินค้า

### จัดเตรียมเครื่องมือก่อนเริ่มพัฒนาระบบ

จัดเตรียมเครื่องมือที่ใช้ในการพัฒนาเว็บแอ��泲ิเคชัน แต่บางท่านอาจจะสามารถใช้เครื่องมืออื่น ๆ นอกเหนือจากนี้ก็สามารถทำได้ จากตัวอย่าง เครื่องมือที่ใช้มีดังนี้

1. NetBeans 8.0 HTML5 & PHP Installer for Windows/English (EN)
2. XAMPP Win32 v.1.8.3 (Apache 2.4.4,MySQL 5.5.32,PHP 5.4.19,phpMyAdmin 4.0.4)
3. Yii Framework v1.1.15 released on Jun 29, 2014

สร้างเว็บแอปพลิเคชัน : เขียนคำสั่ง yiic command เพื่อสร้างเว็บแอปฯ ดังรูปภาพด้วยคำสั่ง : yiic webapp E:\xampp\htdocs\weshopping

```
C:\Windows\system32\cmd.exe - yiic webapp E:\xampp\htdocs\weshopping
E:\xampp\htdocs\yii\framework>yiic webapp E:\xampp\htdocs\weshopping
Create a Web application under 'E:\xampp\htdocs\weshopping'? <yes|no> [no]:
```

ตั้งค่าระบบ : กำหนดชื่อเว็บแอปฯ ตั้งค่าฐานข้อมูล และเปิดระบบ Gii Code Generate

คำสั่ง : PHP

ไฟล์ config/main.php

```
return array(
    'basePath'=>dirname(__FILE__).DIRECTORY_SEPARATOR..,
    'name'=>'WeShopping', // กำหนดชื่อเว็บแอปฯ
    'language' => 'th', // ตั้งภาษาเริ่มต้น ภาษาไทย
    'timeZone'=>'Asia/Bangkok', // ตั้งค่า TimeZone
    'theme'=>'bootstrap',
    'preload' => array('log','booster'), // เพิ่ม booster ลงใน
    'import'=>array(
        .....
        'application.modules.user.models.*', // นำเข้า Yii-User Extentions
        'application.modules.user.components.*', // นำเข้า Yii-User Extentions
        'application.modules.admin.models.*', // นำเข้า admin Modules
    ),
    'modules'=>array(
        'gii'=>array(
            'class'=>'system.gii.GiiModule',
            'password'=>'12345', // กำหนดรหัสผ่าน Gii
            'ipFilters'=>array('127.0.0.1 '::1'),
            'generatorPaths'=>array('booster.gii'), // เพิ่มเดิม booster.gii
        ),
        'user'=>array(
            'tableUsers'=>'tbl_users', // ชื่อฐานข้อมูลเก็บข้อมูล users
            'tableProfiles'=>'tbl_profiles', // ชื่อฐานข้อมูลเก็บข้อมูล profiles
            'tableProfileFields'=>'tbl_profiles_fields', // ชื่อฐานข้อมูลเก็บข้อมูล Fields profiles
        ),
        'backoffice'=>array(), // เพิ่ม Modules backoffice (สร้างเอง)
    ),
    'aliases'=>array(
        'booster'=>'application.extensions.booster',
    ),
);
```

```

'components'=>array(
    'user'=>array(
        'loginUrl'=>array('/user/login'),
        'allowAutoLogin'=>true
    ),
    'urlManager'=>array(
        .....
        'rules'=>array(
            'gii'=>'gii', // เพิ่มเดิมคำสั่งนี้
            'gii/<controller:\w+>'=>'gii/<controller>', // เพิ่มเดิมคำสั่งนี้
            'gii/<controller:\w+>/<action:\w+>'=>'gii/<controller>/<action>', // เพิ่มเดิมคำสั่งนี้
            '<controller:\w+>/<id:\d+>'=>'<controller>/view',
            '<controller:\w+>/<action:\w+>/<id:\d+>'=>'<controller>/<action>',
            '<controller:\w+>/<action:\w+>'=>'<controller>/<action>',
        ),
        'showScriptName'=>false, // เพิ่มคำสั่งนี้
    ),
    'db'=>array(
        'connectionString' => 'mysql:host=localhost;dbname=db_weshopping', // ใช้ MySQL
        'emulatePrepare' => true,
        'charset' => 'utf8',
        'username' => 'root', // username เข้าใช้ฐานข้อมูล
        'password' => 'password', // password เข้าใช้ฐานข้อมูล
    ),
    'booster' => array( // เพิ่ม เพื่อໄwake เรียกใช้งาน Widget Booster
        'class' => 'booster.components.Booster',
    ),
    // สร้าง Class ใช้งานเอง
    'myclass' => array(
        'class'=>'Myclass'
    ),
),
);

```

## ສ້າງໄຟ້ : Shopping (Shopping.php)

កំសៀង : PHP

ไฟร์เรคทอรี่ : [skeleton]/protected/components/Shopping.php

```
class Shopping extends CApplicationComponent {  
    /* คำสั่งจะเขียนเพิ่มเติมทีหลัง */  
}
```

## ดาวโหลด Yii Booster version 4.0.1

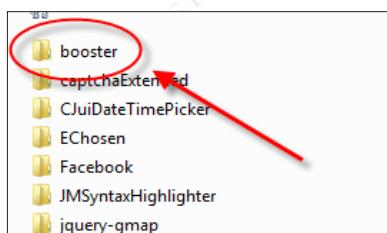
Extension เกี่ยวกับการแสดงผล UI ให้กับผู้ใช้งาน ด้วยการรวมฟังก์ชั่น รูปแบบการแสดงของ Twitter Bootstrap v.3.1.1 อาทิเช่น การวางแผนกราฟ เว็บ ให้รองรับการแสดงผลหลากหลายอุปกรณ์ มือถือ คอมพิวเตอร์ เป็นต้น



ติดตาม ภาคผนวก คู่มือการใช้งาน Yii Booster v.4.0.1 (ท้ายเล่ม)

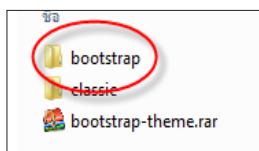
## ดาวโหลด Yii Booster v.4.0.1 || ดาวโหลดที่นี่ <http://goo.gl/wL098g>

แตกไฟล์ yiibooster-4.0.1.zip และแก้ไขชื่อไฟล์เดอร์เป็น “booster” ไว้ในไฟล์เดอร์ extensions ตรง ไดเรกทอรี่ [skeleton]/protected/extensions/



## ดาวโหลด Twitter Bootstrap Theme v.3.1.1 || ดาวโหลดที่นี่ <http://goo.gl/qAi8QI>

แตกไฟล์ bootstrap-theme-weshopping.rar และเปลี่ยนชื่อไฟล์เดอร์เป็น “bootstrap” ไว้ในไฟล์เดอร์ themes ตรง ไดเรกทอรี่ [skeleton]/themes/



## ดาวโหลด SEO Extension v.0.9.2 || ดาวโหลดที่นี่ <http://goo.gl/26T2ba>

Extension เกี่ยวกับการสร้างเว็บให้สนับสนุนการทำงานของ Google Bot หรือการทำ SEO การไปรับโภตเว็บไซต์ให้ดีดีหน้าหนึ่งเรียงลำดับการค้นหาของ Google Bot ที่ใช้ KeyWord ของเราที่ได้กำหนดไว้

### ติดตั้ง SEO Extension v.0.9.2

หลังจากดาวโหลดไฟล์แล้ว เราจะได้ไฟล์ชื่อว่า seo-0.9.2c.zip ให้ทำการแตกไฟล์ไว้ในโฟล์เดอร์ extensions โดยสร้างโฟล์เดอร์ใหม่ขึ้นมาชื่อว่า “seo” และนำไฟล์ที่แตกได้นำไปเก็บไว้ในโฟล์เดอร์นี้

ต่อไปเข้าไปยังไฟล์ Layouts หลักของเรา ในที่นี่ให้เราเข้าไปแก้ไขไฟล์ชื่อว่า “main.php” ที่อยู่ในไฟล์เดอร์ [skeleton]\themes\bootstrap\views\layouts\main.php และเพิ่มคำสั่งดังต่อไปนี้ ลงในส่วนของ Tag <head> ... </head>

คำสั่ง : PHP

ไฟล์ที่ต้องแก้ไข : [skeleton]\themes\bootstrap\views\layouts\main.php

```
<?php  
// เรียกใช้งาน SEO Extension  
Yii::app()->controller->widget('ext.seo.widgets.SeoHead',array(  
    'httpEquivs'=>array(  
        'Content-Type'=>'text/html; charset=utf-8',  
        'Content-Language'=>'en-US',  
    ),  
    'defaultDescription'=>'We Shopping ระบบร้านค้าออนไลน์', // กำหนด Description เมื่องต้น  
    'defaultKeywords'=>'shopping,yii framework,drivesoftcenter', // กำหนด Keywords เมื่องต้น  
));  
?  
</head>
```

ขัดเตรียมฐานข้อมูล : ชื่อฐานข้อมูล “db\_weshopping” || ดาวโหลดได้ที่นี่ <http://goo.gl/oecIwF>

สร้างโมเดลข้อมูล (สร้างโมเดลด้วย Gii System) || แนะนำดาวโหลดที่นี่ <http://goo.gl/bxhzh8>

1. โมเดลข้อมูลสินค้า (tbl\_product)
2. โมเดลข้อมูลการสั่งซื้อ (tbl\_order)
3. โมเดลข้อมูลรายละเอียดการสั่งซื้อ (tbl\_orderdetail)
4. โมเดลข้อมูลการติดต่อ (tbl\_contact)
5. โมเดลข้อมูลการชำระเงิน (tbl\_payment)

\*\*\* ไฟล์โมเดล ข้อมูลผู้ใช้งาน ข้อมูลพิเศษข้อมูลผู้ใช้ และข้อมูลไปรษณีย์ไฟล์ มีอยู่ใน Yii-User Extension อยู่แล้ว ไม่ต้องสร้าง นะครับ

### ติดตั้ง Yii User Extension v.0.3.61 || ดาวโหลดที่นี่ <http://goo.gl/pTZySp>

เพื่อจัดการข้อมูลผู้ใช้งาน มีระบบที่ถูกติดตั้งมาพร้อมกับ Yii User v.0.3.61 ดังนี้ ระบบจัดการข้อมูลผู้ใช้งานระบบจัดการข้อมูลฟิล์ตผู้ใช้ ระบบจัดการข้อมูลไฟล์ ระบบสมัครสมาชิก ระบบล็อกอินเข้าสู่ระบบ ระบบลืมรหัสผ่าน ระบบเปลี่ยนรหัสผ่าน, ระบบแสดงข้อมูลผู้ใช้งาน

\*\*\* ศึกษาเพิ่มเติมได้ที่ บทที่ 7 ติดตั้งและใช้งาน Extension สำหรับ Yii framework

### สร้าง Module : ชื่อว่า “backoffice” || ดาวโหลดไฟล์ที่ <http://goo.gl/NooggK>

เพื่อใช้เป็นระบบจัดการหลังร้าน (BackOffice) ให้กับเว็บแอปพลิเคชันของเรา ซึ่งจะประกอบไปด้วย ระบบต่าง ๆ ดังนี้ ระบบจัดการข้อมูลสินค้า ระบบจัดการข้อมูลสั่งซื้อ ระบบจัดการข้อมูลชำระเงิน ระบบจัดการข้อมูลการติดต่อ

\*\*\* สามารถศึกษาเพิ่มเติมได้ที่ บทที่ 8 รวมเทคนิคต่าง ๆ ที่นาสนใจ

### มาเริ่มต้นพัฒนาเว็บแอปพลิเคชัน กันเลย !

หลังการติดตั้ง Yii User Extension สร้างโมดูล Backoffice สร้างคลาส Component สร้างฐานข้อมูล สร้างโมเดลข้อมูลครบทั้งหมด และการตั้งค่าระบบ Config เสร็จเรียบร้อยแล้ว เราต้องมาเรียนรู้วิธีการใช้งานระบบชื่อขายสินค้าออนไลน์กันเลย

ต้องบอกก่อนนะว่า ขอแนะนำให้เปิด Code ตัวอย่าง ประกอบพร้อมกับเรียนรู้ไปด้วย เพื่อความเข้าใจ ระบบการทำงานได้เร็วขึ้น

