

บทที่ 7

JSON

ธีระยุทธ ทองเครือ

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยขอนแก่น



JSON

- ❖ JSON (JavaScript Object Notation) คือ รูปแบบข้อมูลที่ใช้ในการแลกเปลี่ยนระหว่างแอปพลิเคชัน ซึ่งมีความง่ายในการประมวลผลมากกว่ารูปแบบ XML
- ❖ JSON อยู่ในรูปแบบ Text ที่ไม่ขึ้นกับภาษาโปรแกรมใดๆ
- ❖ JSON มีโครงสร้างอยู่ 2 แบบ
 - แบบ Object คือ เก็บชุดของ key และ value ซึ่งอาจมองเป็น record ก็ได้
 - แบบ Array คือ รายการของ value หรือ object



รูปแบบ JSON Object



- ❖ เป็นรายการแบบไม่สนใจลำดับ
- ❖ แต่ละรายการประกอบด้วย key และ value
- ❖ key คือ คำอธิบายข้อมูลจะต้องขึ้นต้นและปิดท้ายด้วย "" เสมอ
- ❖ key และ value คั่นด้วย : และแต่ละคู่คั่นด้วย ,
- ❖ ส่วนของ Object ทั้งหมดจะครอบด้วย { และ }

```
{  
  key → "id": "01",  
  key → "book_name": "Java Programming",  
        "edition": "third",  
        "author": "Robert"  
}
```

value



ชนิดข้อมูลของ JSON



ชนิด	คำอธิบาย
Number	เป็นชนิด integer, double หรือ floating-point โดยไม่ต้องใส่ " " ครอบ
String	จะต้องขึ้นต้นและปิดท้ายด้วย " "
Boolean	ค่าเป็น true หรือ false
null	ค่าว่าง

```
{  
  "name": "Seksit",  
  "age": 35,  
  "distinction": true  
}
```



รูปแบบ JSON Array

- ❖ เป็นรายการแบบมีลำดับ
- ❖ มีการระบุชื่อ key ซึ่งเป็นชื่ออาร์เรย์ตามด้วยเครื่องหมาย :
- ❖ สมาชิกของ Array จะเก็บภายใต้ [และ]
- ❖ แต่ละสมาชิกจะคั่นด้วย ,

key
"destination": [
 "Udonthani",
 "Bangkok",
 "Khon Kaen"
] } สมาชิกอาร์เรย์



การใช้ JSON Array



manager และ staff มีหลายคนจึงกำหนดให้ value มีค่าเป็นอาร์เรย์

```
{  
  "President": "Alan Isaac",  
  "CEO": "David Richardson",  
  
  "manager": [  
    "Sachin Tendulkar",  
    "Virender Sehwag",  
    "Gautam Gambhir"  
  ],  
  
  "staff": [  
    "Alastair Cook",  
    "Jonathan Trott",  
    "Kevin Pietersen"  
  ]  
}
```



Array สามารถมีสมาชิกเป็น Object ได้



```
{
  "book": [
    {
      "id": "01",
      "language": "Java",
      "edition": "third",
      "author": "Herbert Schildt"
    },
    {
      "id": "07",
      "language": "C++",
      "edition": "second",
      "author": "E.Balagurusamy"
    }
  ]
}
```



กิจกรรม

❖ จากข้อมูลรายการสั่งซื้อสินค้าต่อไปนี้ จงออกแบบโครงสร้างข้อมูลของ JSON

Customer	Items order	Qty	Price
Mr. Lee	A4 paper	5	120
	Ruler	2	25
Mr. Yue	Flash Drive	2	130
	A4 paper	3	120

ใช้ JSON Editor จาก <http://jsoneditoronline.org>



ข้อแตกต่างระหว่าง JSON กับ XML



- ❖ XML ใช้แท็กในการระบุจุดเริ่มต้นและสิ้นสุด ขณะที่ JSON ใช้ { }
- ❖ JSON ใช้คำอธิบายน้อยกว่าทำให้ง่ายและรวดเร็วในการเขียน
- ❖ JSON มี Array ซึ่งสมาชิกแต่ละตัวของ Array จะไม่มีชื่อ
- ❖ XML สามารถใช้คำอะไรก็ได้ตั้งชื่อแท็ก ขณะที่ JSON จะต้องไม่เป็น reserve word ใน JavaScript



JSON และ XML



❖ JSON มีการอธิบายข้อมูลด้วยการระบุชื่อเช่นเดียวกับ XML

JSON

```
"company": "Volkswagen"
```

XML

```
<company>Volkswagen</company>
```



JSON และ XML



- ❖ JSON ไม่ต้องมี Root Element
- ❖ JSON ไม่ต้องมีแท็กเปิดและปิด

JSON

```
{  
  "company": "Volkswagen",  
  "name": "Vento",  
  "price": 800000  
}
```

XML

```
<car>  
  <company>Volkswagen</company>  
  <name>Vento</name>  
  <price>800000</price>  
</car>
```



JSON และ XML



❖ JSON มีลำดับชั้นเช่นเดียวกับ XML เพราะสามารถเขียน Object หรือ Array ซ้อนกัน
ได้

JSON

```
{  
  "title": "Kind of Blue",  
  "track": {  
    "length": 9.22,  
    "name": "So What"  
  }  
}
```

