

มาตรฐานการพัฒนาซอฟต์แวร์

Software Development Standard

ฝ่ายบริหารจัดการคุณภาพ (Quality Management)

ห้องปฏิบัติการวิจัยวิศวกรรมระบบสารสนเทศ

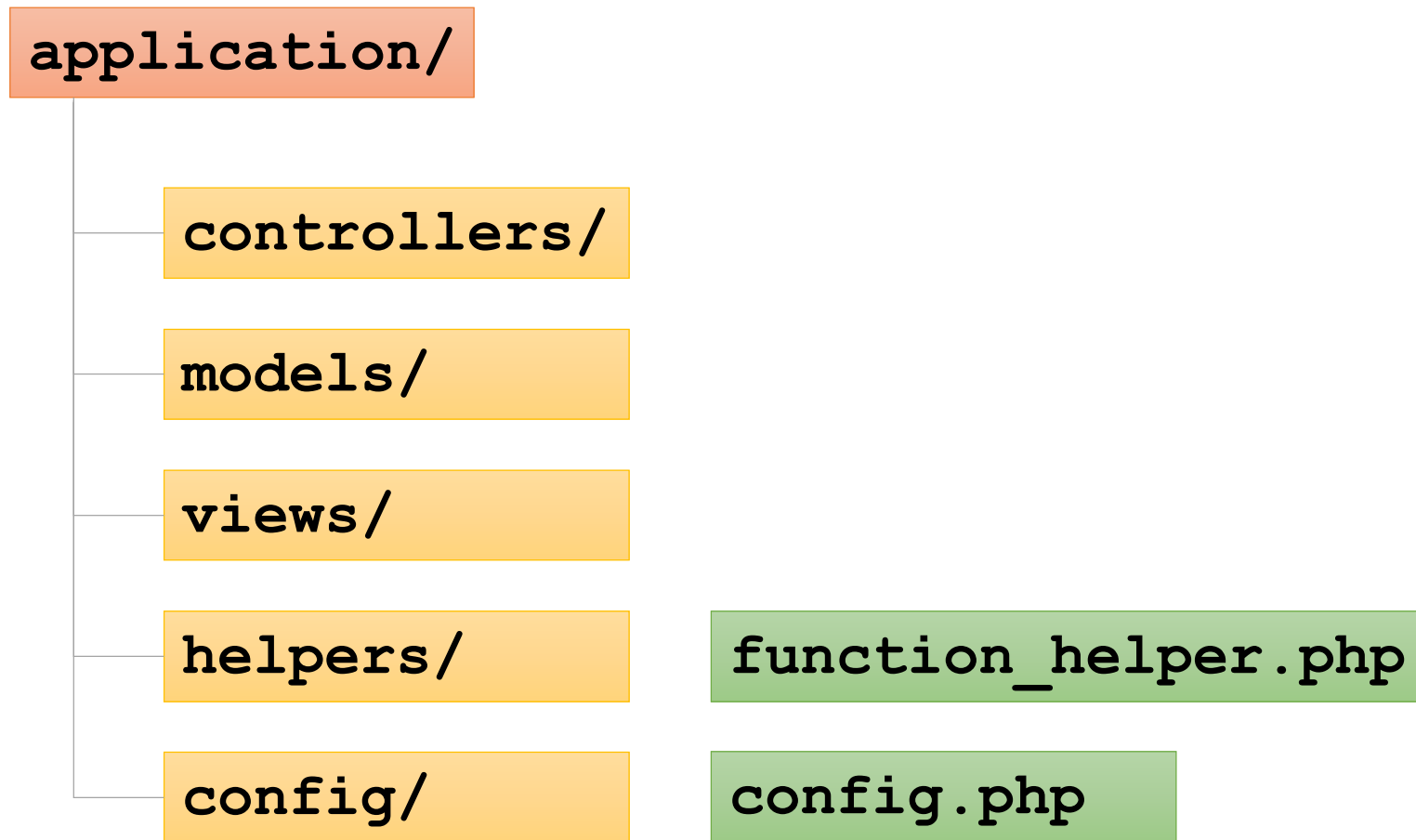
มาตรฐานการพัฒนาซอฟต์แวร์ (SDS)

- มาตรฐานการเขียนโปรแกรม (Coding Standard)
- มาตรฐานส่วนติดต่อผู้ใช้ (UI Standard)

มาตรฐานการเขียนโปรแกรม (Coding Standard)

1. การตั้งชื่อไฟล์และคลาส
2. การตั้งชื่อฟังก์ชัน
3. การตั้งชื่อตัวแปร
4. การจัดทำมาตรฐานเกี่ยวกับฐานข้อมูล
5. การตั้งชื่อตัวแปรของ Config
6. การตั้งชื่อฟังก์ชันของ Helper
7. การเขียน Comments

โครงสร้างไดเรกทอรี



1. การตั้งชื่อไฟล์และคลาส

1.1 การตั้งชื่อไฟล์และคลาสของ Controller

1.2 การตั้งชื่อไฟล์และคลาสของ Model

1.3 การตั้งชื่อไฟล์ View

1.1 การตั้งชื่อไฟล์และคลาสของ Controller

- ตั้งชื่อไฟล์ด้วยตัวอักษรภาษาอังกฤษตัวแรกเป็นตัวใหญ่จากนั้นพิมพ์เล็กทั้งหมด และตัวอักษรติดกันทั้งหมด
 - กรณีชื่อไฟล์ หรือคลาสมีจำนวนคำมากกว่า 1 คำ คั่นคำด้วยเครื่องหมายขีดล่าง (_) ได้
 - กรณีชื่อไฟล์ หรือคลาสต้องระบุตัวเลข ระบุตัวเลขอยู่ตำแหน่งท้ายสุด
- ชื่อคลาสเป็นชื่อเดียวกันกับชื่อไฟล์ และขึ้นต้นด้วยตัวอักษรภาษาอังกฤษพิมพ์ใหญ่
- ตั้งชื่อไฟล์ หรือคลาสตามโมดูลของระบบ