### แนะนำระบบที่ถูกติดตั้งมาด้วย Yii User Module เรียบร้อยแล้ว

- ระบบสมัครสมาชิก || <http://localhost/ecommerce/user/registration>
- ระบบเขียนข้อความสมัครสมาชิก
- ระบบล็อกอิน || <http://localhost/ecommerce/user/login>
- ระบบปรับปรุงข้อมูลสมาชิก || <http://localhost/ecommerce/user/admin>
- ระบบเปลี่ยนรหัสผ่าน || <http://localhost/ecommerce/user/recovery/recovery>
- ระบบแจ้งลืมรหัสผ่าน || <http://localhost/ecommerce/user/recovery>

รายการระบบค้างบนทั้งหมดนี้ ถูกติดตั้งมาพร้อมให้เราสามารถเรียกใช้งานได้หลังจากเราทำการติดตั้ง Yii User Module สำเร็จแล้ว ซึ่งจะช่วยให้เราประหยัดเวลาในการพัฒนาเว็บแอปฯ ได้มากขึ้น

### ดาวโหลด : รูปภาพสินค้า || ได้ที่นี่ [http://goo.gl/Ajr8aj \(.RAR\)](http://goo.gl/Ajr8aj)

เมื่อดาวโหลดไฟล์รูปภาพสินค้า (ตัวอย่าง) มาเรียบร้อยแล้ว ให้เรานำรูปภาพทั้งหมดไปวางไว้ในไฟล์เครื่อง “product” ในไฟล์เดียวกัน [skeleton]\images\

ปรับปรุง : ไฟล์ “หน้าแรก” ของเว็บแอปพลิเคชั่นของเรา

คำสั่ง : PHP

ไฟล์เดิมที่รี : [skeleton]/themes/bootstrap/views/site/index.php

```
<?php $this->pageTitle = Yii::app()->name; ?>
<div class="row">
    <div class="col-lg-12 col-md-12 col-sm-12">
        <?php $this->beginWidget('booster.widgets.TbJumbotron',array(
            'heading'=>'ยินดีต้อนรับสู่ เว็บชื่อขายสินค้าออนไลน์',
        )); ?>
        <p>บริการทุกระดับ ประทับใจ ..เมื่อท่านเรียกใช้บริการของเรา</p>
        <?php $this->endWidget(); ?>
    </div>
</div>
<div class="row">
    <div class="col-lg-11 col-md-11 col-sm-12">
        <h3>รายการสินค้า</h3>
        <?php $dataProvince = Product::model()->findAll();?>
        <div class="row">
            <?php foreach ($dataProvince as $value): ?>
                <div class="col-lg-3 col-md-3 col-sm-6">
                    <div class="thumbnail">
                        <?php $pathImg = Yii::app()->baseUrl.'./images/product/'.$value->ProductIMG;?>
                        
                        <div class="caption">
                            <h4><?php echo $value->ProductName;?></h4>
                            <p>ราคา: <?php echo number_format($value->SalePrice,2);?> บาท</p><hr />
                            <p class="text-center">
                                <?php $url = Yii::app()->baseUrl.'./orders/product/id/'.$value->ProductID;?>
                                <a href="<?php echo $url;?>" class="btn btn-primary" role="button">สั่งซื้อ</a>
                            </p>
                        </div>
                    </div>
                </div>
            </div>
            <?php endforeach; ?>
        </div>
    </div>
</div>
```

สร้างไฟล์ : ProductController.php

คำสั่ง : PHP

ไดเรกทอรี่ : [skeleton]/protected/controllers/ProductController.php

```
class ProductController extends Controller{
    public function actionIndex(){
        $this->render('index');
    }
}
```

สร้างไฟล์ : Views (index.php) ในไฟล์เดอร์ “product”

คำสั่ง : PHP

ไดเรกทอรี่ : [skeleton]/protected/views/product/index.php

```
<?php
$this->breadcrumbs = array('Product');
?>
<div class="row">
<div class="col-lg-12 col-md-12 col-sm-12">
<h3>รายการสินค้า</h3>
<?php $dataProvince = Product::model()->findAll();?>
<div class="row">
<?php foreach ($dataProvince as $value): ?>
<div class="col-lg-3 col-md-3 col-sm-6">
<div class="thumbnail">
<?php $pathImg = Yii::app()->baseUrl.'/images/product/'.$value->ProductIMG;?>

<div class="caption">
<h4><?php echo $value->ProductName;?></h4>
<p>ราคา: <?php echo number_format($value->SalePrice,2);?> บาท</p><hr />
<p class="text-center">
<?php $url = Yii::app()->baseUrl.'/orders/product/id/'.$value->ProductID;?>
<a href="<?php echo $url;?>" class="btn btn-primary" role="button">สั่งซื้อ</a>
</p>
</div>
</div>
</div>
<?php endforeach; ?>
</div>
</div>
</div>
```

สร้างไฟล์ : PaymentController.php

คำสั่ง : PHP

ไฟล์เดิมที่อยู่ : [skelton]/protected/controllers/PaymentController.php

```
class PaymentController extends Controller {
    public function actionIndex(){
        if(Yii::app()->user->id){
            $model = new Payment();
            if(isset($_POST['Payment'])){
                $model->attributes=$_POST['Payment'];
                $model->UserID = Yii::app()->user->id;
                $model->PaymentDate = date("Y-m-d H:i:s");
                if($model->validate() && $model->save()){
                    Yii::app()->user->setFlash('contact','ขอบคุณ คุณได้แจ้งข้อมูลการชำระเงิน เรียบร้อยแล้ว..');
                    $this->refresh();
                }
            }
            $this->render('index',array('model'=>$model));
        }else{
            $this->render('message',array('message'=>'กรุณาล็อกอินเข้าสู่ระบบก่อนครับ'));
        }
    }
}
```

สร้างไฟล์ : Views (index.php) ในไฟล์เดิมที่ “payment”

คำสั่ง : PHP

ไฟล์เดิมที่อยู่ : [skelton]/protected/views/payment/index.php

```
<?php
$this->breadcrumbs = array('แจ้งชำระเงิน');
?>
<h1>แจ้งชำระเงิน</h1><hr>
<?php if(Yii::app()->user->hasFlash('contact')): ?>
<?php $this->widget('booster.widgets.TbAlert', array(
    'alerts'=>array('contact'),
)); ?>
<?php else: ?>
<?php
$form = $this->beginWidget('booster.widgets.TbActiveForm', array(
    'id'=>'payment-form',
    'type'=>'horizontal',
    'enableClientValidation'=>true,
    'clientOptions'=>array('validateOnSubmit'=>true),
));
?>
```

```

<p class="note">กรอกข้อมูลที่มีเครื่องหมาย <span class="required">*</span> อยู่ให้ครบถ้วน.</p>
<?php echo $form->errorSummary($model); ?>
<?php echo $form->textFieldGroup($model,'OrdersID'); ?>
<?php echo $form->hiddenField($model,'UserID'); ?>
<?php echo $form->textFieldGroup($model,'TotalPrice',array('size'=>60,'maxlength'=>128)); ?>
<?php echo $form->textAreaGroup($model,'PaymentDetail',array('rows'=>6)); ?>
<p class="text-center">
    <?php
        $this->widget('booster.widgets.TbButton',array(
            'buttonType'=>'submit',
            'context'=>'primary',
            'label'=>'บันทึกข้อมูล',
        ));
    ?>
</p>
<?php $this->endWidget(); ?>
<?php endif; ?>

```

**สร้างไฟล์ :** Views (message.php) ในโฟลเดอร์ “payment”

**คำสั่ง :** PHP

**ไดเรกทอรี่ :** [skeleton]/protected/views/payment/message.php

```

<hr><br><br><br>
<center><h2><?php echo $message;?></h2></center>
<br><br><br> <hr>

```

**สร้างไฟล์ :** OrdersController.php

**คำสั่ง :** PHP

**ไดเรกทอรี่ :** [skeleton]/protected/controllers/OrdersController.php

```

class OrdersController extends Controller{
    public function actionIndex(){
        $this->render('index');
    }
    public function actionUser(){
        $dataProvider = new CActiveDataProvider('Order',array(
            'criteria' => array(
                'condition'=>'CustomerID=:CusID',
                'params'=>array(':CusID'=>Yii::app()->user->id,
            )
        ));
        $this->render('order_user',array(
            'dataProvider'=>$dataProvider
        ));
    }
}

```

```
// แสดงรายการสั่งซื้อสินค้า ในตะกร้าสินค้า
public function actionShowdetail(){
    $this->render('detail');
}

// บันทึกข้อมูล ลงตะกร้าสินค้า
public function actionProduct($id=''){
    if(Yii::app()->user->id){
        if(!empty($id)){
            $session = new CHttpSession();
            $session->open();
            $mProduct = Product::model()->findByPk($id);
            $arr = $session['order'];
            if(empty($arr)){
                $val[$mProduct->ProductID] = array(
                    'id' => $mProduct->ProductID,
                    'name' => $mProduct->ProductName,
                    'price' => $mProduct->SalePrice,
                    'amount' => 1
                ); $session['order'] = $val;
            }else{
                $status = 0;
                foreach ($arr as $key => $value) {
                    if($key == $id){
                        $arr[$key]['amount'] = $arr[$key]['amount'] + 1;
                        $session['order'] = $arr;
                        $status = 1;
                    }
                }
                if($status==0){
                    $arr[$mProduct->ProductID] = array(
                        'id' => $mProduct->ProductID,
                        'name' => $mProduct->ProductName,
                        'price' => $mProduct->SalePrice,
                        'amount' => 1
                    ); $session['order'] = $arr;
                }
            }
        }
        $this->redirect(Yii::app()->user->returnUrl);
    }else{
        $this->redirect(array('user/login'));
    }
}
```

```

public function actionDetail(){
    $this->render('detail2');
}

// บันทึกรายการสั่งซื้อสินค้า
public function actionSave(){
    $session = new CHttpSession(); $session->open();
    $order = $session['order'];
    if(!empty($order)){
        $criteria = new CDbCriteria();
        $criteria->select = 'OrdersID';
        $criteria->order = 'OrdersID DESC';
        $criteria->limit = 1;
        $mOrder = Order::model()->find($criteria);
        if(empty($mOrder)){
            $AutoID = 'ORD00001';
        }else{
            $AutoID1 = substr($mOrder->OrdersID,3,5);
            $AutoID2 = $AutoID1 + 1;
            $AutoID3 = str_pad($AutoID2, 5, '0', STR_PAD_LEFT);
            $AutoID = 'ORD' . $AutoID3;
        }
        $total = 0;
        foreach ($order as $value) {
            $model_detail = new Orderdetail();
            $total = $total + ($value['price'] * $value['amount']);
            $arr1 = array(
                'OrdersID' => $AutoID, 'ProductID' => $value['id'],
                'OrderAmount' => $value['amount'], 'ProductPrice' => $value['price'],
                'OrderPriceTotal' => ($value['price'] * $value['amount']),
            );
            $model_detail->attributes = $arr1;
            $model_detail->save();
        }
        $model_order = new Order();
        $arr2 = array(
            'OrdersID' => $AutoID, 'CustomerID' => Yii::app()->user->id,
            'OrderDate' => date("Y-m-d H:i:s"), 'OrderPriceTotal' => $total,
        );
        $model_order->attributes = $arr2;
        $model_order->save();
        unset($session['order']);
    }
    $session->close();
    $this->render('message',array('message'=>'บันทึกข้อมูลเสร็จเรียบร้อยแล้ว.'));
}