XML

```
<cd>  
  <title>Kind of Blue</title>  
  <track>  
    <length>9.22</length>  
    <name>So What</name>  
  </track>  
</cd>
```



กิจกรรม

❖ จากโครงสร้างของข้อมูลเอกสาร XML ต่อไปนี้จึงเขียนให้อยู่ในรูปแบบ JSON

```
<student id="566003204-1">  
  <stdName>Robert</stdName>  
  <gpa>3.06</gpa>  
  <register>  
    <subject sid="322433" grade="A" />  
    <subject sid="322236" grade="C" />  
  </register>  
</student>
```

ใช้ JSON Editor จาก <http://jsoneditoronline.org>



การดึงข้อมูลจาก JSON



- ❖ **JSON Path** คือ ประโยคในการระบุเส้นทางในการเข้าถึงข้อมูลที่อยู่ในรูปแบบ JSON เหมาะกับการใช้ดึงข้อมูลเพียงบางส่วนจาก JSON
- ❖ **JSON Data Binding** คือ การแปลงข้อมูลใน JSON ให้อยู่ในรูปแบบ object ของ Java (POJO) เหมาะกับผู้ที่ต้องการใช้ข้อมูลหลายๆส่วนจาก JSON ในที่นี้จะใช้ Jackson Library



เว็บทดสอบ JsonPath



❖ เว็บไซต์สำหรับทดสอบ JsonPath คือ <http://jsonpath.herokuapp.com/>

The screenshot shows the Jayway JsonPath Evaluator interface. The browser address bar displays `jsonpath.herokuapp.com`. The page title is "Jayway JsonPath Evaluator" with a version of 2.2.0 and a timestamp of 2016-03-01 09:22:35. A dropdown menu shows "Goessner example". The main JSON input area contains a sample JSON object with a "store" property containing an array of "book" objects. Below the JSON, the "JsonPath" input field contains the query `$.store.book[*].author`, and a "Go!" button is visible. On the right, the "Jayway" tab is selected, showing a result of "0 millis" and a link to "About implementation...". The result area displays an array of author names: ["Nigel Rees", "Evelyn Waugh", "Herman Melville", "J. R. R. Tolkien"].

ข้อมูลในรูปแบบ JSON ที่จะใช้ทดสอบ

ระบุคำสั่ง JsonPath

ผลลัพธ์จาก JsonPath



JSONPath Operator




Operator	คำอธิบาย
\$	แทน root ของ JSON ใช้หน้า JSONPath ทุกครั้ง
@	ใช้กำหนดเงื่อนไข หรือกรองข้อมูล
*	ใช้แทนตัวเลข หรือตัวอักษรใดๆ
..	ละชื่อ key ของ JSONPath ทั้งหมดไปถึงชื่อ key ที่ระบุ
.<name>	ใช้ระบุชื่อ key ย่อยที่ต้องการ
['<name>' (, '<name>')]	ใช้ระบุชื่อ key ย่อยที่ต้องการแบบหลายค่า
[<number> (, <number>)]	ใช้ระบุสมาชิกของอาร์เรย์ที่ต้องการ (อาร์เรย์จะเริ่มต้นที่ 0)
[start:end]	ใช้กำหนดจุดเริ่มต้นและสิ้นสุด (ไม่รวมจุดสิ้นสุด) ของอาร์เรย์ที่ต้องการ
[?(<expression>)]	ใช้ระบุเงื่อนไข



Filter Operators

Operator	คำอธิบาย
==	การเท่ากันทั้งซ้ายและขวา (ข้อสังเกต 1 จะไม่เท่ากับ '1')
!=	ไม่เท่ากับ
<	น้อยกว่า
<=	น้อยกว่าหรือเท่ากับ
>	มากกว่า
>=	มากกว่าหรือเท่ากับ
=~	เปรียบเทียบตามรูปแบบ regular expression [?(@.name =~ /foo.*?/i)]
in	มีค่าตามที่กำหนด [?(@.size in ['S', 'M'])]
nin	ไม่มีค่าตามที่กำหนด
size	ขนาดของอาร์เรย์หรือ String
empty	อาร์เรย์หรือ String เป็นค่าว่าง



❖ ขอชื่อผู้เขียนทั้งหมด

Path แบบเต็ม: \$.store.book[*].author


Path แบบย่อ: \$..author

ผลลัพธ์:

```
[  
  "Nigel Rees",  
  "Evelyn Waugh",  
  "Herman Melville",  
  "J. R. R. Tolkien"  
]
```

โหลดจาก <http://dev.cs.kku.ac.th/jw/slide/book.json>

```
{  
  "store": {  
    "book": [  
      {  
        "category": "reference",  
        "author": "Nigel Rees",  
        "price": 8.95  
      },  
      {  
        "category": "fiction",  
        "author": "Evelyn Waugh",  
        "price": 12.99  
      },  
      {  
        "category": "fiction",  
        "author": "Herman Melville",  
        "isbn": "0-553-21311-3",  
        "price": 8.99  
      },  
      {  
        "category": "fiction",  
        "author": "J. R. R. Tolkien",  
        "isbn": "0-395-19395-8",  
        "price": 22.99  
      }  
    ],  
    "bicycle": {  
      "color": "red",  
      "price": 19.95  
    }  
  },  
  "expensive": 10  
}
```