Person.php

```
class Person extends CI_Controller{  
    ...  
}
```

Person_management.php

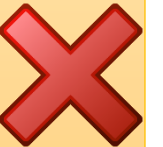
```
class Person_management extends CI_Controller{  
    ...  
}
```

1.1 การตั้งชื่อไฟล์และคลาสของ Controller (ต่อ)

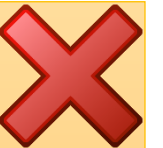
- คอนโทรลเลอร์หลักของระบบ ตั้งชื่อตามรูปแบบ ชื่อระบบ_controller
- กรณีไม่ใช่คอนโทรลเลอร์หลักของระบบ ไม่ตั้งชื่อไฟล์ขึ้นต้นด้วยชื่อระบบ
- ไม่ตั้งชื่อไฟล์ขึ้นต้นด้วยคำ เช่น c_, con_, controller_

```
Hr_controller.php  
class Hr_controller extends CI_Controller{  
    ...  
}
```

```
Hr_person.php  
class hr_person extends CI_Controller(){  
    ...  
}
```



```
C_person.php  
class C_person extends CI_Controller(){  
    ...  
}
```



1.1 การตั้งชื่อไฟล์และคลาสของ Controller (ต่อ)

โครงสร้างไดเรกทอรีของ Controller

แบบที่ 1 controllers/ชื่อระบบ/ชื่อคอนโทรลเลอร์.php

แบบที่ 2 controllers/ชื่อระบบ/ชื่อโมดูลหรืองาน/ชื่อ
คอนโทรลเลอร์.php

1.1 การตั้งชื่อไฟล์และคลาสของ Controller (ต่อ)

คอนโทรลเลอร์สำหรับข้อมูลพื้นฐานของระบบ

แบบที่ 1

- ตั้งชื่อไฟล์หรือคลาส ลงท้ายด้วย `_base`

```
Position_base.php  
class Position_base extends CI_Controller{  
    ...  
}
```

แบบที่ 2

- ตั้งชื่อไฟล์เดอร์ว่า `base`
- ตั้งชื่อไฟล์หรือคลาส โดยไม่มี `_base`

```
base/Position.php  
class Position extends CI_Controller{  
    ...  
}
```

```
base/Education.php  
class Education extends CI_Controller{  
    ...  
}
```

1.1 การตั้งชื่อไฟล์และคลาสของ Controller (ต่อ)

คอนโทรลเลอร์สำหรับรายงานของระบบ

แบบที่ 1

- ตั้งชื่อไฟล์หรือคลาส ลงท้ายด้วย
_report

```
Salary_report.php  
class salary_report extends CI_Controller{  
    ...  
}
```

แบบที่ 2

- ตั้งชื่อไฟล์เดอร์ว่า report
- ตั้งชื่อไฟล์หรือคลาส โดยไม่มี _report

```
report/Salary.php  
class salary extends CI_Controller{  
    ...  
}
```

```
report/Worktime.php  
class Worktime extends CI_Controller{  
    ...  
}
```

1.1 การตั้งชื่อไฟล์และคลาสของ Controller (ต่อ)

คอนโทรลเลอร์สำหรับเชื่อมโยงข้อมูลกับระบบสารสนเทศอื่น

- แบบที่ 1 ชื่อระบบหรือชื่องาน_service
- แบบที่ 2 service/ชื่อระบบหรือชื่องาน

Person_service.php

```
class Person_service extends CI_Controller{  
    ...  
}
```

คอนโทรลเลอร์สำหรับรับ-ส่งค่าในรูปแบบ AJAX

- แบบที่ 1 ชื่อระบบหรือชื่องาน_ajax
- แบบที่ 2 ajax/ชื่อระบบหรือชื่องาน

Person_ajax.php

```
class Person_ajax extends CI_Controller{  
    ...  
}
```

1.2 การตั้งชื่อไฟล์และคลาสของ Model

- โมเดลประกอบด้วย 2 ไฟล์ คือ Da และ M
 - กรณีไฟล์ Da ตั้งชื่อขึ้นต้นด้วย Da_
 - กรณีไฟล์ M ตั้งชื่อขึ้นต้นด้วย M_
- ตั้งชื่อไฟล์ด้วยตัวอักษรภาษาอังกฤษตัวแรกพิมพ์ใหญ่เท่านั้น และตัวหนังสือที่เหลือพิมพ์เล็กเท่านั้น ตัวอักษรติดกันทั้งหมด และคั่นคำด้วยเครื่องหมายขีดล่าง (_) ได้
- ชื่อคลาสเป็นชื่อเดียวกันกับชื่อไฟล์ และขึ้นต้นด้วยตัวอักษรภาษาอังกฤษพิมพ์ใหญ่
- ตั้งชื่อไฟล์ หรือคลาสตามชื่อตารางใน

Da_hr_position.php

```
class Da_hr_position extends CI_Model{  
    ...  
}
```

M_hr_position.php

```
class M_hr_position extends Da_hr_position{  
    ...  
}
```