```

**สร้างไฟล์ :** Views (detail.php) ในโฟล์เดอร์ “orders”

**คำสั่ง :** PHP

**ไดเรกทอรี่ :** [skelton]/protected/views/orders/detail.php

```
<?php
$this->breadcrumbs = array('รายละเอียดการสั่งซื้อ');
$session = new CHttpSession();
$session->open();
$ordder = $session['order'];

?>
<div class="container">
<div class="row">
<div class="col-lg-12">
<h3>สรุปรายการสินค้าทั้งหมด </h3><hr>
<a href=<?php echo Yii::app()->baseUrl;?>">
    <button class="btn btn-success">กลับไปหน้าหลัก</button></a><br><br>
<table style="width:100%" border="1">
    <thead bgcolor="#FF3">
        <tr>
            <td class="text-center" style="width:8%><strong>ลำดับ</strong></td>
            <td class="text-left" style="width:47%><strong>รายการสินค้า</strong></td>
            <td class="text-right" style="width:15%><strong>จำนวน</strong></td>
            <td class="text-right" style="width:15%><strong>ราคา</strong></td>
            <td class="text-right" style="width:15%><strong>รวม</strong></td>
        </tr>
    </thead>
    <tbody>
        <?php $row = 0; $total = 0; $amount = 0; $price = 0 ;?>
        <?php foreach ($ordder as $value) :?>
            <?php $amount = $amount + $value['amount'];
            $price = $price + $value['price'];
            $total = $total + ($value['price']*$value['amount']);
            $row++; ?>
            <tr>
                <td class="text-center"><?php echo $row;?></td>
                <td class="text-left"><?php echo $value['name'];?></td>
                <td class="text-right"><?php echo number_format($value['amount']);?></td>
                <td class="text-right"><?php echo number_format($value['price']);?>.-</td>
                <td class="text-right">
                    <?php echo number_format($value['price']*$value['amount']);?>.-</td>
            </tr>
        <?php endforeach; ?>
    </tbody>
```

```

<tfoot>
    <tr>
        <td class="text-right" colspan="2">รวม</td>
        <td class="text-right" ><?php echo number_format($amount);?></td>
        <td class="text-right" ><?php echo number_format($price);?>.-</td>
        <td class="text-right" ><?php echo number_format($total);?>.-</td>
    </tr>
</tfoot>
</table><br>
<p class="text-right"><a href="../save">
    <button class="btn btn-primary">สั่งซื้อสินค้า</button></a></p>
</div>
</div>
</div><br><hr>

```

สร้างไฟล์ : Views (detail2.php) ในโฟลเดอร์ “orders”

คำสั่ง : PHP

ไดเรกทอรี่ : [skeleton]/protected/views/orders/detail2.php

```

<?php
$this->breadcrumbs = array('รายละเอียดการสั่งซื้อ');
$criteria = new CDbCriteria();
$criteria->condition = 'OrdersID=:ID';
$criteria->params = array(':ID'=>$_GET['id']);
$mOrderDetail = Orderdetail::model()->with(array('myproduct'))->findAll($criteria);
?>
<div class="container">
    <div class="row">
        <div class="col-lg-12">
            <h3>สรุปรายการสินค้าทั้งหมด</h3><hr>
            <a href="php echo Yii::app()-&gt;baseUrl;?&gt;"&gt;
                &lt;button class="btn btn-success"&gt;กลับไปหน้าหลัก&lt;/button&gt;&lt;/a&gt;&lt;br&gt;&lt;br&gt;
            &lt;table style="width:100%" border="1"&gt;
                &lt;thead bgcolor="#FF3"&gt;
                    &lt;tr&gt;
                        &lt;td class="text-center" style="width:8%&gt;&lt;strong&gt;ลำดับ&lt;/strong&gt;&lt;/td&gt;
                        &lt;td class="text-left" style="width:47%&gt;&lt;strong&gt;รายการสินค้า&lt;/strong&gt;&lt;/td&gt;
                        &lt;td class="text-right" style="width:15%&gt;&lt;strong&gt;จำนวน&lt;/strong&gt;&lt;/td&gt;
                        &lt;td class="text-right" style="width:15%&gt;&lt;strong&gt;ราคา&lt;/strong&gt;&lt;/td&gt;
                        &lt;td class="text-right" style="width:15%&gt;&lt;strong&gt;รวม&lt;/strong&gt;&lt;/td&gt;
                    &lt;/tr&gt;
                &lt;/thead&gt;
                &lt;tbody&gt;
</pre

```

```

<?php
    $row = 0;
    $total = 0;
    $amount = 0;
    $price = 0;
?>
<?php foreach ($mOrderDetail as $value) :?>
<?php
    $amount = $amount + $value->OrderAmount;
    $price = $price + $value->ProductPrice;
    $total = $total + $value->OrderPriceTotal;
    $row++;
?>
<tr>
    <td class="text-center"><?php echo $row;?></td>
    <td class="text-left"><?php echo $value->myproduct->ProductName;?></td>
    <td class="text-right"><?php echo number_format($value->OrderAmount);?></td>
    <td class="text-right"><?php echo number_format($value->ProductPrice);?>.-</td>
    <td class="text-right"><?php echo number_format($value->OrderPriceTotal);?>.-</td>
</tr>
<?php endforeach; ?>
</tbody>
<tfoot>
    <tr>
        <td class="text-right" colspan="2">รวม</td>
        <td class="text-right"><?php echo number_format($amount);?></td>
        <td class="text-right"><?php echo number_format($price);?>.-</td>
        <td class="text-right"><?php echo number_format($total);?>.-</td>
    </tr>
</tfoot>
</table>
</div>
</div>
</div>
<br><hr>

```

**สร้างไฟล์ :** Views (detail.php) ในไฟล์เดอร์ “orders”

คำสั่ง : PHP

ไดเรกทอรี่ : [skeleton]/protected/views/orders/message.php

```

<hr><br><br><br><br>
<center><h2><?php echo $message;?></h2></center>
<br><br><br> <hr>

```

สร้างไฟล์ : Views (order\_user.php) ในโฟลเดอร์ “orders”

คำสั่ง : PHP

ไดเรกทอรี่ : [skeleton]/protected/views/orders/order\_user.php

```
<?php $this->breadcrumbs = array('รายการสั่งซื้อสินค้า'); ?>
<br><h2>รายการสั่งซื้อสินค้า (ทั้งหมด)</h2><hr>
<?php $this->widget('zii.widgets.CListView', array(
    'dataProvider'=>$dataProvider, 'itemView'=>'_view',
)); ?>
```

สร้างไฟล์ : Views (\_view.php) ในโฟลเดอร์ “orders”

คำสั่ง : PHP

ไดเรกทอรี่ : [skeleton]/protected/views/orders/\_view.php

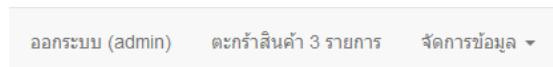
```
<b><?php echo CHtml::encode($data->getAttributeLabel('OrdersID'))); ?>:</b>
<?php echo CHtml::link(CHtml::encode($data->OrdersID), array('orders/detail/id/'.$data->Order-
sID)); ?><br />
<b><?php echo CHtml::encode($data->getAttributeLabel('CustomerID'))); ?>:</b>
<?php echo CHtml::encode($data->CustomerID); ?><br />
<b><?php echo CHtml::encode($data->getAttributeLabel('OrderDate'))); ?>:</b>
<?php echo CHtml::encode($data->OrderDate); ?><br />
<b><?php echo CHtml::encode($data->getAttributeLabel('OrderPriceTotal'))); ?>:</b>
<?php echo CHtml::encode($data->OrderPriceTotal); ?><br />
<br>
```

### ระบบสั่งซื้อสินค้า

ระบบสั่งซื้อสินค้าเป็นระบบหลักที่จำเป็นต้องมีสำหรับเว็บแอปพลิเคชัน E-Commerce ซึ่งระบบการทำงานมีดังนี้

ขั้นตอนที่ 1. สมาชิก สามารถทำการ “สั่งซื้อสินค้า” ได้ในหน้าแรกและหน้าเมนูสินค้า

ขั้นตอนที่ 2. การสั่งซื้อสินค้า ให้ “สมาชิก” กดปุ่ม “สั่งซื้อ” (สีเขียว) และเมื่อสมาชิก กดปุ่มระบบจะเข้าไปยัง Action Product ของ Orders Controller เพื่อทำการบันทึกข้อมูลลงใน “ตะกร้าสินค้า” หากบันทึกข้อมูลสำเร็จ ระบบจะพารามาชิกกลับมายัง “หน้าแรก” สุดท้ายจะแสดง “ข้อมูลตะกร้าสินค้า” ตรงเมนูด้านบน ดังรูปภาพด้านล่าง



ขั้นตอนที่ 3. สมาชิกสามารถ “ตรวจสอบรายการสินค้า” ภายในตะกร้าสินค้า ได้โดย กดที่ “ตะกร้าสินค้า 3 รายการ” ได้โดย และระบบจะเข้าไปยัง Action Showdetail ของ OrdersController เพื่อแสดงรายการสินค้าทั้งหมดที่มีอยู่ใน “ตะกร้าสินค้า” ที่เราได้สั่งซื้อไว้ ดังรูปภาพด้านล่าง

สรุประการสินค้าทั้งหมด

กลับไปหน้าหลัก				
ลำดับ	รายการสินค้า	จำนวน	ราคา	รวม
1	Canon EOS 60D Body Kiss	1	20,500.-	20,500.-
2	Canon EOS 7D + Len 18-200 IS	1	43,000.-	43,000.-
3	Canon EOS 5D Mark III Body	1	82,500.-	82,500.-
	รวม	3	146,000.-	146,000.-

[ล้างข้อมูล](#)

ขั้นตอนที่ 4. เมื่อสมาชิกต้องการจะ “สั่งซื้อสินค้า” สมาชิกกดปุ่ม “สั่งซื้อสินค้า” (สีน้ำเงิน) เพื่อบันทึกข้อมูลเข้าสู่ระบบฐานข้อมูล ซึ่ง Action Save ของ Orders Controller จะเป็นตัวจัดการบันทึกข้อมูลและแสดงผลดังรูปด้านล่าง

We Shopping หน้าแรก สินค้า แจ้งชำระเงิน ติดต่อเรา ออกใบอนุญาต (admin) ลงทะเบียนค้าไม่เพเบลินค้า จัดการข้อมูล ▾

### บันทึกข้อมูลเสร็จเรียบร้อยแล้ว

ขั้นตอนที่ 5. เมื่อสมาชิกได้ทำการ “สั่งซื้อสินค้าเรียบร้อยแล้ว” สมาชิกสามารถตรวจสอบข้อมูลการสั่งซื้อสินค้าของสมาชิกได้ โดยสมาชิกคลิกที่แท็บเมนูด้านบน “จัดการ” -> “ข้อมูลการสั่งซื้อสินค้า” เมื่อสมาชิกคลิกเมนูดังกล่าว ระบบจะเข้าไปปั้ง Action User ของ Orders Controller เพื่อทำการแสดงรายการสั่งซื้อสินค้าทั้งหมดของสมาชิกให้ทราบ

We Shopping หน้าแรก สินค้า แจ้งชำระเงิน ติดต่อเรา ออกใบอนุญาต (admin) ลงทะเบียนค้าไม่เพเบลินค้า จัดการข้อมูล ▾

Home / รายการสั่งซื้อสินค้า

### รายการสั่งซื้อสินค้า (ทั้งหมด)

Displaying 1-2 of 1 result.

Orders: ORD00001  
 Customer: 1  
 Order Date: 2014-07-28 21:02:12  
 Order Price Total: 146000

Orders: ORD00002  
 Customer: 1  
 Order Date: 2014-07-28 22:41:17  
 Order Price Total: 146000

ขั้นตอนที่ 6. สามารถคลิกดูรายการสินค้าภายในรายการสั่งซื้อสินค้าได้ โดยคลิกที่ “เลขที่” ในสั่งซื้อสินค้า” จากนั้นระบบจะเข้าไปยัง Action Detail ของ Orders Controller เพื่อจะแสดงรายการสินค้าทั้งหมด

หน้าจอแสดงรายละเอียดการสั่งซื้อที่ระบุหมายเลข 123456789. ข้อมูลแสดงในตารางดังนี้:

ลำดับ	รายการสินค้า	จำนวน	ราคา	รวม
1	Canon EOS 5D Mark III Body	1	82,500.-	82,500.-
2	Canon EOS 60D Body Kiss	1	20,500.-	20,500.-
3	Canon EOS 7D + Len 18-200 IS	1	43,000.-	43,000.-
	รวม	3	146,000.-	146,000.-

### ระบบการแจ้งชำระเงิน

ระบบการแจ้งชำระเงินเป็นระบบที่ใช้เพื่อ ให้สมาชิก “แจ้งข้อมูลการชำระเงิน” ซึ่งข้อมูลที่สมาชิกต้องมีประกอบด้วย เลขที่ใบสั่งซื้อสินค้า ยอดที่ชำระ และรายละเอียดเพิ่มเติม สมาชิกกดเมนู “แจ้งชำระเงิน” ด้านบนแท็บเมนู ซึ่งระบบการทำงานจะทำงานภายใต้ Action Index ของ Payment Controller เพื่อทำการบันทึกข้อมูลการชำระเงิน ลงฐานข้อมูล

หน้าจอแสดงฟอร์มแจ้งชำระเงิน:

กรอกข้อมูลที่มีเครื่องหมาย \* อยู่ในคีย์บอร์ด.