❖ ขอชื่อผู้เขียน และราคาทั้งหมด

Path: `$..['author','price']`

ผลลัพธ์:

```
[
  {
    "author" : "Nigel Rees",
    "price" : 8.95
  },
  {
    "author" : "Evelyn Waugh",
    "price" : 12.99
  },
  {
    "author" : "Herman Melville",
    "price" : 8.99
  },
  {
    "author" : "J. R. R. Tolkien",
    "price" : 22.99
  }
]
```

```
{
  "store": {
    "book": [
      {
        "category": "reference",
        "author": "Nigel Rees",
        "price": 8.95
      },
      {
        "category": "fiction",
        "author": "Evelyn Waugh",
        "price": 12.99
      },
      {
        "category": "fiction",
        "author": "Herman Melville",
        "isbn": "0-553-21311-3",
        "price": 8.99
      },
      {
        "category": "fiction",
        "author": "J. R. R. Tolkien",
        "isbn": "0-395-19395-8",
        "price": 22.99
      }
    ],
    "bicycle": {
      "color": "red",
      "price": 19.95
    }
  },
  "expensive": 10
}
```



❖ ต้องการชื่อผู้เขียนของหนังสือเล่มที่ 2

Path: \$.store.book[1].author

ผลลัพธ์:

"Evelyn Waugh"

❖ ต้องการราคาสินค้าทั้งหมด

Path: \$..price

ผลลัพธ์: [

8.95,
12.99,
8.99,
22.99,
19.95

]

```
{
  "store": {
    "book": [
      {
        "category": "reference",
        "author": "Nigel Rees",
        "price": 8.95
      },
      {
        "category": "fiction",
        "author": "Evelyn Waugh",
        "price": 12.99
      },
      {
        "category": "fiction",
        "author": "Herman Melville",
        "isbn": "0-553-21311-3",
        "price": 8.99
      },
      {
        "category": "fiction",
        "author": "J. R. R. Tolkien",
        "isbn": "0-395-19395-8",
        "price": 22.99
      }
    ],
    "bicycle": {
      "color": "red",
      "price": 19.95
    }
  },
  "expensive": 10
}
```




❖ ต้องการข้อมูลหนังสือเล่มที่ 3

Path: `$.book[2]`

ผลลัพธ์:

```
[
  {
    "category" : "fiction",
    "author" : "Herman Melville",
    "isbn" : "0-553-21311-3",
    "price" : 8.99
  }
]
```

```
{
  "store": {
    "book": [
      {
        "category": "reference",
        "author": "Nigel Rees",
        "price": 8.95
      },
      {
        "category": "fiction",
        "author": "Evelyn Waugh",
        "price": 12.99
      },
      {
        "category": "fiction",
        "author": "Herman Melville",
        "isbn": "0-553-21311-3",
        "price": 8.99
      },
      {
        "category": "fiction",
        "author": "J. R. R. Tolkien",
        "isbn": "0-395-19395-8",
        "price": 22.99
      }
    ],
    "bicycle": {
      "color": "red",
      "price": 19.95
    }
  },
  "expensive": 10
}
```




❖ ต้องการราคาหนังสือเล่มสุดท้าย

Path: `$.book[3].price`

ผลลัพธ์:

```
[  
  22.99  
]
```

```
{  
  "store": {  
    "book": [  
      {  
        "category": "reference",  
        "author": "Nigel Rees",  
        "price": 8.95  
      },  
      {  
        "category": "fiction",  
        "author": "Evelyn Waugh",  
        "price": 12.99  
      },  
      {  
        "category": "fiction",  
        "author": "Herman Melville",  
        "isbn": "0-553-21311-3",  
        "price": 8.99  
      },  
      {  
        "category": "fiction",  
        "author": "J. R. R. Tolkien",  
        "isbn": "0-395-19395-8",  
        "price": 22.99  
      }  
    ],  
    "bicycle": {  
      "color": "red",  
      "price": 19.95  
    }  
  },  
  "expensive": 10  
}
```



❖ ต้องการข้อมูลหนังสือ เล่มที่ 1, 2 และ 4

Path: `$.book[0,1,3]`

ผลลัพธ์:

```
[
  {
    "category": "reference",
    "author": "Nigel Rees",
    "title": "Sayings of the Century",
    "price": 8.95
  },
  {
    "category": "fiction",
    "author": "Evelyn Waugh",
    "title": "Sword of Honour",
    "price": 12.99
  },
  {
    "category": "fiction",
    "author": "J. R. R. Tolkien",
    "title": "The Lord of the Rings",
    "isbn": "0-395-19395-8",
    "price": 22.99
  }
]
```

```
{
  "store": {
    "book": [
      {
        "category": "reference",
        "author": "Nigel Rees",
        "price": 8.95
      },
      {
        "category": "fiction",
        "author": "Evelyn Waugh",
        "price": 12.99
      },
      {
        "category": "fiction",
        "author": "Herman Melville",
        "isbn": "0-553-21311-3",
        "price": 8.99
      },
      {
        "category": "fiction",
        "author": "J. R. R. Tolkien",
        "isbn": "0-395-19395-8",
        "price": 22.99
      }
    ],
    "bicycle": {
      "color": "red",
      "price": 19.95
    }
  },
  "expensive": 10
}
```