1.2 การตั้งชื่อไฟล์และคลาสของ Model (ต่อ)

- โมเดลหลักของระบบ ตั้งชื่อตามรูปแบบชื่อระบบ_model

Hr_model.php

```
class Hr_model extends CI_Model{  
    ...  
}
```

1.3 การตั้งชื่อไฟล์ View

- ชื่อไฟล์ขึ้นต้นด้วย v_
- ตั้งชื่อไฟล์ด้วยตัวอักษรภาษาอังกฤษพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง (_) ได้
- ตั้งชื่อไฟล์วิวสอดคล้องกับชื่อฟังก์ชันที่เรียกใช้วิว
 - กรณีฟังก์ชันเรียกวิว มากกว่า 1 วิว ตั้งชื่อไฟล์ วิวขยายจากชื่อฟังก์ชันที่เรียกใช้วิว
 - กรณีไฟล์วิว ถูกเรียกจากหลายฟังก์ชัน ตั้งชื่อไฟล์ตามชื่อฟังก์ชันแรกที่ใช้วิว

```
class Position extends CI_Controller{  
    ...  
    function position_input() {  
        ...  
        $this->load->view('v_position_input', $data);  
        ...  
        $this->load->view('v_position_input_2', $data);  
    }  
  
    function position_edit() {  
        ...  
        $this->load->view('v_position_input', $data);  
    }  
}
```

2. การตั้งชื่อฟังก์ชัน

2.1 การตั้งชื่อฟังก์ชันของ Controller

2.2 การตั้งชื่อฟังก์ชันของ Model

2.1 การตั้งชื่อฟังก์ชันของ Controller

- ตั้งชื่อไฟล์ด้วยตัวอักษรภาษาอังกฤษพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง (_) ได้
- ประเภทของฟังก์ชัน มีดังนี้
 - ฟังก์ชันสำหรับแสดงหน้าจอบันทึก หรือแก้ไขข้อมูล
 - ฟังก์ชันสำหรับบันทึก หรือแก้ไขฐานข้อมูล
 - ฟังก์ชันสำหรับลบข้อมูลในฐานข้อมูล
 - ฟังก์ชันสำหรับแสดงผล
 - ฟังก์ชันสำหรับนำเข้า และอ่านข้อมูลจากไฟล์ (ไฟล์ excel)
 - ฟังก์ชันสำหรับส่งออกข้อมูลในรูปแบบต่างๆ
 - ฟังก์ชันสำหรับแสดงหน้าต่างป๊อปอัพ
 - ฟังก์ชันสำหรับเชื่อมโยงข้อมูลกับระบบสารสนเทศอื่น
 - ฟังก์ชันสำหรับรับ-ส่งค่าในรูปแบบ AJAX

ฟังก์ชันสำหรับแสดงหน้าจอการบันทึก หรือแก้ไขข้อมูล

- แสดงหน้าจอบันทึกข้อมูลอย่างเดียว
หรือทั้งบันทึก และแก้ไขข้อมูล
 - ตั้งชื่อลงท้ายด้วย `_input` หรือ
 - ตั้งชื่อด้วย `input`
- แสดงหน้าจอแก้ไขข้อมูลอย่างเดียว
 - ตั้งชื่อลงท้ายด้วย `_edit` หรือ
 - ตั้งชื่อด้วย `edit`

```
class Position extends CI_Controller{  
    ...  
    function input() {  
        ...  
    }  
    ...  
    function position_input() {  
        ...  
    }  
}
```

ฟังก์ชันสำหรับบันทึก หรือแก้ไขฐานข้อมูล

- บันทึกอย่างเดียว
 - ตั้งชื่อลงท้ายด้วย `_insert` หรือ
 - ตั้งชื่อด้วย `insert`
- แก้ไขอย่างเดียว
 - ตั้งชื่อลงท้ายด้วย `_update` หรือ
 - ตั้งชื่อด้วย `update`
- ทั้งบันทึก และแก้ไข
 - ตั้งชื่อลงท้ายด้วย `_save` หรือ
 - ตั้งชื่อด้วย `save`

```
class Position extends CI_Controller{  
    ...  
    function insert() {  
        ...  
    }  
    ...  
    function position_insert() {  
        ...  
    }  
}
```

ฟังก์ชันสำหรับลบข้อมูลในฐานข้อมูล

- ตั้งชื่อลงท้ายด้วย `_delete` หรือ
- ตั้งชื่อด้วย `delete`

```
class Position extends CI_Controller{  
    ...  
    function delete() {  
        ...  
    }  
    ...  
    function position_delete() {  
        ...  
    }  
}
```

ฟังก์ชันสำหรับแสดงผล

- หน้าหลักแสดงข้อมูล
 - ตั้งชื่อลงท้ายด้วย `_show` หรือ
 - ตั้งชื่อด้วย `show`
- หน้าแสดงข้อมูลแบบลงรายละเอียด
 - ตั้งชื่อลงท้ายด้วย `_detail` หรือ
 - ตั้งชื่อด้วย `detail`

```
class Position extends CI_Controller{  
    ...  
    function show() {  
        ...  
    }  
    ...  
    function position_show() {  
        ...  
    }  
}
```

ฟังก์ชันสำหรับนำเข้า และอ่านข้อมูลจากไฟล์ (ไฟล์ excel)

- ตั้งชื่อลงท้ายด้วย `_import` หรือ
- ตั้งชื่อด้วย `import`

```
class Position extends CI_Controller{  
    ...  
    function import() {  
        ...  
    }  
    ...  
    function position_import() {  
        ...  
    }  
}
```