เลขที่ใบสั่งซื้อสินค้า *	<input type="text"/>
ยอดที่ชำระ *	<input type="text"/>
รายละเอียดเพิ่มเติม *	<input type="text"/>

ปุ่ม “บันทึกข้อมูล”

## ระบบการติดต่อทางร้าน

ระบบการติดต่อทางร้าน มีไว้เพื่อใช้สำหรับ สมาชิก ต้องการสอบถามข้อมูลหรือข้อกติดต่อ พูดคุยกับทางร้าน โดยสามารถคลิกที่แท็บเมนูด้านบน “ติดต่อเรา” ซึ่งระบบทั้งหมดนี้ จะทำงานภายใต้ Action Index ของ Contact Controller ทั้งหมด และเพื่อทำการบันทึกข้อมูลการติดต่อ ลงฐานข้อมูล

## ระบบจัดการข้อมูลสินค้า ข้อมูลการสั่งซื้อ การชำระเงิน และข้อมูลการติดต่อ

ระบบจัดการข้อมูลสินค้า ข้อมูลการสั่งซื้อ การชำระเงิน และข้อมูลการติดต่อ ถูกสร้างขึ้นด้วยใช้ Gii System (CRUD Generator) ซึ่งการทำงานของระบบต่าง ๆ จะเหมือนกันทุกระบบ ซึ่งหมายความว่าไม่ต้องเรียนรู้ใหม่ เนื่องจากระบบการทำงานไม่ซับซ้อนและเรียนรู้ได้เร็ว เพราะเป็นระบบสำเร็จรูป แต่permจะอธิบายระบบการทำงานย่อยๆ ได้จาก Gii System (CRUD Generator)

### Gii System (ศึกษาเพิ่มเติมได้ที่ บทที่ 4 วิธีการใช้งาน Gii Code Generator)

ระบบย่อที่จะได้หลังจากการสร้างไฟล์ ด้วย Gii System (CRUD Generator) จะประกอบไปด้วย 4 ระบบดังนี้

1. C: Create ระบบการสร้างข้อมูลใหม่
2. R: Read ระบบการดึงข้อมูลออกมาแสดงผล
3. U: Update ระบบการแก้ไขข้อมูล
4. D: Delete ระบบการลบข้อมูล

ซึ่งหลังจากการสร้างไฟล์ด้วย Gii System (CRUD Generator) เราจะได้ระบบการจัดการข้อมูล (เพิ่ม, ลบ, แก้ไข, บันทึก, และค้นหา) มาให้เราใช้งาน โดยอัตโนมัติ

## สร้างระบบ : การจัดการข้อมูล ด้วย Gii Bootstrap Generator

หลังจากที่เราได้ทำการติดตั้ง Yii Booster + Theme Bootstrap ความสามารถพิเศษอย่างหนึ่งที่เราจำเป็นต้องรู้ก็คือ Gii Bootstrap Generator (CRUD Generator) คือการสร้างไฟล์ Create, Read, Update, Delete ตามรูปแบบ Form ของ Bootstrap ซึ่งการใช้งานเหมือน CRUD Generator ทุกประการ

### ระบบจัดการข้อมูลทั้งหมด 4 ระบบ ประกอบด้วย

- 1.ระบบจัดการข้อมูลสินค้า (ProductController.php)
- 2.ระบบจัดการข้อมูลสั่งซื้อ (OrderController.php)
- 3.ระบบจัดการข้อมูลการชำระเงิน (PaymentController.php)
- 4.ระบบจัดการข้อมูลการติดต่อเรา (ContactController.php)

Gii Bootstrap Generator || เข้าไปที่ URL <http://localhost/weshopping/gii/bootstrap>

The screenshot shows the Gii code generator interface. The left sidebar has a 'Generators' section with several options: Controller Generator, Crud Generator, Form Generator, Model Generator, Module Generator, and Bootstrap Generator. The 'Bootstrap Generator' option is highlighted with a red circle. The main content area is titled 'Bootstrap Generator'. It contains fields for 'Model Class' (with 'Controller' highlighted in yellow), 'Controller ID' (empty), 'Base Controller Class' (with 'Controller' highlighted in yellow), 'Code Template' (set to 'default'), and a 'Preview' button. At the bottom, it says 'Powered by Yii Framework'.

ต่อไปให้เรากรอกชื่อ Model ลงในช่อง Model Class และกรอก Controller ID เป็น backoffice/ControllerID เพื่อตัวอย่าง

ถึงที่เราต้องสร้าง ProductController, OrderController, PaymentController, ContactController รวมทั้งหมด 4 Controller

The screenshot shows the Gii Bootstrap Generator interface with the 'Model Class' field containing 'Product' and the 'Controller ID' field containing 'backoffice/product'.

สมมุติ สร้าง ProductController ใน BackOffice  
หรือ ดาวโหลด BackOffice Modules ทั้งหมด ได้ที่  
ลิงก์นี้ || ดาวโหลดที่ URL <http://goo.gl/NooggK>

## บทที่ 10 ตัวอย่างกรณีศึกษาระบบชี้อุปกรณ์ขายสินค้าออนไลน์พื้นฐาน

เมื่อเราสร้างระบบจัดการข้อมูล 4 ระบบ เตรียมเรียบร้อยแล้ว ให้เราล็อกอินเข้าสู่ระบบด้วย Username, Password ของ “admin” ! เพื่อเข้าใช้งาน BackOffice เพื่อใช้งานระบบจัดการข้อมูล สามารถเข้าดูรายการเมนูตามรูปด้านล่างนี้

ออกระบบ (admin)	ตะกร้าสินค้าในเพิ่มสินค้า	จัดการข้อมูล ▾
		<ul style="list-style-type: none"><li>ข้อมูลการซื้อขาย แสดงข้อมูลลูกค้า เปลี่ยนชื่อผู้ดูแลระบบ เปลี่ยนรหัสผ่านใหม่</li><li>จัดการข้อมูลสินค้า จัดการข้อมูลความต้องการ จัดการข้อมูลการชำระเงิน จัดการข้อมูลติดต่อเรา</li></ul>

Operations  
Create Product  
Manage Product  
Displaying 1-10 of 1 result.

ตัวอย่าง : ระบบจัดการข้อมูลสินค้า || ที่นี่ <http://localhost/weshopping/backoffice/product>

We Shopping หน้าแรก สินค้า แจ้งชำระเงิน ติดต่อเรา ออกระบบ (admin) ตะกร้าสินค้าในเพิ่มสินค้า จัดการข้อมูล ▾

Home / Products

### Products

Product: CCA001  
Product Name: Canon EOS 5D Mark III Body  
Sale Price: 82500  
Amount: 10  
Create Date: 2013-08-30 20:44:55  
Product Img: CCA001.jpg

Operations  
Create Product  
Manage Product  
Displaying 1-10 of 1 result.

ตัวอย่าง : ระบบจัดการข้อมูลสั่งซื้อสินค้า || ที่นี่ <http://localhost/weshopping/backoffice/order>

We Shopping หน้าแรก สินค้า แจ้งชำระเงิน ติดต่อเรา ออกระบบ (admin) ตะกร้าสินค้าในเพิ่มสินค้า จัดการข้อมูล ▾

Home / Orders

### Orders

No results found.

Operations  
Create Order  
Manage Order

ตัวอย่าง : ระบบจัดการข้อมูลการชำระเงิน || ที่นี่ <http://localhost/weshopping/backoffice/payment>

We Shopping หน้าแรก สินค้า แจ้งชำระเงิน ติดต่อเรา ออกระบบ (admin) ตะกร้าสินค้าในเพิ่มสินค้า จัดการข้อมูล ▾

Home / Payments

### Payments

No results found.

Operations  
Create Payment  
Manage Payment

ตัวอย่าง : ระบบจัดการข้อมูลติดต่อเรา || ที่นี่ <http://localhost/weshopping/backoffice/contact>

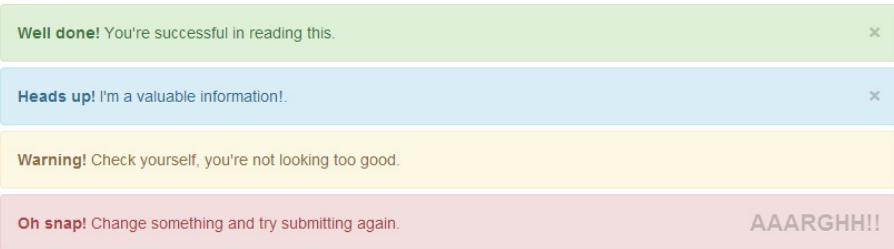
The screenshot shows a web-based administration interface for managing contacts. At the top, there's a navigation bar with links for 'We Shopping', 'หน้าแรก' (Home), 'สินค้า' (Products), 'แจ้งข่าวสารเงิน' (Financial News), 'ติดต่อเรา' (Contact Us), 'อัปโหลดรูป' (Upload Photo), 'ลงกรุํรับสินค้าในเพบสินค้า' (Receive goods in the shopping cart), and 'จัดการข้อมูล' (Manage Data) with a dropdown arrow. Below the navigation is a breadcrumb trail: 'Home / Contacts'. On the right side of the header, there are buttons for 'Operations', 'Create Contact', and 'Manage Contact'. The main content area displays a message 'No results found.' and features a stylized logo composed of three overlapping leaf-like shapes in orange, green, and blue.

กานต์ ก้าวกล้า  
fenics128[at]gmail.com

# ກາຄພນວກ ດ

## (ຄຸນມືອກຮາຍ ໄຊງານ Yii Booster V.4.0.1)

## Yii Booster v.4.0.1 : Widget Alert



```
$user = Yii::app()->getComponent('user');
$user->setFlash('success',
    "<strong>Well done!</strong> You're successful in reading this."
);
$user->setFlash('info',
    "<strong>Heads up!</strong> I'm a valuable information!."
);
$user->setFlash('warning',
    "<strong>Warning!</strong> Check yourself, you're not looking too good."
);
$user->setFlash('error',
    "<strong>Oh snap!</strong> Change something and try submitting again."
);

// Render them all with single `TbAlert`
$this->widget('booster.widgets.TbAlert', array(
    'fade'=>true,
    'closeText'=>'&times;; // false equals no close link
    'events'=>array(),
    'htmlOptions'=>array(),
    'userComponentId'=>'user',
    'alerts'=>array( // configurations per alert type
        // success, info, warning, error or danger
        'success'=>array('closeText'=>'&times;'),
        'info', // you don't need to specify full config
        'warning'=>array('closeText'=>false),
        'error'=>array('closeText'=>'AAARGHH!!')
    ),
));
)
```

**Yii Booster v.4.0.1 : Widget Badge**

```
$this->widget('booster\widgets\TbBadge',
array(
    'context' => 'success',
    // 'default', 'success', 'info', 'warning', 'danger'
    'label' => '2',
)
);
```

**Yii Booster v.4.0.1 : Widget Badge**

I am BATMAN!

```
$this->widget(
    'booster\widgets\TbJumbotron',
array(
    'heading' => 'I am BATMAN!',
)
);
```

**Yii Booster v.4.0.1 : Widget Label**

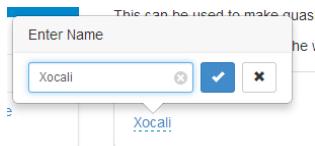
**Success**

```
$this->widget(
    'booster\widgets\TbLabel',
array(
    'context' => 'success',
    // 'default', 'primary', 'success', 'info', 'warning', 'danger'
    'label' => 'Success',
)
);
```

## Yii Booster v.4.0.1 : Widget Progress

```
$this->widget('booster.widgets.TbProgress',array(  
    'context' => 'success', // 'success', 'info', 'warning', or 'danger'  
    'percent' => 60,  
)  
);
```

## Yii Booster v.4.0.1 : Widget Editable Field



```
$this->widget('booster.widgets.TbEditableField',array(  
    'type' => 'text',  
    'model' => $model,  
    'attribute' => 'name', // $model->name will be editable  
    'url' => $endpoint, //url for submit data  
)  
);
```

## Yii Booster v.4.0.1 : Widget Editable Detail View



```
$this->widget('booster.widgets.TbEditableDetailView',array(  
    'id' => 'region-details',  
    'data' => $model,  
    'url' => $endpoint,  
    'attributes' => array(  
        'country_code',  
        'name',  
)  
)  
);
```