❖ ต้องการข้อมูลหนังสือที่มี index ของ
อาร์เรย์ตั้งแต่ 0 ถึง 1


Path: `$.book[0:2]`

2 คือ ไม่รวม 2
(exclusive)

ผลลัพธ์:

```
[
  {
    "category": "reference",
    "author": "Nigel Rees",
    "price": 8.95
  },
  {
    "category": "fiction",
    "author": "Evelyn Waugh",
    "price": 12.99
  }
]
```

```
{
  "store": {
    "book": [
      {
        "category": "reference",
        "author": "Nigel Rees",
        "price": 8.95
      },
      {
        "category": "fiction",
        "author": "Evelyn Waugh",
        "price": 12.99
      },
      {
        "category": "fiction",
        "author": "Herman Melville",
        "isbn": "0-553-21311-3",
        "price": 8.99
      },
      {
        "category": "fiction",
        "author": "J. R. R. Tolkien",
        "isbn": "0-395-19395-8",
        "price": 22.99
      }
    ],
    "bicycle": {
      "color": "red",
      "price": 19.95
    }
  },
  "expensive": 10
}
```



❖ ต้องการหนังสือที่มี isbn

Path: `$..book[?(@.isbn)]`

ผลลัพธ์:

```
[
  {
    "category" : "fiction",
    "author" : "Herman Melville",
    "isbn" : "0-553-21311-3",
    "price" : 8.99
  },
  {
    "category" : "fiction",
    "author" : "J. R. R. Tolkien",
    "isbn" : "0-395-19395-8",
    "price" : 22.99
  }
]
```

```
{
  "store": {
    "book": [
      {
        "category": "reference",
        "author": "Nigel Rees",
        "price": 8.95
      },
      {
        "category": "fiction",
        "author": "Evelyn Waugh",
        "price": 12.99
      },
      {
        "category": "fiction",
        "author": "Herman Melville",
        "isbn": "0-553-21311-3",
        "price": 8.99
      },
      {
        "category": "fiction",
        "author": "J. R. R. Tolkien",
        "isbn": "0-395-19395-8",
        "price": 22.99
      }
    ],
    "bicycle": {
      "color": "red",
      "price": 19.95
    }
  },
  "expensive": 10
}
```



❖ ต้องการชื่อผู้เขียนที่อยู่ใน category

ชื่อ "fiction"


Path:

`$.book[?(@.category=='fiction')].author`

ผลลัพธ์:

```
[  
  "Evelyn Waugh",  
  "Herman Melville",  
  "J. R. R. Tolkien"  
]
```

```
{  
  "store": {  
    "book": [  
      {  
        "category": "reference",  
        "author": "Nigel Rees",  
        "price": 8.95  
      },  
      {  
        "category": "fiction",  
        "author": "Evelyn Waugh",  
        "price": 12.99  
      },  
      {  
        "category": "fiction",  
        "author": "Herman Melville",  
        "isbn": "0-553-21311-3",  
        "price": 8.99  
      },  
      {  
        "category": "fiction",  
        "author": "J. R. R. Tolkien",  
        "isbn": "0-395-19395-8",  
        "price": 22.99  
      }  
    ],  
    "bicycle": {  
      "color": "red",  
      "price": 19.95  
    }  
  },  
  "expensive": 10  
}
```


❖ ต้องการหนังสือที่มีราคาน้อยกว่า 10

Path: \$.store.book[?(@.price < 10)]

ผลลัพธ์:

```
[
  {
    "category" : "reference",
    "author" : "Nigel Rees",
    "price" : 8.95
  },
  {
    "category" : "fiction",
    "author" : "Herman Melville",
    "isbn" : "0-553-21311-3",
    "price" : 8.99
  }
]
```

```
{
  "store": {
    "book": [
      {
        "category": "reference",
        "author": "Nigel Rees",
        "price": 8.95
      },
      {
        "category": "fiction",
        "author": "Evelyn Waugh",
        "price": 12.99
      },
      {
        "category": "fiction",
        "author": "Herman Melville",
        "isbn": "0-553-21311-3",
        "price": 8.99
      },
      {
        "category": "fiction",
        "author": "J. R. R. Tolkien",
        "isbn": "0-395-19395-8",
        "price": 22.99
      }
    ],
    "bicycle": {
      "color": "red",
      "price": 19.95
    }
  },
  "expensive": 10
}
```



❖ ต้องการหนังสือที่มีราคาน้อยกว่าค่าใน

key expensive

Path: `$.book[?(@.price <= $('expensive'))]`

ผลลัพธ์:

```
[
  {
    "category": "reference",
    "author": "Nigel Rees",
    "price": 8.95
  },
  {
    "category": "fiction",
    "author": "Herman Melville",
    "isbn": "0-553-21311-3",
    "price": 8.99
  }
]
```

```
{
  "store": {
    "book": [
      {
        "category": "reference",
        "author": "Nigel Rees",
        "price": 8.95
      },
      {
        "category": "fiction",
        "author": "Evelyn Waugh",
        "price": 12.99
      },
      {
        "category": "fiction",
        "author": "Herman Melville",
        "isbn": "0-553-21311-3",
        "price": 8.99
      },
      {
        "category": "fiction",
        "author": "J. R. R. Tolkien",
        "isbn": "0-395-19395-8",
        "price": 22.99
      }
    ],
    "bicycle": {
      "color": "red",
      "price": 19.95
    }
  },
  "expensive": 10
}
```