ฟังก์ชันสำหรับส่งออกข้อมูลในรูปแบบต่างๆ

- ส่งออกข้อมูลรูปแบบไฟล์ Excel
 - ตั้งชื่อด้วย excel หรือลงท้ายด้วย _excel
- ส่งออกข้อมูลรูปแบบไฟล์ Word
 - ตั้งชื่อด้วย word หรือลงท้ายด้วย _word
- ส่งออกข้อมูลรูปแบบไฟล์ PDF
 - ตั้งชื่อด้วย pdf หรือลงท้ายด้วย _pdf
- ส่งออกข้อมูลรูปแบบตัวอย่างก่อนพิมพ์
 - ตั้งชื่อด้วย print หรือลงท้ายด้วย _print
- ส่งออกข้อมูลหลายรูปแบบในฟังก์ชันเดียว
 - ตั้งชื่อด้วย export หรือลงท้ายด้วย _export

```
class Position extends CI_Controller{  
  
    ...  
    function excel() {  
  
        ...  
    }  
  
    ...  
    function position_excel() {  
  
        ...  
    }  
  
}
```

ฟังก์ชันสำหรับแสดงหน้าต่างป๊อปอัพ

- ตั้งชื่อลงท้ายด้วย `_popup` หรือ
- ตั้งชื่อด้วย `popup`

```
class Position extends CI_Controller{  
    ...  
    function popup() {  
        ...  
    }  
    ...  
    function position_popup() {  
        ...  
    }  
}
```

ฟังก์ชันสำหรับเชื่อมโยงข้อมูลกับระบบสารสนเทศอื่น

- รับข้อมูล

- ตั้งชื่อขึ้นต้นด้วย `get_service_` หรือ
- ตั้งชื่อด้วย `get_service`

- ส่งข้อมูล

- ตั้งชื่อขึ้นต้นด้วย `post_service_` หรือ
- ตั้งชื่อด้วย `post_service`

```
class Position extends CI_Controller{  
    ...  
    function get_service() {  
        ...  
    }  
    ...  
    function get_service_position() {  
        ...  
    }  
}
```


ฟังก์ชันสำหรับรับ-ส่งค่าในรูปแบบ AJAX

- ตั้งชื่อลงท้ายด้วย `_ajax` หรือ
- ตั้งชื่อด้วย `ajax`

```
class Position extends CI_Controller{  
    ...  
    function ajax() {  
        ...  
    }  
    ...  
    function position_ajax() {  
        ...  
    }  
}
```

2.2 การตั้งชื่อฟังก์ชันของ Model

ฟังก์ชันในไฟล์ da

- ประกอบด้วยฟังก์ชันหลัก 4 ฟังก์ชัน ได้แก่
 - insert()
 - update()
 - delete()
 - และ get_by_key() เท่านั้น

```
class Da_hr_person extends CI_Model {  
    ...  
    function insert() {  
        ...  
    }  
    function update() {  
        ...  
    }  
    function delete() {  
        ...  
    }  
    function get_by_key() {  
        ...  
    }  
}
```

2.2 การตั้งชื่อฟังก์ชันของ Model (ต่อ)

ฟังก์ชันในไฟล์ m

- ประกอบด้วยฟังก์ชันอื่นๆ นอกเหนือจาก 4 ฟังก์ชันหลัก
- ตั้งชื่อฟังก์ชันด้วยตัวอักษรภาษาอังกฤษพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง (_)

2.2 การตั้งชื่อฟังก์ชันของ Model (ต่อ)

ฟังก์ชันในไฟล์ m

- ตั้งชื่อฟังก์ชันตามรูปแบบ ดังนี้
 - `action_data_by_condition()` หรือ
 - `action_data_for_something()`
- `action` คือ การกระทำ
- `data` คือ ข้อมูลที่ต้องการ
- `by_condition` คือ เงื่อนไขการค้นหา
- `for_something` คือ ถูกเรียกใช้เพื่อฟังก์ชันโมดูล หรือเงื่อนไขโดยเฉพาะ (ถ้าสำคัญ)

```
class M_hr_person extends Da_hr_position{  
    ...  
    function get_all() {  
        ...  
    }  
    function get_person_name_by_id() {  
        ...  
    }  
    function get_person_name_for_ajax() {  
        ...  
    }  
}
```

2.2 การตั้งชื่อฟังก์ชันของ Model (ต่อ)

ประเภทของฟังก์ชัน มีดังนี้

- ฟังก์ชันสำหรับคิวรี เพื่อดึงข้อมูล
- ฟังก์ชันสำหรับคิวรี โดยใช้ SQL function
- ฟังก์ชันสำหรับคิวรี เพื่อตรวจสอบข้อมูล
- ฟังก์ชันสำหรับอัปเดตบางฟิลด์
- ฟังก์ชันสำหรับลบ โดยไม่อ้างอิง PK

ฟังก์ชันสำหรับควรี เพื่อดึงข้อมูล

- ดึงข้อมูลทั่วไปที่ไม่มีการค้นหา หรือมีการค้นหาแบบมีเงื่อนไขไม่ซับซ้อน
 - ตั้งชื่อขึ้นต้นด้วย `get_`
- ดึงข้อมูลที่มีการค้นหาแบบมีเงื่อนไขซับซ้อน
 - ตั้งชื่อขึ้นต้นด้วย `search_`

```
class M_hr_person extends Da_hr_person{  
    ...  
    function get_all() {  
        ...  
    }  
    function get_person_by_pos_id() {  
        ...  
    }  
    function search_person_for_report() {  
        ...  
    }  
}
```