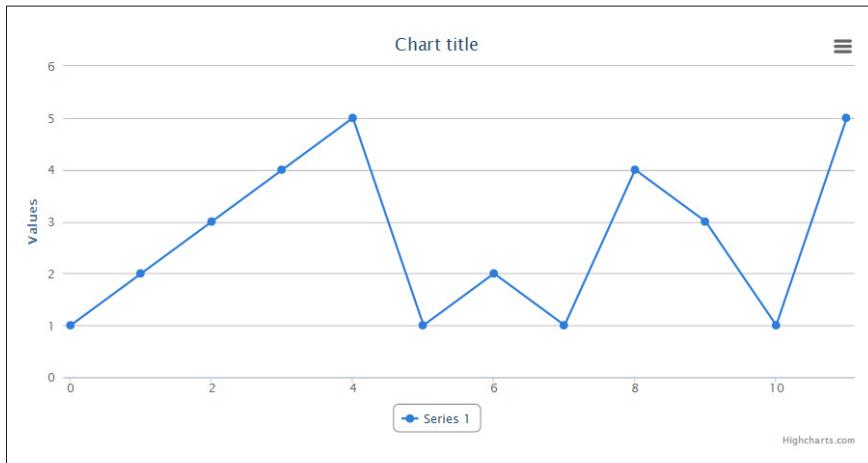
## Yii Booster v.4.0.1 : Widget Editable Column

		Displaying 1-10 of 3940 results.
Country code	Name	
CA	Alberta	
CA	British Columbia	
CA	Manitoba	
CA	New Brunswick	
CA	Newfoundland	
CA	Nova Scotia	
CA	Nunavut	
CA	Ontario	
CA	Prince Edward Island	
CA	Quebec	

« 1 2 3 4 5 6 7 8 9 10 »

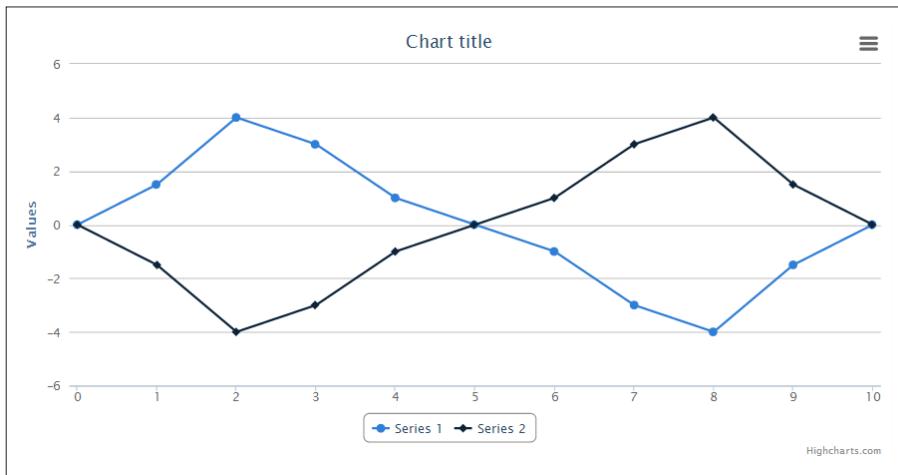
```
// data
$dataProvider = new CActiveDataProvider('Region');
// TbGridView
$this->widget(
    'booster.widgets.TbGridView',
    array(
        'type' => 'striped bordered',
        'dataProvider' => $dataProvider,
        'columns' => array(
            'country_code',
            array(
                'class' => 'booster.widgets.TbEditableColumn',
                'name' => 'name',
                'sortable' => false,
                'editable' => array(
                    'url' => $this->owner->createUrl('example/editable'),
                    'placement' => 'right',
                    'inputclass' => 'span3'
                )
            )
        ),
    )
);
```

## Yii Booster v.4.0.1 : Widget TbHighCharts (Basic)



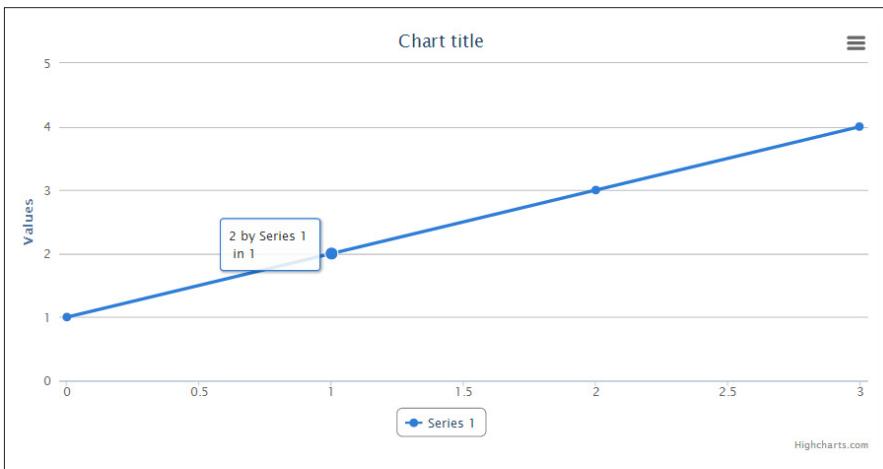
```
$this->widget('booster.widgets.TbHighCharts',array  
    'options' => array(  
        'series' => array(  
            ['data'=>[1, 2, 3, 4, 5, 1, 2, 1, 4, 3, 1, 5]]  
        )  
    )  
);
```

## Yii Booster v.4.0.1 : Widget TbHighCharts (Zeros in data)



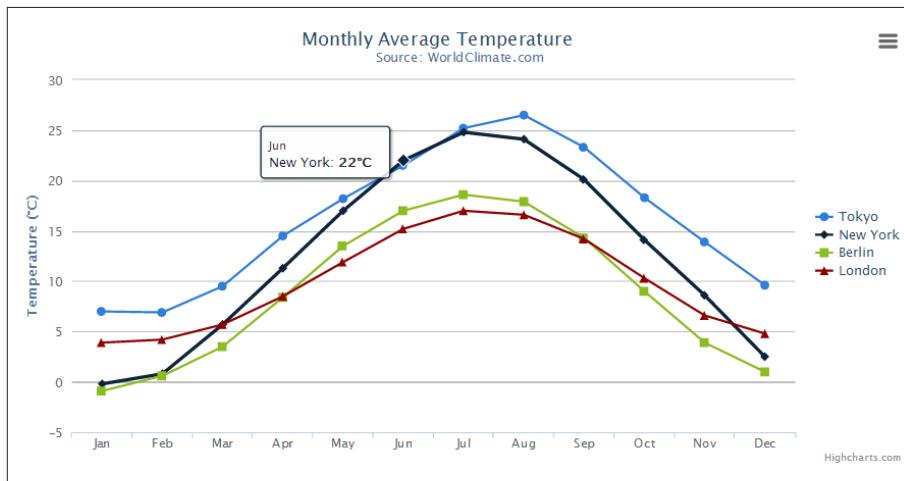
```
$this->widget('booster.widgets.TbHighCharts',array(
    'options'=>array(
        'series'=>array(
            ['data'=>[0, 1.5, 4, 3, 1, 0, -1, -3, -4, -1.5, 0]],
            [
                'data' => new CJavaScriptExpression(
                    '[0, -3/2, -2*2, -0.3e1, -Math.pow(354,0), 1-1, 3/3, 3, 4, 4-2.5, 0]'.
                )
            ]
        )
    )
));
});
```

#### Yii Booster v.4.0.1 : Widget TbHighCharts (Javascript callback in options)



```
$this->widget('booster.widgets.TbHighCharts',array(
    'options' => array(
        'tooltip' => array(
            'formatter' => new CJavaScriptExpression(
                'function() {
                    return Highcharts.numberFormat(this.y, 0) + ` by ` + this.series.name` in `+ this.x;
                }',
            ),
        ),
        'series'=>array(['data'=>[1, 2, 3, 4]])
    )
));
});
```

## Yii Booster v.4.0.1 : Widget TbHighCharts (Official demo)



```
$this->widget('booster.widgets.TbHighCharts',array(
    'options'=>array(
        'title'=>array('text'=>'Monthly Average Temperature','x'=>-20), //center
        'subtitle'=>array('text'=>'Source:WorldClimate.com','x'=>-20),
        'xAxis'=>array('categories'=>['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec']),
        'yAxis'=>array(
            'title'=>array('text'=>'Temperature (°C)'),
            'plotLines'=>[['value'=>0,'width'=>1,'color'=> '#808080']],
        ),
        'tooltip'=>array('valueSuffix'=>'°C'),
        'legend'=>array(
            'layout'=>'vertical','align'=>'right','verticalAlign'=>'middle','borderWidth'=>0
        ),
        'series'=>array(
            ['name'=>'Tokyo','data'=>[7.0, 6.9, 9.5, 14.5, 18.2, 21.5, 25.2, 26.5, 23.3, 18.3, 13.9, 9.6]],
            ['name'=>'New York','data'=>[-0.2, 0.8, 5.7, 11.3, 17.0, 22.0, 24.8, 24.1, 20.1, 14.1, 8.6, 2.5]],
            ['name'=>'Berlin','data'=> [-0.9, 0.6, 3.5, 8.4, 13.5, 17.0, 18.6, 17.9, 14.3, 9.0, 3.9, 1.0]],
            ['name'=>'London','data'=> [3.9, 4.2, 5.7, 8.5, 11.9, 15.2, 17.0, 16.6, 14.2, 10.3, 6.6, 4.8]]
        )
    ),
    'htmlOptions'=>array(
        'style'=>'min-width:310px;height:400px;margin:0 auto'
    )
));
```

## Yii Booster v.4.0.1 : Widget Active Form (Vertical)

The screenshot shows a vertical layout for an Active Form. It contains four fields: a text input field, a password input field, a checkbox labeled "Check me out", and a submit button labeled "Login". The fields are stacked vertically from top to bottom.

```
$form = $this->beginWidget('booster.widgets.TbActiveForm',array(
    'id'=>'verticalForm',
    'htmlOptions'=>array('class'=>'well'), // for inset effect
))
);
echo $form->textFieldGroup($model, 'textField');
echo $form->passwordFieldGroup($model, 'password');
echo $form->checkboxGroup($model, 'checkbox');
$this->widget('booster.widgets.TbButton',
    array('buttonType' => 'submit', 'label' => 'Login')
);
$this->endWidget();
unset($form);
```

## Yii Booster v.4.0.1 : Widget Active Form (Inline Form)

The screenshot shows an inline layout for an Active Form. It contains three separate input fields: "Text input", "Password", and "Check me out", followed by a submit button labeled "Log in". The fields are arranged horizontally.

```
$form = $this->beginWidget('booster.widgets.TbActiveForm',array(
    'id' => 'inlineForm',
    'type' => 'inline',
    'htmlOptions' => array('class' => 'well'),
));
echo $form->textFieldGroup($model, 'textField');
echo $form->passwordFieldGroup($model, 'password');
echo $form->checkboxGroup($model, 'checkbox');
$this->widget('booster.widgets.TbButton',
    array('buttonType' => 'submit', 'label' => 'Log in')
);
$this->endWidget();
unset($form);
```

## Yii Booster v.4.0.1 : Widget Active Form (Horizontal Form)

A screenshot of a Yii Booster horizontal form. It contains two text input fields, one password input field, a checkbox labeled "Check me out", and a "Login" button.

```
$form = $this->beginWidget('booster.widgets.TbActiveForm',array(  
    'id' => 'verticalForm','type' => 'horizontal',  
    'htmlOptions' => array('class' => 'well'), // for inset effect  
));  
echo $form->textFieldGroup($model, 'textField');  
echo $form->passwordFieldGroup($model, 'password');  
echo $form->checkboxGroup($model, 'checkbox');  
$this->widget('booster.widgets.TbButton',  
    array('buttonType' => 'submit', 'label' => 'Login')  
);  
$this->endWidget();  
unset($form);
```

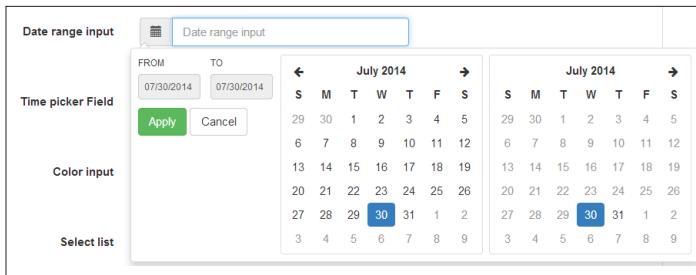
## Yii Booster v.4.0.1 : Widget Active Form (Inputs showcase)

A screenshot of a Yii Booster text input field. The placeholder text "In addition to freeform text, any HTML5 text-based input appears like so." is visible below the input field.