กิจกรรม

❖ จงเขียน JSON Path สำหรับดึงข้อมูลจากเอกสาร XML ซึ่งมีโครงสร้างและข้อมูลดังภาพ

เงื่อนไข	JSON Path
แสดงชื่อลูกค้าทั้งหมด	
แสดงชื่อลูกค้าคนสุดท้าย	
แสดงชื่อสินค้าที่มีราคา มากกว่า 100 บาท	
แสดงรายการสินค้าของ ลูกค้าที่ชื่อ "Mr. Yue"	

```
{
  "customers": [
    {
      "name": "Mr. Lee",
      "item": [
        {
          "item_name": "A4 Paper",
          "price": 120
        }
      ]
    },
    {
      "name": "Mr. Yue",
      "item": [
        {
          "item_name": "Flash Drive",
          "price": 130
        },
        {
          "item_name": "book",
          "price": 20
        }
      ]
    },
    {
      "name": "Mr. Kim",
      "item": [
        {
          "item_name": "Ruler",
          "price": 25
        }
      ]
    }
  ]
}
```



JsonPath Library



❖ สร้าง Maven Project โดยเลือกต้นแบบเป็น webapp

❖ เพิ่ม Library (หรือ Dependency) จาก <https://github.com/jayway/JsonPath> ให้ไว้

ภายใต้แท็ก <dependencies>

ใน pom.xml

- ▼ Libraries
 - > Apache Tomcat v9.0 [Apache Tomcat]
 - > JRE System Library [jre1.8.0_112]
- ▼ Maven Dependencies
 - > junit-3.8.1.jar - C:\Users\NG-Dor
 - > mysql-connector-java-6.0.5.jar - C:\Users\NG-Dor
 - > json-path-2.2.0.jar - C:\Users\NG-Dor
 - > json-smart-2.2.1.jar - C:\Users\NG-Dor
 - > accessors-smart-1.1.jar - C:\Users\NG-Dor
 - > asm-5.0.3.jar - C:\Users\NG-Dor
 - > slf4j-api-1.7.16.jar - C:\Users\NG-Dor

GitHub - jayway/JsonPath

GitHub, Inc. [US] | <https://github.com/jayway/JsonPath>

Jayway JsonPath

A Java DSL for reading JSON documents.

build passing maven central 2.2.0 javadoc 2.2.0

Jayway JsonPath is a Java port of Stefan Goessner JsonPath implementation.

News

- 29 Feb 2016 - Released JsonPath 2.2.0
- 22 Nov 2015 - Released JsonPath 2.1.0
- 19 Mar 2015 - Released JsonPath 2.0.0
- 11 Nov 2014 - Released JsonPath 1.2.0
- 01 Oct 2014 - Released JsonPath 1.1.0
- 26 Sep 2014 - Released JsonPath 1.0.0

For details see [change log](#).

Getting Started

JsonPath is available at the Central Maven Repository. Maven users add this to your POM.

```
<dependency>  
  <groupId>com.jayway.jsonpath</groupId>  
  <artifactId>json-path</artifactId>  
  <version>2.2.0</version>  
</dependency>
```

If you need help ask questions at [Stack Overflow](#). Tag the question 'jsonpath' and 'java'.



การใช้ JSONPath ดึงข้อมูลจากไฟล์ JSON



// 1. อ่านไฟล์ JSON เก็บลง String

```
String json = new String(Files.readAllBytes(Paths.get("D:/customer.json")), "UTF8");
```

// 2. แปลงข้อมูลจาก String ให้อยู่ในรูปแบบ Object ของ JSON

```
Object document = Configuration.defaultConfiguration().jsonProvider().parse(json);
```

// 3. ดึงข้อมูลด้วย JSON Path

// 3.1 กรณีที่ผลลัพธ์ไม่เป็น Array

```
String cusName = JsonPath.read(document, "$.customers[2].name");  
System.out.println("ชื่อลูกค้าคนสุดท้าย คือ " + cusName);
```

// 3.2 กรณีที่ผลลัพธ์เป็น Array ต้องระบุ type ใน Array ให้ถูกด้วย ว่าเป็น String หรือ Integer

```
List<String> name = JsonPath.read(document, "$..item[?(@.price>100)].item_name");  
System.out.println("สินค้าที่มีราคามากกว่า 100 บาท ได้แก่");  
for(int i=0; i<name.size(); i++) {  
    System.out.println(name.get(i));  
}
```

ชื่อลูกค้าคนสุดท้าย คือ Mr. Kim
สินค้าที่มีราคามากกว่า 100 บาท ได้แก่
A4 Paper
Flash Drive

ผลลัพธ์



การใช้ JSONPath ดึงข้อมูลจาก Web API



❖ การติดต่อกับ Web API จำเป็นต้อง
ใช้ไลบรารีช่วยในการส่งและรับ
ข้อมูล ในที่นี้เลือกใช้ OkHttp

❖ เพิ่ม OkHttp Library โดยใช้ข้อมูล
จาก <http://square.github.io/okhttp/>
ใส่ไว้ภายใต้แท็ก <dependencies>
ใน pom.xml