ฟังก์ชันสำหรับคิวรี โดยใช้ SQL function

- ตั้งชื่อขึ้นต้นด้วย SQL function
เช่น count_, sum_, max_,
min_ เป็นต้น

```
class M_hr_person extends Da_hr_person{  
    ...  
    function count_person() {  
        $sql = 'SELECT count(*)  
            ...'  
    }  
    function max_salary() {  
        $sql = 'SELECT max(ps_salary)  
            ...'  
    }  
}
```

ฟังก์ชันสำหรับคิวรี เพื่อตรวจสอบข้อมูล

- คำนวณค่าข้อมูลเป็นไบนารี เช่น 0 หรือ 1, true หรือ false, y หรือ n เป็นต้น
 - ตั้งชื่อขึ้นต้นด้วย check_

```
class M_hr_person extends Da_hr_person{  
    ...  
    function check_active_person() {  
        ...  
        if(...)   
            return 0  
        else  
            return 1  
    }  
}
```


ฟังก์ชันสำหรับอัปเดตบางฟิลด์

- กรณีอัปเดต 1-2 ฟิลด์
 - ตั้งชื่อตามรูปแบบ update_ชื่อฟิลด์ที่ต้องการอัปเดต
- กรณีอัปเดตมากกว่า 2 ฟิลด์
 - ตั้งชื่อตามรูปแบบ update_ชื่อการทำงาน

```
class M_hr_person extends Da_hr_person{  
    ...  
    function update_ps_prefix_and_ps_name() {  
        ...  
    }  
    function update_person_retire() {  
        ...  
    }  
}
```

ฟังก์ชันสำหรับลบ โดยไม่อ้างอิง PK

- ตั้งชื่อตามรูปแบบ delete_by_ชื่อฟิลด์

```
class M_hr_person extends Da_hr_person{  
    ...  
    function delete_by_pos_id() {  
        ...  
    }  
}
```

3. การตั้งชื่อตัวแปร

3.1 ตัวแปรแทน Object ของ Model

3.2 ตัวแปรที่รับค่ามาจากฐานข้อมูล

3.3 ตัวแปรสำหรับรับค่าจาก Fetch Array และ Object

3.4 ตัวแปร Array

3.5 ตัวแปร JSON

3.6 ตัวแปรนับรอบของลูป

3.1 ตัวแปรแทน Object ของ Model

- ตั้งชื่อขึ้นต้นด้วย \$m ตามด้วยชื่อย่อของตาราง

```
$this->load->model( 'm_hr_person' , 'mps' );  
$this->mps->get_all();
```

```
$mps = $this->mps;  
$mps->get_all();
```

3.2 ตัวแปรที่รับค่ามาจากฐานข้อมูล

- ตั้งชื่อตัวแปรด้วยตัวอักษรภาษาอังกฤษพิมพ์เล็กเท่านั้น และค้นค่าด้วยเครื่องหมายขีดล่าง (_) ได้

```
$rs_person = $this->mps->get_all();  
$data['rs_person'] = $this->mps->get_all();
```

- กรณีรับค่าหลายเรคคอร์ด
 - ตั้งชื่อตามรูปแบบ \$rs_ชื่อย่อหรือชื่อเต็มของข้อมูล

```
$qu_person = $this->mps->get_by_key();  
$data['qu_person'] = $this->mps->get_by_key();  
$qu_person->row()->ps_fname;  
$qu_person->row()->ps_lname;
```

- กรณีรับค่าเรคคอร์ดเดียว
 - ตั้งชื่อตามรูปแบบ \$qu_ชื่อย่อหรือชื่อเต็มของข้อมูล

3.2 ตัวแปรที่รับค่ามาจากฐานข้อมูล (ต่อ)

- กรณีรับค่าฟิลด์เดียว หรือรับค่าจาก SQL function

- ตั้งชื่อด้วยคำที่สื่อความหมาย

```
$num_person = $this->mps->count_person();  
$num_person->row()->num_person;  
  
$sum_salary = $this->mps->sum_salary();  
$sum_salary->row()->sum_salary;
```

- กรณีรับค่าเป็นอาร์เรย์สำหรับ Drop down list

- ตั้งชื่อขึ้นต้นด้วย \$opt_

```
$opt_person = $this->mps->get_options();  
  
form_dropdown('person', $opt_person, '');
```

3.3 ตัวแปรสำหรับรับค่าจาก Fetch Array และ Object

- ตั้งชื่อตัวแปรด้วยตัวอักษรภาษาอังกฤษพิมพ์เล็กเท่านั้น และค้นค่าด้วยเครื่องหมายชี้ด้านล่าง (_) ได้
- กรณีรับค่าจากการ Fetch Array
 - ค่า Key ตั้งชื่อตามรูปแบบ `$key_ชื่อข้อมูล`
 - ค่า Value ตั้งชื่อตามรูปแบบ `$val_ชื่อข้อมูล`
- กรณีรับค่าจากการ Fetch Object
 - ตั้งชื่อตามรูปแบบ `$row_ชื่อข้อมูล`

```
$arr_status = array( '0' => 'available',  
                    '1' => 'unavailable',  
                    );  
foreach($arr_status as $key_sts => $val_sts){  
    $key_sts; //0, 1  
    $val_sts; //available, unavailable  
}
```

```
foreach($rs_person->result() as $row_ps){  
    $row_ps->ps_fname;  
    $row_ps->ps_lname;  
    ...  
}
```