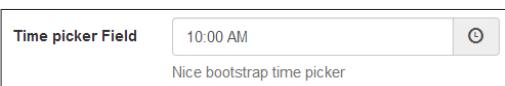
```
$form->textFieldGroup($model,'textField',array(  
    'wrapperHtmlOptions'=>array('class'=>'col-sm-5'),  
    'hint'=>'In addition to freeform text, any HTML5 text-based input appears like so.'  
));
```

A screenshot of a Yii Booster date input field. It includes a calendar icon and the placeholder text "Click inside! This is a super cool date field."

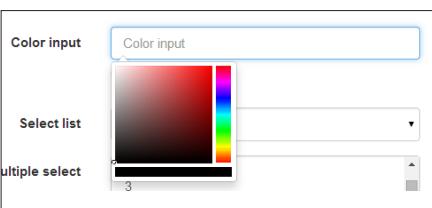
```
$form->datePickerGroup($model,'dateField',array(  
    'widgetOptions'=>array('options'=>array('language'=>'es')),  
    'wrapperHtmlOptions'=>array('class'=>'col-sm-5'),  
    'hint'=>'Click inside! This is a super cool date field.',  
    'prepend'=><i class="glyphicon glyphicon-calendar"></i>'  
));
```



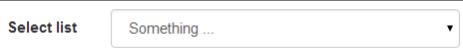
```
$form->dateRangeGroup($model,'dateRangeField',array
    'widgetOptions'=>array(
        'callback' => 'js:function(start, end){console.log(start.toString("MMMM d, yyyy") + " - " + end.
        toString("MMMM d, yyyy"));}',
        ),
        'wrapperHtmlOptions'=>array('class'=>'col-sm-5'),
        'hint' => 'Click inside! An even a date range field!.',
        'prepend' => '<i class="glyphicon glyphicon-calendar"></i>'
));
});
```



```
$form->timePickerGroup($model,'timepicker',array(
    'widgetOptions' => array(
        'wrapperHtmlOptions' => array('class' => 'col-sm-3'),
        ),
        'hint' => 'Nice bootstrap time picker',
));
});
```



```
$form->colorpickerGroup($model,'colorpicker',array(
    'wrapperHtmlOptions' => array('class' => 'col-sm-5'),
    'hint' => 'Color fields, not bad',
));
});
```



```
$form->dropDownListGroup($model,'dropdown',array(
    'wrapperHtmlOptions'=>array('class' => 'col-sm-5',
    ),
    'widgetOptions' => array(
        'data' => array('Something ...','1','2','3','4','5'),
        'htmlOptions' => array(),
    )
));
});
```



```
$form->dropDownListGroup($model,'multiDropdown',array(
    'wrapperHtmlOptions' => array(
        'class' => 'col-sm-5',
    ),
    'widgetOptions' => array(
        'data' => array('1','2','3','4','5'),
        'htmlOptions' => array('multiple' => true),
    )
));
});
```



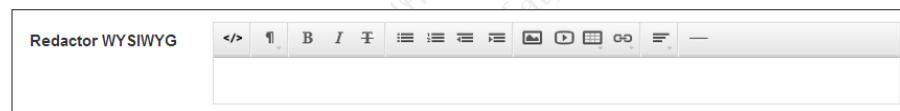
```
$form->select2Group($model,'select2',array(
    'wrapperHtmlOptions'=>array('class' => 'col-sm-5'),
    'widgetOptions' => array(
        'asDropDownList' => false,
        'options' => array(
            'tags' => array('clever', 'is', 'better', 'clevertech'),
            'placeholder' => 'type clever, or is, or just type!',
            /* 'width' => '40%', */
            'tokenSeparators' => array(,,')
        )
    )
));
});
```

**File input**  ไม่ได้เลือกไฟล์ใด

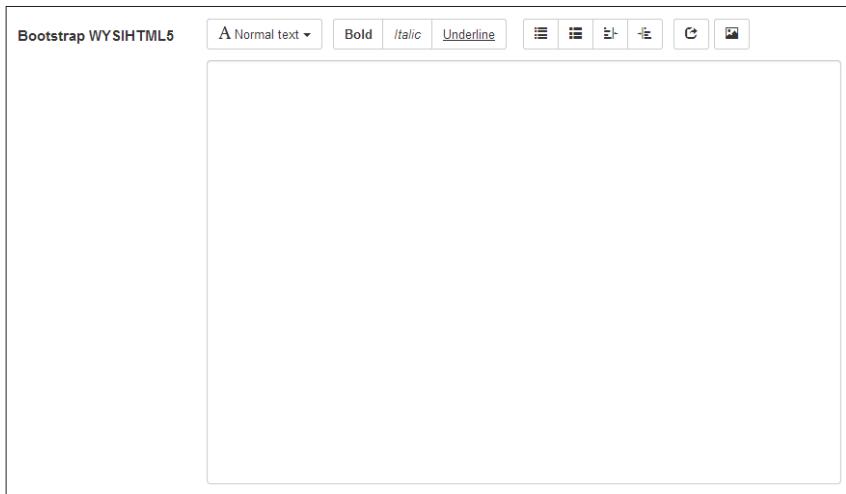
```
$form->fileFieldGroup($model, 'fileField', array(
    'wrapperHtmlOptions' => array(
        'class' => 'col-sm-5',
    ),
));
```



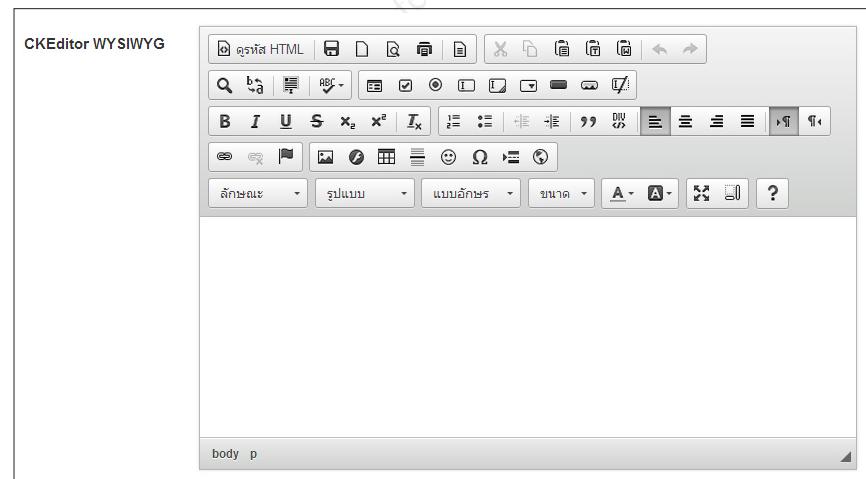
```
$form->textAreaGroup($model, 'textarea', array(
    'wrapperHtmlOptions' => array(
        'class' => 'col-sm-5',
    ),
    'widgetOptions' => array(
        'htmlOptions' => array('rows' => 5),
    )
));
```



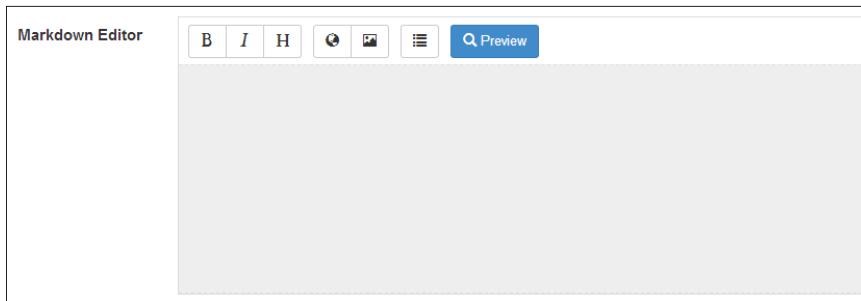
```
$form->redactorGroup($model, 'redactor', array(
    'widgetOptions' => array(
        'editorOptions' => array(
            'class' => 'span4',
            'rows' => 5,
            'options' => array(
                'plugins' => array('clips', 'fontfamily'),
                'lang' => 'sv'
            )
        )
    )
));
```



```
$form->html5EditorGroup($model,'wysihtml5',array(  
    'widgetOptions' => array(  
        'editorOptions' => array(  
            'class' => 'span4',  
            'rows' => 5,  
            'height' => '200',  
            'options' => array('color' => true)  
        ),  
    ),  
));
```



```
$form->ckEditorGroup($model,'ckeditor',array(
    'wrapperHtmlOptions' => array(
        /* 'class' => 'col-sm-5', */
    ),
    'widgetOptions' => array(
        'editorOptions' => array(
            'fullpage' => 'js:true',
        )
    )
));
});
```



```
$form->markdownEditorGroup($model,'markdown',array(
    'widgetOptions' => array(
        'htmlOptions' => array('style'=>'height: 200px;')
    )
));
});
```



```
$form->switchGroup($model, 'switch',array(
    'widgetOptions' => array(
        'events'=>array(
            'switchChange'=>'js:function(event, state) {
                console.log(this); // DOM element
                console.log(event); // jQuery event
                console.log(state); // true | false
            }'
        )
    )
));
});
```

**Disabled input**

Disabled input

```
$form->textFieldGroup($model,'disabled',array(
    'wrapperHtmlOptions' => array(
        'class' => 'col-sm-5',
    ),
    'widgetOptions' => array(
        'htmlOptions' => array('disabled' => true)
    )
));
```

**Prepend text**

@

Prepend text

```
$form->textFieldGroup($model,'prepend',array(
    'wrapperHtmlOptions' => array(
        'class' => 'col-sm-5',
    ),
    'prepend' => '@'
));
```

**Append text**

Append text

.00

```
$form->textFieldGroup($model,'append',array(
    'wrapperHtmlOptions' => array(
        'class' => 'col-sm-5',
    ),
    'append' => '.00'
));
```



Disabled checkbox

```
$form->checkboxGroup($model,'disabledCheckbox',array(
    'widgetOptions' => array(
        'htmlOptions' => array(
            'disabled' => true
        )
    )
));
```

**Inline checkboxes** 1  2  3

```
$form->checkboxListGroup($model, 'inlineCheckboxes', array(
    'widgetOptions' => array(
        'data' => array('1', '2', '3')
    ),
    'inline'=>true
));
```

**Checkboxes**

- Option one is this and that - be sure to include why it's great
- Option two can also be checked and included in form results
- Option three can - yes, you guessed it - also be checked and included in form results

**Note:** Labels surround all the options for much larger click areas.

```
$form->checkboxListGroup($model, 'checkboxes', array(
    'widgetOptions' => array(
        'data' => array(
            'Option one is this and that - be sure to include why it\'s great',
            'Option two can also be checked and included in form results',
            'Option three can - yes, you guessed it - also be checked and included in form results',
        ),
    ),
    'hint' => '<strong>Note:</strong> Labels surround all the options for much larger click areas.'
));
```

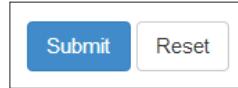
**Radio Button** Radio Button

```
$form->radioButtonGroup($model, 'radioButton');
```

**Radio buttons**

- Option one is this and that - be sure to include why it's great
- Option two can is something else and selecting it will deselect option one

```
$form->radioButtonListGroup($model, 'radioButtons', array(
    'widgetOptions' => array(
        'data' => array(
            'Option one is this and that - be sure to include why it\'s great',
            'Option two can is something else and selecting it will deselect option one',
        )
    )
));
```



```
$this->widget('booster.widgets.TbButton',array(  
    'buttonType'=>'submit',  
    'context'=>'primary',  
    'label'=>'Submit'  
));  
$this->widget('booster.widgets.TbButton',array(  
    'buttonType'=>'reset',  
    'label' => 'Reset'  
));
```