The screenshot shows the OkHttp website with the following content:

- Header: OkHttp, Download v3.6.0
- Section: Download
- Button: ↓ v3.6.0 JAR
- Text: You'll also need **Okio**, which OkHttp uses for fast I/O and resizable buffers. Download the **latest JAR**.
- Text: The source code to OkHttp, its samples, and this website is [available on GitHub](#).
- Section: MAVEN
- Code Block:

```
<dependency>
  <groupId>com.squareup.okhttp3</groupId>
  <artifactId>okhttp</artifactId>
  <version>3.6.0</version>
</dependency>
```




การใช้ JSONPath ดึงข้อมูลจาก Web API



```
String placeName = "ขอนแก่น";
```

```
OkHttpClient client = new OkHttpClient();
```

```
// 1. กำหนด URL ของเว็บเซอร์วิส Google
```

```
String googleUrl = "https://maps.googleapis.com/maps/api/geocode/json?address=" + placeName;
```

```
// 2. เรียกใช้ Web API
```

```
Request request = new Request.Builder().url(googleUrl).build();
```

```
Response response = client.newCall(request).execute();
```

```
// 3. แปลงข้อมูลจาก String ให้อยู่ในรูปแบบ Object ของ JSON
```

```
Object document = Configuration.defaultConfiguration().jsonProvider().parse(response.body().string());
```

```
// 4. ดึงข้อมูลด้วย JSONPath
```

```
Double lat = JsonPath.read(document, "$.results[0].geometry.location.lat");
```

```
Double lng = JsonPath.read(document, "$.results[0].geometry.location.lng");
```

```
System.out.println(lat + ", " + lng);
```



JSON Data Binding



- ❖ JSON Data Binding คือ การแปลงข้อมูลใน JSON ให้อยู่ในรูปแบบ object ของ Java เหมาะกับแอปพลิเคชันที่ต้องการใช้ข้อมูลหลายๆส่วนจาก JSON
- ❖ การแปลงข้อมูลใน JSON ให้อยู่ในรูปแบบ object ของ Java (POJO) มีขั้นตอนดังนี้
 - สร้างคลาส JavaBean เก็บข้อมูลจากเว็บเซอร์วิส
 - เขียนชุดคำสั่งในการเรียกใช้ และดึงข้อมูลที่ได้ไปแสดงผล
- ❖ ในที่นี่จะใช้ Jackson Library
 - <https://github.com/FasterXML/jackson-databind>

```
<dependency>  
  <groupId>com.fasterxml.jackson.core</groupId>  
  <artifactId>jackson-databind</artifactId>  
  <version>2.7.0</version>  
</dependency>
```



Google Geocoding API

<https://developers.google.com/maps/documentation/geocoding/>



Geocoding API Request Format

A Geocoding API request must be of the following form:

```
https://maps.googleapis.com/maps/api/geocode/output?parameters
```

where **output** may be either of the following values:

- **json** (recommended) indicates output in JavaScript Object Notation (JSON)
- **xml** indicates output as XML

To access the Geocoding API over HTTP, use:

```
http://maps.googleapis.com/maps/api/geocode/output?parameters
```

HTTP is not recommended for applications that include sensitive user data, such as a user's location, in requests.

Some parameters are required while some are optional. As is standard in URLs, parameters are separated using the ampersand (&) character.

Google Maps API for Work users must include valid **client** and **signature** parameters with their Geocoding requests. Please refer to [Google Maps API for Work Web Services](#) for more information.

The rest of this page describes [geocoding](#) and [reverse geocoding](#) separately, because different parameters are available for each type of request.

Geocoding (Latitude/Longitude Lookup)

Required parameters in a geocoding request:

- **address** — The street address that you want to geocode, in the format used by the national postal service of the country concerned. Additional address elements such as business names and unit, suite or floor numbers should be avoided. Please refer to [the FAQ](#) for additional guidance.
or
- **components** — A component filter for which you wish to obtain a geocode. See [Component Filtering](#) for more information. The components filter will also be accepted as an optional parameter if an **address** is provided.



Google Geocoding API

Server: <https://maps.google.com>

Resource:

/maps

/api

/geocode

/json

?address=[ชื่อสถานที่]



<https://maps.googleapis.com/maps/api/geocode/json?address=ขอนแก่น>

The screenshot shows a web browser window with the URL <https://maps.googleapis.com/maps/api/geocode/json?address=ขอนแก่น>. The browser displays the JSON response from the Google Geocoding API. The response is a JSON object with a 'results' array containing one result for 'Khon Kaen, Thailand'. The result includes address components, types, formatted address, geometry (bounds and location), and location details. A green arrow points to the 'location' object in the JSON response, which contains the latitude and longitude coordinates for Khon Kaen.

```
{
  "results": [
    {
      "address_components": [
        {
          "long_name": "Khon Kaen",
          "short_name": "จ.ขอนแก่น",
          "types": [
            "administrative_area_level_1",
            "political"
          ]
        },
        {
          "long_name": "Thailand",
          "short_name": "TH",
          "types": [
            "country",
            "political"
          ]
        }
      ],
      "formatted_address": "Khon Kaen, Thailand",
      "geometry": {
        "bounds": {
          "northeast": {
            "lat": 17.094254,
            "lng": 103.1846
          },
          "southwest": {
            "lat": 15.6330732,
            "lng": 101.7500362
          }
        },
        "location": {
          "lat": 16.4419355,
          "lng": 102.8359921
        }
      }
    }
  ]
}
```