3.4 ตัวแปร Array

- กรณีข้อมูลเป็นอาร์เรย์
 - ตั้งชื่อตัวแปรขึ้นต้นด้วย
\$arr_

```
$arr_status = array( '0' => 'available',  
                    '1' => 'unavailable',  
                    );
```

```
$arr_status = array('available', 'unavailable');
```

```
...  
<input type="checkbox" name="position[]" value="1">  
<input type="checkbox" name="position[]" value="2">  
<input type="checkbox" name="position[]" value="3">  
...  
$arr_position = $this->input->post('position');
```


3.5 ตัวแปร JSON

- กรณีข้อมูลเป็น JSON
 - ตั้งชื่อตัวแปรขึ้นต้นด้วย `$json_`

```
...  
$data['json_data'] = ...;  
$data['json_message'] = "complete";  
echo json_encode($data);  
...
```

3.6 ตัวแปรนับรอบของลูป

- กรณีตั้งชื่อตัวแปรเพื่อบรรอบของลูปแบบตัวอักษรเดียว ตั้งชื่อด้วยตัวอักษรที่สื่อความหมาย
- ตัวอย่างเช่น
 - กรณีนับรอบของลูป 1 ชั้น ตั้งชื่อด้วย \$i
 - กรณีนับรอบของลูปซ้อน 2-3 ชั้น ตั้งชื่อด้วย \$i, \$j, \$k หรือ \$m, \$n หรือ \$x, \$y, \$z ร่วมกัน

```
$i=0;
while ($i<100) {
    $j=0;
    while ($j<100) {
        ...
        $j++;
    }
    $i++;
}
```

4. การจัดทำมาตรฐานเกี่ยวกับฐานข้อมูล

4.1 การตั้งชื่อฐานข้อมูล

4.2 การตั้งชื่อตาราง

4.3 การตั้งชื่อฟิลด์

4.4 การเขียน Comment ของตารางและฟิลด์

4.1 การตั้งชื่อฐานข้อมูล

- ตั้งชื่อด้วยตัวอักษรภาษาอังกฤษพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง (_)
- ตั้งชื่อตามรูปแบบ ชื่อย่อของไซต์_ชื่อหรือชื่อย่อของระบบdb (ลงท้ายด้วย db)

SMEs Site

> sme_hrdb

SCPHKK Site

> kk_hrdb

4.2 การตั้งชื่อตาราง

- ตั้งชื่อด้วยตัวอักษรภาษาอังกฤษพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง (_)
- ตั้งชื่อตามรูปแบบ ชื่อย่อของระบบ_ชื่อโมดูลการทำงานหรือชื่อที่บ่งบอกข้อมูลที่เก็บ
- ชื่อย่อของระบบ ความยาวไม่เกิน 4 ตัวอักษร

sme_hrdb

```
> hr_person  
> hr_position  
> hr_education  
> hr_department
```

4.3 การตั้งชื่อไฟล์

- ตั้งชื่อด้วยตัวอักษรภาษาอังกฤษพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง (_)
- ตั้งชื่อตามรูปแบบ ชื่อย่อของตาราง_ชื่อไฟล์
- ชื่อย่อของตาราง ความยาวไม่เกิน 4 ตัวอักษร หรือเกิน 4 ตัวอักษร ในกรณีที่ไม่สามารถลดคำให้น้อยลงได้

4.3 การตั้งชื่อฟิลด์ (ต่อ)

- ชื่อฟิลด์ที่เป็น PK ตั้งชื่อฟิลด์เป็น id
- ชื่อฟิลด์ที่เป็นความหมายหรือข้อมูลหลักของตาราง ตั้งชื่อฟิลด์เป็น name
- ชื่อฟิลด์ที่บ่งบอกลำดับของข้อมูล ตั้งชื่อฟิลด์เป็น seq
- ชื่อฟิลด์ที่บ่งบอกลำดับชั้น ตั้งชื่อฟิลด์เป็น level
- ชื่อฟิลด์ที่อ้างอิง PK ของตารางอื่น (ฟิลด์ที่เป็น FK) ตั้งชื่อฟิลด์เป็น PK ของตารางที่อ้างอิง
- ชื่อฟิลด์ที่อ้างอิง PK ของตารางตนเอง ในลักษณะความสัมพันธ์แบบแม่-ลูก ตั้งชื่อฟิลด์เป็น parent_id

ps_id

ps_name, pos_name, edu_name

dp_seq

dp_level

ps_pos_id, ps_dp_id

dp_parent_id

4.4 การเขียน Comment ของตารางและฟิลต์

- คอมเมนต์ หรือนิยามความหมายกำกับไว้ทุกตาราง และทุกฟิลต์
- ตาราง คอมเมนต์ว่าเก็บข้อมูลอะไร หรือใช้สำหรับทำอะไร
- ฟิลต์ คอมเมนต์ว่าเก็บข้อมูลอะไร
- ตัวอย่างของข้อมูลที่เก็บลงฟิลต์ คอมเมนต์โดยระบุตัวอย่างภายในวงเล็บ
- ฟิลต์ที่อ้างอิง PK ของตารางอื่น คอมเมนต์เหมือนกันกับฟิลต์ที่อ้างอิง และระบุชื่อตารางและ/หรือฐานข้อมูลภายในวงเล็บต่อท้าย

5. การตั้งชื่อตัวแปรของ Config

- ตั้งชื่อด้วยตัวอักษรภาษาอังกฤษพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง (_) ได้
- ตั้งชื่อตามรูปแบบ `$config["ชื่อย่อของระบบ_ชื่อข้อมูล"]`