### Yii Booster v.4.0.1 : Widget Button



```
$this->widget('booster.widgets.TbButton',array(  
    'label' => 'Common',  
));  
$this->widget('booster.widgets.TbButton',array(  
    'label' => 'Primary','context' => 'primary',  
));  
$this->widget('booster.widgets.TbButton',array(  
    'label' => 'Success','context' => 'success',  
));  
$this->widget('booster.widgets.TbButton',array(  
    'label' => 'Info','context' => 'info',  
));  
$this->widget('booster.widgets.TbButton',array(  
    'label' => 'Warning','context' => 'warning',  
));  
$this->widget('booster.widgets.TbButton',array(  
    'label' => 'Danger','context' => 'danger',  
));  
$this->widget('booster.widgets.TbButton',array(  
    'label' => 'Danger','context' => 'link',  
));
```



```
$this->widget('booster.widgets.TbButton',array(
    'label' => 'Large','size' => 'large'
));
$this->widget('booster.widgets.TbButton',array(
    'label' => 'Default','size' => 'default'
));
$this->widget('booster.widgets.TbButton',array(
    'label' => 'Small','size' => 'small'
));
$this->widget('booster.widgets.TbButton',array(
    'label' => 'Extra small','size' => 'extra_small'
));
```

### Yii Booster v.4.0.1 : Widget Button Group



```
$this->widget('booster.widgets.TbButtonGroup',array(
    'buttons' => array(
        array('label' => 'Left', 'url' => '#'),
        array('label' => 'Middle', 'url' => '#'),
        array('label' => 'Right', 'url' => '#')
    ),
));
```

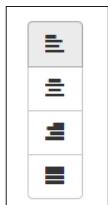


```
$this->widget('booster.widgets.TbButtonGroup',array(
    'justified' => true,
    'buttons' => array(
        array('label' => 'Left', 'url' => '#'),
        array('label' => 'Middle', 'url' => '#'),
        array('label' => 'Right', 'url' => '#')
    ),
));
```



```
$this->widget('booster.widgets.TbButtonGroup',array(
    'buttons' => array(
        array('label' => '1', 'url' => '#'),
        array('label' => '2', 'url' => '#'),
        array('label' => '3', 'url' => '#'),
        array('label' => '4', 'url' => '#'),
    ),
));
$this->widget('booster.widgets.TbButtonGroup',array(
    'buttons' => array(
        array('label' => '5', 'url' => '#'),
        array('label' => '6', 'url' => '#'),
        array('label' => '7', 'url' => '#'),
    ),
));
$this->widget('booster.widgets.TbButtonGroup',array(
    'buttons' => array(
        array('label' => '8', 'url' => '#'),
    ),
));

```



```
$this->widget('booster.widgets.TbButtonGroup',array(
    'stacked' => true,
    'buttons' => array(
        array('buttonType' => 'button', 'url' => '#', 'icon' => 'align-left'),
        array('buttonType' => 'button', 'url' => '#', 'icon' => 'align-center'),
        array('buttonType' => 'button', 'url' => '#', 'icon' => 'align-right'),
        array('buttonType' => 'button', 'url' => '#', 'icon' => 'align-justify'),
    ),
));

```

← Lefst    ↑ Middle    → Right

```
$this->widget('booster\widgets\TbButtonGroup',array(
    'context' => 'primary',
    'toggle' => 'radio',
    'buttons' => array(
        array('label' => 'Lefst', 'icon'=>'arrow-left'),
        array('label' => 'Middle', 'icon'=>'arrow-up'),
        array('label' => 'Right', 'icon'=>'arrow-right'),
    ),
));
```



```
$this->widget('booster\widgets\TbButtonGroup',array(
    'toggle' => 'checkbox',
    'buttons' => array(
        array('label' => 'Left'),
        array('label' => 'Middle'),
        array('label' => 'Right'),
    ),
));
```



```
$this->widget('booster\widgets\TbButtonGroup',array(
    'size' => 'large',
    'context' => 'warning',
    'buttons' => array(
        array(
            'label' => 'Warning',
            'items' => array(
                array('label' => 'Action', 'url' => '#'),
                array('label' => 'Another action', 'url' => '#'),
                array('label' => 'Something else', 'url' => '#'),
            )
        ),
    ),
));
```



```
$this->widget('booster.widgets.TbButtonGroup',array(  
    'context' => 'primary',  
    // 'primary', 'info', 'success', 'warning', 'danger' or 'inverse'  
    'buttons' => array(  
        array('label' => 'Action', 'url' => '#'),  
        array(  
            'items' => array(  
                array('label' => 'Action', 'url' => '#'),  
                array('label' => 'Another action', 'url' => '#'),  
            )  
        ),  
    ),  
));
```



```
$this->widget('booster.widgets.TbButtonGroup',array(  
    'size' => 'large',  
    'buttons' => array(  
        array(  
            'label' => 'Action',  
            'items' => array(  
                array('label' => 'Action', 'url' => '#'),  
                array('label' => 'Another action', 'url' => '#'),  
            ),  
            'label' => 'More options',  
            'items' => array(  
                array('label'=>'Second level link', 'url'=> '#'),  
                array(  
                    'label'=>'Second level link',  
                    'items'=>array(array('label'=>'Third level link', 'url'=> '#'),)  
                ),  
            ),  
        ),  
    ),  
)),  
));
```

**Yii Booster v.4.0.1 : Widget Switch**

```
$this->widget('booster.widgets.TbSwitch',array(
    'name' => 'testToggleButton',
    'events' => array(
        'switchChange' => 'js:function($el, status, e){console.log($el, status, e);}'
    )
));
```



```
$this->widget('booster.widgets.TbSwitch',array(
    'name' => 'testToggleButtonB',
    'options' => array(
        'size' => 'large', //null, 'mini', 'small', 'normal', 'large
        'onColor' => 'success', // 'primary', 'info', 'success', 'warning', 'danger', 'default'
        'offColor' => 'danger', // 'primary', 'info', 'success', 'warning', 'danger', 'default'
    ),
));
```

**Yii Booster v.4.0.1 : Widget Select2**

```
$this->widget('booster.widgets.TbSelect2',array(
    'asDropDownList' => false,
    'name' => 'clevertech',
    'options' => array(
        'tags' => array('clever', 'is', 'better', 'clevertech'),
        'placeholder' => 'type clever, or is, or just type!',
        'width' => '40%',
        'tokenSeparators' => array('/', ' ')
    )
));
```

## Yii Booster v.4.0.1 : Widget Typeahead input



```
$this->widget('booster.widgets.TbTypeahead',array(
    'name' => 'demo-typeahead',
    'datasets' => array(
        'source' => array(
            'Alabama', 'Alaska', 'Arizona', 'Arkansas', 'California',
            'Colorado', 'Connecticut', 'Delaware', 'Florida', 'Georgia',
            'Hawaii', 'Idaho', 'Illinois', 'Indiana', 'Iowa', 'Kansas',
            'Kentucky', 'Louisiana', 'Maine', 'Maryland', 'Massachusetts',
            'Michigan', 'Minnesota', 'Mississippi', 'Missouri', 'Montana',
        ),
        ),
    'options' => array(
        'hint' => true,'highlight' => true,'minLength' => 1
    ),
),
));
);
```

```
$form->typeAheadGroup($model, 'name',array(
    'widgetOptions' => array(
        'options'=>array(
            'hint' => true,
            'highlight' => true,
            'minLength' => 1
        ),
        'datasets'=>compact('source')
    ),
    'labelOptions' => array(
        'label' => 'Chtulhu sleeps in:',
    )
));
});
```

**Yii Booster v.4.0.1 : Widget Time Picker**

```
$this->widget('booster.widgets.TbTimePicker',array(
    'name' => 'some_time',
    'wrapperHtmlOptions' => array('class' => 'col-md-3'),
));
```



```
$this->widget('booster.widgets.TbTimePicker',array(
    'name' => 'some_time','value' => '23:11',
    'noAppend' => true, // mandatory
    'options' => array(
        'disableFocus' => true, // mandatory
        'showMeridian' => false // irrelevant
    ),
    'wrapperHtmlOptions' => array('class' => 'col-md-3'),
));
```

**Yii Booster v.4.0.1 : Widget Date Picker**

```
$this->widget('booster.widgets.TbDatePicker',array(
    'name' => 'some_date',
    'htmlOptions' => array('class'=>'col-md-4'),
));
```



```
$this->widget('booster.widgets.TbDatePicker',array(
    'name' => 'some_date_jap',
    'options' => array('language' => 'ja')
));
```

## Yii Booster v.4.0.1 : Widget GridView

#	First name	Last name	Language	Hours worked	
1	Mark	Otto	CSS	10	 
2	Jacob	Thornton	JavaScript	20	 
3	Stu	Dent	HTML	10	 

```
$this->widget('booster.widgets.TbGridView',array(  
    'dataProvider'=>$dataProvider,  
    'template'=>"{items}",  

```

#	First name	Last name	Language	Hours worked	
1	Mark	Otto	CSS	10	 
2	Jacob	Thornton	JavaScript	20	 
3	Stu	Dent	HTML	10	 

```
$this->widget('booster.widgets.TbGridView',array(  
    'type' => 'striped',  
    'dataProvider'=>$dataProvider,  
    'template'=>"{items}",  

```

#	First name	Last name	Language	Hours worked	
1	Mark	Otto	CSS	10	 
2	Jacob	Thornton	JavaScript	20	 
3	Stu	Dent	HTML	10	 

```
$this->widget('booster.widgets.TbGridView',array(  
    'type' => 'bordered',  
    'dataProvider'=>$dataProvider,  
    'template'=>"{items}",  

```

#	First name	Last name	Language	Hours worked	
1	Mark	Otto	CSS	10	
2	Jacob	Thornton	JavaScript	20	
3	Stu	Dent	HTML	10	

```
$this->widget('booster.widgets.TbGridView', array(
    'type' => 'striped bordered condensed',
    'dataProvider' => $dataProvider,
    'template' => "{items}",
    'filter' => $model->search(),
));
```

### Yii Booster v.4.0.1 : Widget Panel

Basic Box

My Basic Content (you can use renderPartial here too :))

```
$this->widget('booster.widgets.TbPanel',array(
    'title' => 'Basic Box',
    'context' => 'primary',
    'headerIcon' => 'home',
    'content' => 'My Basic Content (you can use renderPartial here too :))'
));
```

Advanced Box

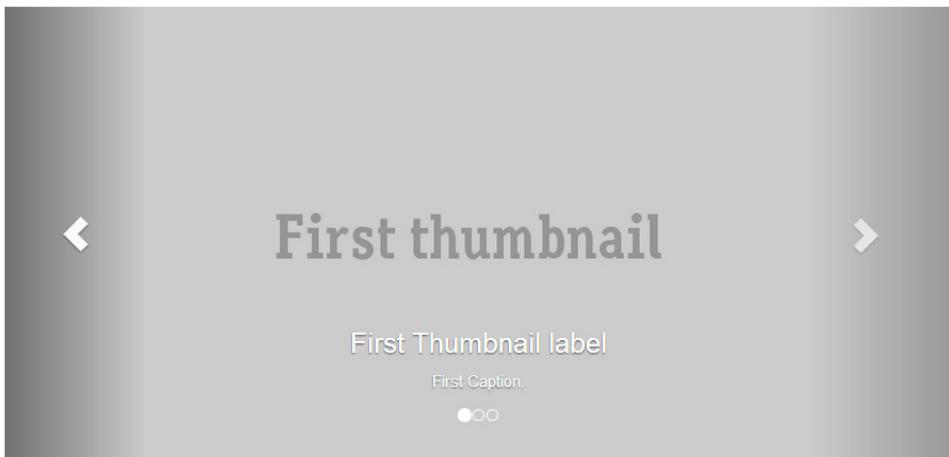
#	First name	Last name	Language	Hours worked
1	Mark	Otto	CSS	10
2	Jacob	Thornton	JavaScript	20
3	Stu	Dent	HTML	15

```
$this->beginWidget('booster.widgets.TbPanel',array(
    'title' => 'Advanced Box',
    'padContent' => false,
    'headerIcon' => 'th-list',
    'htmlOptions' => array('class' => 'bootstrap-widget-table')
));
```

```

<table class="table">
    <thead>
        <tr>
            <th>#</th>
            <th>First name</th>
            <th>Last name</th>
            <th>Language</th>
            <th>Hours worked</th>
        </tr>
    </thead>
    <tbody>
        <tr class="odd">
            <td style="width: 60px">1</td>
            <td>Mark</td>
            <td>Otto</td>
            <td>CSS</td>
            <td>10</td>
        </tr>
        <tr class="even">
            <td style="width: 60px">2</td>
            <td>Jacob</td>
            <td>Thornton</td>
            <td>JavaScript</td>
            <td>20</td>
        </tr>
        <tr class="odd">
            <td style="width: 60px">3</td>
            <td>Stu</td>
            <td>Dent</td>
            <td>HTML</td>
            <td>15</td>
        </tr>
    </tbody>
</table>
<?php $this->endWidget(); ?>

```

**Yii Booster v.4.0.1 : Widget Carousel**

```
$this->widget(
    'booster.widgets.TbCarousel',
    array(
        'items' => array(
            array(
                'image' => 'images/first-placeholder830x400.gif',
                'label' => 'First Thumbnail label',
                'caption' => 'First Caption.'
            ),
            array(
                'image' => 'images/second-placeholder830x400.gif',
                'label' => 'Second Thumbnail label',
                'caption' => 'Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi.'
            ),
            array(
                'image' => 'images/third-placeholder830x400.gif',
                'label' => 'Third Thumbnail label',
                'caption' => 'Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non m'
            ),
        ),
    )
);
```

## Yii Booster v.4.0.1 : Widget Collapse

### Collapsible Group Item #1

Anim pariatur cliche reprehenderit, enim eiusmod high life accusamus terry richardson ad squid. 3 wolf moon officia aute, non cupidatat skateboard dolor brunch. Food truck quinoa nesciunt laborum eiusmod. Brunch 3 wolf moon tempor, sunt aliqua put a bird on it squid single-origin coffee nulla assumenda shoreditch et. Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident. Ad vegan excepteur butcher vice lomo. Leggings occaecat craft beer farm-to-table, raw denim aesthetic synth nesciunt you probably haven't heard of them accusamus labore sustainable VHS.