สร้างคลาสเพื่อ Binding ข้อมูลจาก Web API

```
{
  - results: [
    - {
      - address_components: [
        - {
          long_name: "Khon Kaen",
          short_name: "จ.ขอนแก่น",
          - types: [
            "administrative_area_level_1",
            "political"
          ]
        },
      ],
      - {
        long_name: "Thailand",
        short_name: "TH",
        - types: [
          "country",
          "political"
        ]
      },
    ],
    formatted_address: "Khon Kaen, Thailand",
    - geometry: {
      - bounds: {
        - northeast: {
          lat: 17.094254,
          lng: 103.1846
        },
        - southwest: {
          lat: 15.6330732,
          lng: 101.7500362
        }
      },
      - location: {
        lat: 16.4419355,
        lng: 102.8359921
      }
    }
  ]
}
```

```
class GoogleResult {
    public List<PlaceDetail> results;
}
```

หากต้องการตั้งชื่อ attribute ที่มี
ความแตกต่างจาก key ใน JSON
สามารถใช้ annotation กำกับได้

```
class PlaceDetail {
    @JsonProperty("formatted_address")
    public String address;

    public Geometry geometry;
}
```

```
class Geometry {
    public Location location;
}
```

```
class Location {
    public String lat;
    public String lng;
}
```



การเรียก Web API และใช้คลาส Binding



// 1. กำหนด URL ของเว็บเซอร์วิส

```
String locationName = URLEncoder.encode("ขอนแก่น", StandardCharsets.UTF_8.toString());  
String url = "https://maps.googleapis.com/maps/api/geocode/json?address=" +  
locationName;
```

// 2. Config ค่าเกี่ยวกับการแปลง JSON เป็น Object

```
ObjectMapper mapper = new ObjectMapper();  
mapper.disable(DeserializationFeature.FAIL_ON_UNKNOWN_PROPERTIES);
```

// 3. เรียกใช้ Web API และแปลงเป็น POJO

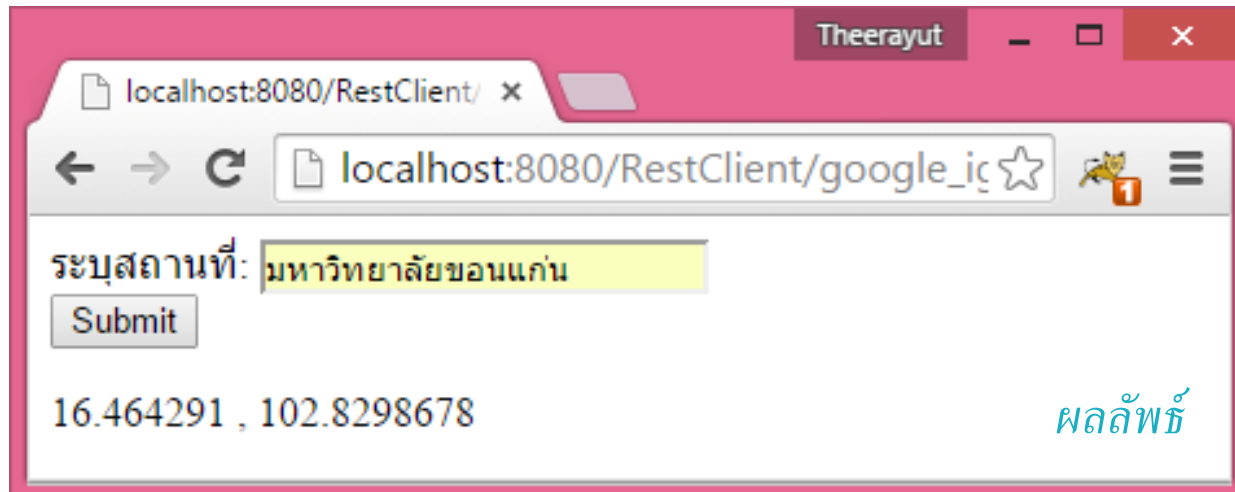
```
GoogleResult googleResult = mapper.readValue(new URL(url), GoogleResult.class);
```

// 4. ดึงค่าจาก POJO

```
System.out.println(googleResult.results.get(0).geometry.location.lat);  
System.out.println(googleResult.results.get(0).geometry.location.lng);
```