5. การตั้งชื่อตัวแปรของ Config (ต่อ)

- โฟลเดอร์ของระบบ ตั้งชื่อว่า folder
- ที่อยู่ไฟล์ที่อัปโหลดของระบบ ตั้งชื่อว่า upload_path
- ที่ตั้งไดเรกทอรีของระบบ ตั้งชื่อว่า root_path
- ชื่อฐานข้อมูลของระบบ ตั้งชื่อว่า db_name
- ที่อยู่รูปภาพต่างๆ ของระบบ ตั้งชื่อว่า image_ชื่อข้อมูล
- ที่อยู่ไอคอนต่างๆ ของระบบ ตั้งชื่อว่า icon_ชื่อข้อมูล
- กรณีตั้งชื่อคอนฟิกเกี่ยวกับระบบอื่น ตั้งชื่อขึ้นต้นด้วยชื่อระบบของตัวเองก่อน

```
$config['hr_folder'] = 'hr';  
$config['hr_upload_path'] = '';  
$config['hr_root_path'] = '';  
$config['hr_db_name'] = 'sme_hrdb';  
$config['hr_image_header'] = 'picture/header.png';  
$config['hr_icon_add'] = 'picture/add.png';  
$config['hr_icon_edit'] = 'picture/edit.png';  
$config['hr_ums_folder'] = '';  
  
$this->config->item('hr_folder');
```

6. การตั้งชื่อฟังก์ชันของ Helper

- ตั้งชื่อด้วยตัวอักษรภาษาอังกฤษพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง (_) ได้
- ตั้งชื่อฟังก์ชันตามรูปแบบ ชื่อย่อของระบบ_ชื่อการทำงาน
- ห้ามตั้งชื่อฟังก์ชันซ้ำกับ function_helper ที่มีอยู่แล้ว
- ห้ามเพิ่ม/ลบ/แก้ไขเกี่ยวกับฟังก์ชันในไฟล์ function_helper โดยพลการ
- กรณีคัดลอกฟังก์ชันมาจากระบบอื่น ตั้งชื่อขึ้นต้นด้วยชื่อระบบของตัวเองก่อน

```
function hr_convert_date() {  
    ...  
}  
function hr_convert_date2() {  
    ...  
}  
function hr_convert_date3() {  
    ...  
}  
function hr_ums_read_file() {  
    ...  
}  
$date = date('Y-m-d');  
$date = hr_convert_date($date);
```

7. การเขียน Comments

7.1 Comment คลาสของ Controller และ Model

7.2 Comment ฟังก์ชันใน Controller, Model และ Helper

7.3 Comment ส่วนของ View

7.4 Comment บรรทัดเดียว หรือตัวแปรต่างๆ

7.5 Comment ระบุขอบเขตส่วนการทำงานนั้นๆ

7.1 Comment คลาสของ Controller และ Model

```
/*  
* Position  
* Position Management  
* @author Patiya  
* @Create Date 2561-10-26  
*/
```

7.1 Comment คลาสของ Controller และ Model

```
/*  
* Position  
* Position Management  
* @author Patiya  
* @Update Date 2562-01-04  
*/
```

7.1 Comment คลาสของ Controller และ Model

```
/*  
* Position  
* Position จัดการข้อมูลตำแหน่ง  
* @author Patiya  
* @Create Date 2561-10-26  
*/
```

7.2 Comment ฟังก์ชันใน Controller, Model และ Helper

```
/*  
* Position_insert  
* Insert position into database  
* @input position_name_th, position_name_en  
* @output      last insert id (pos_id)  
* @author      Patiya  
* @Create Date 2561-10-26  
*/  
Function () {
```


7.2 Comment ฟังก์ชันใน Controller, Model และ Helper

```
/*  
* Position_insert  
* บันทึกข้อมูลตำแหน่ง ชื่อภาษาไทย และภาษาอังกฤษ ลงสู่ฐานข้อมูล  
* @input position_name_th, position_name_en  
* @output      last insert id (pos_id)  
* @author      Patiya  
* @Update Date 2561-10-26  
*/
```

7.3 Comment ส่วนของ View

```
/*  
* v_position_input  
* Display input form for add position  
* @input -  
* @output Input form for add position  
* @author Patiya  
* @Create Date 2561-10-26  
*/
```

7.3 Comment ส่วนของ View

```
/*  
* v_position_input  
* แสดงหน้าจอข้อมูลตำแหน่งที่ถูกบันทึกลงสู่ฐานข้อมูล  
  - บรรทัด 200  
* @input      -  
* @output     Input form for add position  
* @author     Patiya  
* @Create Date 2561-10-26  
*/
```

7.4 Comment บรรทัดเดียว หรือตัวแปรต่างๆ

```
// กำหนดค่าเริ่มต้นของ i
```

```
$i = 0;
```

7.5 Comment ระบุขอบเขตส่วนการทำงานนั้นๆ

```
/* Start Input Position Form */  
...  
...  
/* End Input Position Form */
```

มาตรฐานส่วนติดต่อผู้ใช้ (UI Standard)

1. การแสดงสีปุ่ม (Button Color)
2. การจัดวางตำแหน่งปุ่ม (Button Position)
3. การแสดงกล่องข้อความยืนยัน (Confirm Box)
4. การแสดงผลอื่นๆ
 - การแสดงข้อความแจ้งเตือน Form Validation
 - การแสดง Tooltip
 - การแสดง Placeholder
 - การแสดง Icon หรือรูปภาพแทนการดำเนินการ
 - การแสดง Datepicker และรูปแบบวันที่ (Date Format)

การแสดงสีปุ่ม (Button Color)

Operations

บันทึก

ยืนยัน แก้ไข

ลบ

Normal

Warning

Danger

พิมพ์

ส่งออกเอกสาร

เคลียร์

ยกเลิก

ค้นหา

เพิ่ม

คัดลอก

Exports

Clear, Cancel, Back, Close etc.