### Collapsible Group Item #2

### Collapsible Group Item #3

```
<?php $collapse = $this->beginWidget('booster.widgets.TbCollapse'); ?>
<div class="panel-group" id="accordion">
<div class="panel panel-default">
<div class="panel-heading">
<h4 class="panel-title">
<a data-toggle="collapse" data-parent="#accordion" href="#collapseOne">Group Item #1</a>
</h4>
</div>
<div id="collapseOne" class="panel-collapse collapse in">
<div class="panel-body">
    Anim pariatur cliche reprehenderit, enim eiusmod high life accusamus terry richardson ad squid.
</div>
</div>
</div>
<div class="panel panel-default">
<div class="panel-heading">
<h4 class="panel-title">
<a data-toggle="collapse" data-parent="#accordion" href="#collapseTwo">Group Item #2</a>
</h4>
</div>
<div id="collapseTwo" class="panel-collapse collapse">
<div class="panel-body">
    Anim pariatur cliche reprehenderit, enim eiusmod high life accusamus terry richardson ad squid.
</div>
</div>
</div>
</div>
<?php $this->endWidget(); ?>
```

## Yii Booster v.4.0.1 : Widget Detail View



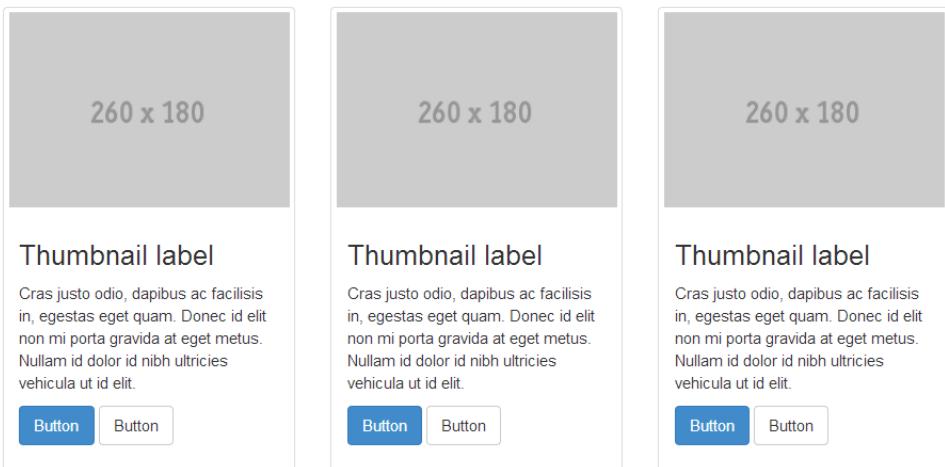
```
$this->widget(
    'booster.widgets.TbDetailView',
    array(
        'data' => array(
            'id' => 1,
            'firstName' => 'Mark',
            'lastName' => 'Otto',
            'language' => 'CSS'
        ),
        'attributes' => array(
            array('name' => 'firstName', 'label' => 'First name'),
            array('name' => 'lastName', 'label' => 'Last name'),
            array('name' => 'language', 'label' => 'Language'),
        ),
    )
);
```

## Yii Booster v.4.0.1 : Widget Tabs



```
$this->widget('booster.widgets.TbTabs',array(
    'type' => 'tabs', // 'tabs' or 'pills'
    'tabs' => array(
        array(
            'label' => 'Home',
            'content' => 'Home Content',
            'active' => true
        ),
        array('label' => 'Profile', 'content' => 'Profile Content'),
        array('label' => 'Messages', 'content' => 'Messages Content'),
    ),
));
```

## Yii Booster v.4.0.1 : Widget Thumbnails



```
echo CHtml::openTag('div', array('class' => 'row-fluid'));
$this->widget(
    'booster.widgets.TbThumbnails',
    array(
        'dataProvider' => $thumbDataProvider,
        'template' => "{items}\n{pager}",
        'itemView' => 'application.views._thumb',
    )
);
echo CHtml::closeTag('div');
```

Here is the content of \_thumb.php template:

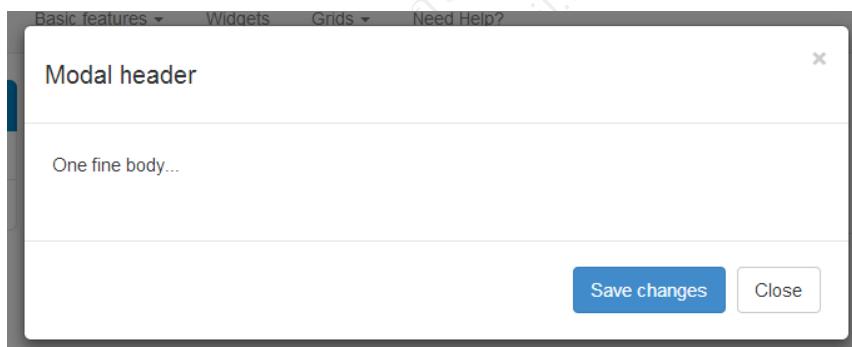
```
<div class="col-sm-6 col-md-4">
    <div class="thumbnail">
        <img src=<?php echo bu('images/placeholder260x180.gif');?>" alt="">
        <div class="caption">
            <h3>Thumbnail label</h3>
            <p>Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.</p>
            <p><a href="#" class="btn btn-primary" role="button">Button</a> <a href="#" class="btn btn-default" role="button">Button</a></p>
        </div>
    </div>
</div>
```

## Yii Booster v.4.0.1 : Widget Wizard



```
$this->widget('booster.widgets.TbWizard',array(
    'type' => 'tabs', // 'tabs' or 'pills'
    'tabs' => array(
        array(
            'label' => 'Home','content' => 'Home Content','active' => true
        ),
        array('label' => 'Profile','content' => 'Profile Content'),
        array('label' => 'Messages','content' => 'Messages Content'),
    ),
));
```

## Yii Booster v.4.0.1 : Widget Modals



```
<?php $this->beginWidget(
    'booster.widgets.TbModal',
    array('id' => 'myModal')
); ?>


<a class="close" data-dismiss="modal">&times;</a>
    <h4>Modal header</h4>



<p>One fine body...</p>


```

```

<div class="modal-footer">
    <?php $this->widget('booster.widgets.TbButton',array(
        'context' => 'primary',
        'label' => 'Save changes',
        'url' => '#',
        'htmlOptions' => array('data-dismiss' => 'modal'),
    )); ?>
    <?php $this->widget('booster.widgets.TbButton',array(
        'label' => 'Close',
        'url' => '#',
        'htmlOptions' => array('data-dismiss' => 'modal'),
    )); ?>
</div>
<?php $this->endWidget(); ?>
<?php $this->widget('booster.widgets.TbButton',array(
    'label' => 'Click me',
    'context' => 'primary',
    'htmlOptions' => array(
        'data-toggle' => 'modal',
        'data-target' => '#myModal',
    ),
)); ;

```

## Yii Booster v.4.0.1 : Widget Breadcrumbs

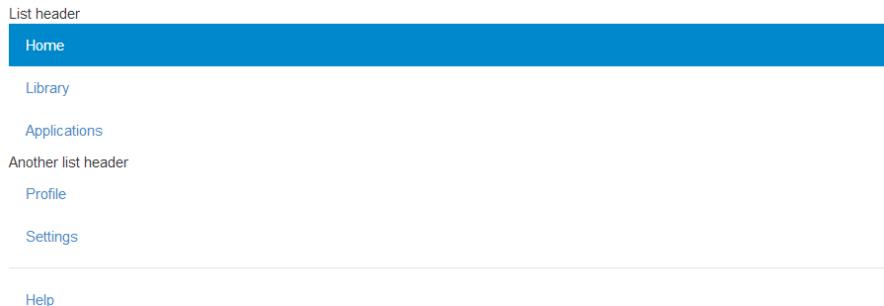


```

$this->widget('booster.widgets.TbBreadcrumbs',array(
    'homeLink' => 'Home',
    'links' => array('')
));
$this->widget('booster.widgets.TbBreadcrumbs',array(
    'links' => array('Library'),
));
$this->widget('booster.widgets.TbBreadcrumbs',array(
    'links' => array('Library' => '#', 'Data'),
));

```

## Yii Booster v.4.0.1 : Widget Menu



```
$this->widget('booster\widgets\TbMenu',array(
    'type' => 'list',
    'items' => array(
        array(
            'label' => 'List header',
            'itemOptions' => array('class' => 'nav-header')
        ),
        array(
            'label' => 'Home',
            'url' => '#',
            'itemOptions' => array('class' => 'active')
        ),
        array('label' => 'Library', 'url' => '#'),
        array('label' => 'Applications', 'url' => '#'),
        array(
            'label' => 'Another list header',
            'itemOptions' => array('class' => 'nav-header')
        ),
        array('label' => 'Profile', 'url' => '#'),
        array('label' => 'Settings', 'url' => '#'),
        array('label' => 'Help', 'url' => '#'),
    )
));
});
```

## Yii Booster v.4.0.1 : Widget Navigation Bar

```
Title Home Link Link
```

```
$this->widget('booster.widgets.TbNavbar',array(  
    'brand' => 'Title',  
    'fixed' => false,  
    'fluid' => true,  
    'items' => array(  
        array(  
            'class' => 'booster.widgets.TbMenu',  
            'type' => 'navbar',  
            'items' => array(  
                array('label' => 'Home', 'url' => '#', 'active' => true),  
                array('label' => 'Link', 'url' => '#'),  
                array('label' => 'Link', 'url' => '#'),  
            )  
        )  
    )  
);
```

```
Title  Submit
```

```
$this->widget(  
    'booster.widgets.TbNavbar',  
    array(  
        'brand' => 'Title',  
        'fixed' => false,  
        'fluid' => true,  
        'items' => array(  
            '<form class="navbar-form navbar-left">  
                <div class="form-group">  
                    <input type="text" class="form-control">  
                </div>  
                <button type="submit" class="btn btn-default">Submit</button>  
            </form>'  
        )  
    );
```

## บรรณานุกรม

\_\_\_\_\_. กำหนด Attribute ให้กับเมธอด Rules ในโมเดล (ออนไลน์)  
แหล่งที่มา : <http://www.yii.in.th/forum/>. 8 ธันวาคม 2556

ก.เกม. การเชื่อมต่อฐานข้อมูลแบบ Multiple-database ที่สนับสนุน Yii Framework  
แหล่งที่มา : <http://blog.palaloy.com/2013/06/yii-9-multiple-database-yii-framework.html>.  
16 ธันวาคม 2556

Administrator. อธิบาย Relation ต่างๆใน Active Record (ออนไลน์)  
แหล่งที่มา : <http://www.yii.in.th/forum/index.php?topic=420.0>. 5 มกราคม 2557

Administrator. วิธีติดตั้ง Rights เพื่อช่วยในการ Access Control ต่างๆ (ออนไลน์)  
แหล่งที่มา : <http://www.yii.in.th/forum/index.php?topic=324.0>. 5 มกราคม 2557



การพัฒนาเว็บแอปพลิเคชัน



กม.  
fenics1281





CR.  
EAKKABIN  
JAIKEAWMA