Others

การจัดวางตำแหน่งปุ่ม (Button Position)

เพิ่ม

พิมพ์ ส่งออกเอกสาร

DEFAULT TABLES								
#	First Name	Last Name	Username	ดำเนินการ				
1	Mark	Otto	@mdo		ดู	แก้ไข	ลบ	พิมพ์
2	Jacob	Thornton	@fat					
3	Larry	the Bird	@twitter					

การจัดวางตำแหน่งปุ่ม (Button Position)

PANEL

▼

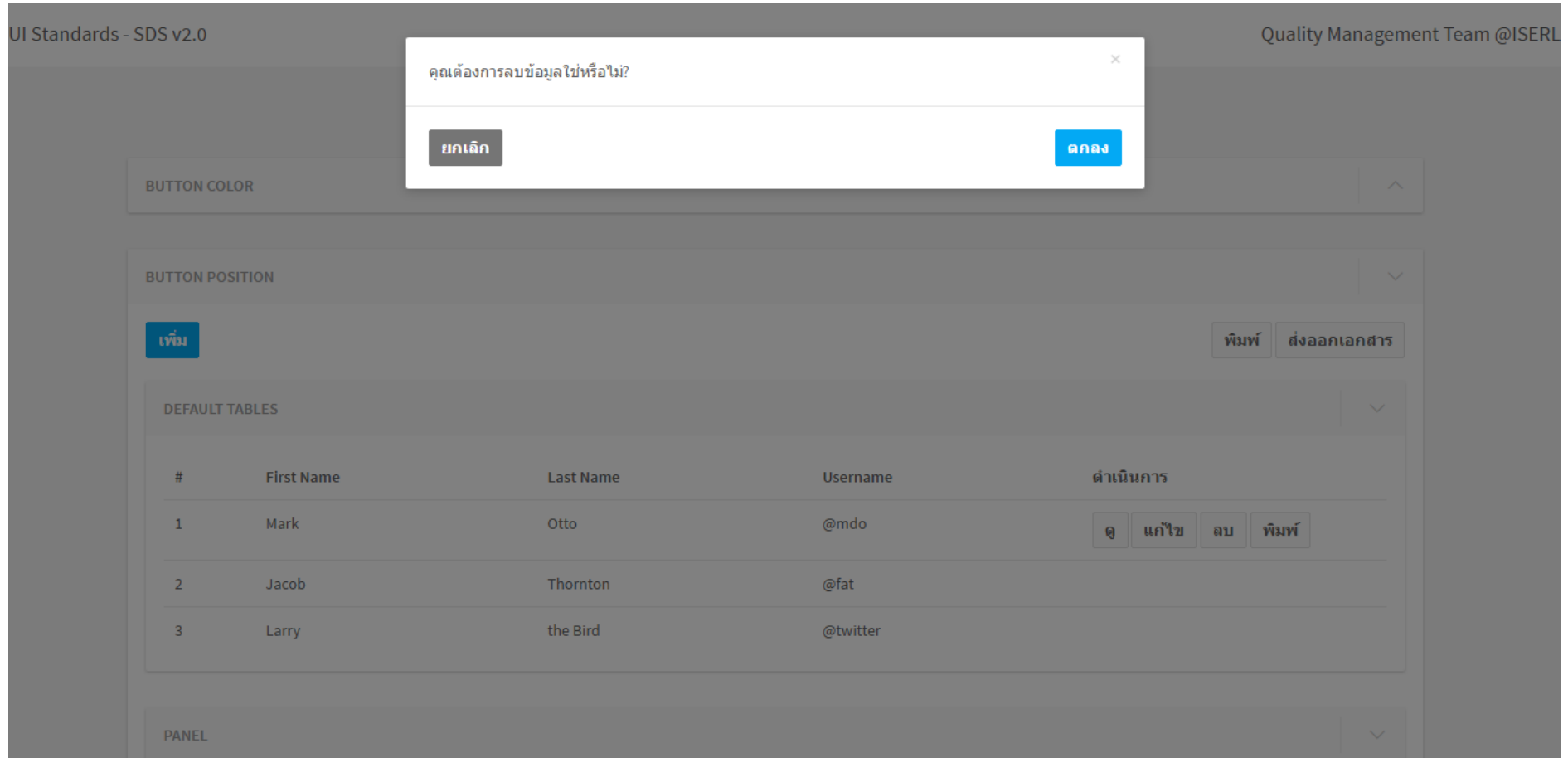
พิมพ์ส่งออกเอกสาร

Contents

เคลียร์ยกเลิก

ค้นหาคัดลอกบันทึกยืนยันแก้ไขลบ

การแสดงกล่องข้อความยืนยัน (Confirm Box)



การแสดง Tooltip

BUTTON POSITION

เพิ่ม

พิมพ์

ส่งออกเอกสาร

DEFAULT TABLES

#	First Name	Last Name	Username	
1	Mark	Otto	@mdo	<div><div>ดูรายละเอียด</div><div>ดูแก้ไขลบพิมพ์</div></div>
2	Jacob	Thornton	@fat	
3	Larry	the Bird	@twitter	