กิจกรรม

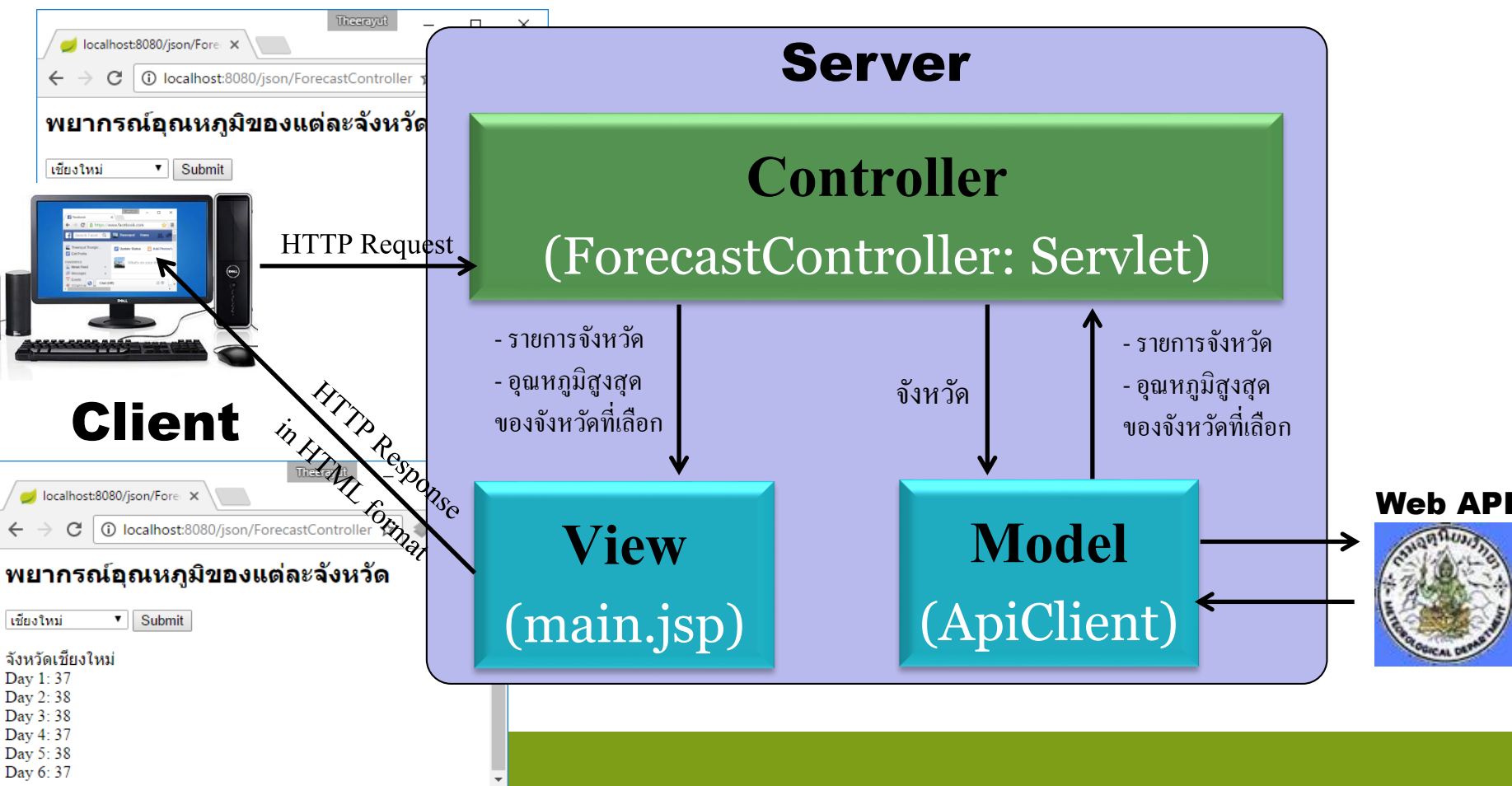
- ❖ จงสร้างเว็บแอปพลิเคชันแบบ MVC โดยมีแบบฟอร์มสำหรับรับชื่อสถานที่ใดๆ และส่งไปค้นหา Latitude และ Longitude ที่ Web API ของ Google Map โดยขอข้อมูลในรูปแบบ JSON หลังจากนั้นดึงข้อมูลจาก JSON มาแสดงผลให้ผู้ใช้งาน โดยใช้วิธี Data Binding





Assignment#8

- ❖ สร้างเว็บไซต์แสดงการพยากรณ์อุณหภูมิ โดยเขียนตามสถาปัตยกรรม MVC ในส่วน Model ให้เรียกใช้ Web API ของกรมอุตุนิยมวิทยา (<http://data.tmd.go.th/api/index1.php>) ชื่อ WeatherForecast7Days (request ข้อมูลแบบ JSON) ส่วนหน้าเว็บผู้ใช้สามารถเลือกจังหวัดที่ต้องการทราบอุณหภูมิที่กรมฯพยากรณ์ไว้ได้





JSON View บน Chrome

<https://github.com/tulios/json-viewer>



The screenshot shows the Chrome Web Store page for the 'JSONView' extension. The page is titled 'JSONView from gildas' and shows a 4.5-star rating from 1,347 reviews, with 810,915 users. The extension is categorized under 'Developer Tools'. A green arrow points to the '+ ADD TO CHROME' button. The page also features a 'Style editor' section with CSS and JSON code, a 'Validate and view JSON documents' section, and a 'Notes' section. The bottom of the page shows a 'Fotor Banner Maker' advertisement and a 'busuu.com' logo.

JSONView - Chrome Web x

chrome web store

theera yut ku.ac.th

JSONView
from gildas

★★★★★ (1347) Developer Tools 810,915 users

OVERVIEW REVIEWS SUPPORT RELATED

Style editor

CSS

```
body {
  white-space: pre;
  font-family: monospace;
}
.property {
  font-weight: bold;
}
.type-null {
  color: gray;
}
.type-boolean {
  color: firebrick;
}
.type-number {
  color: blue;
}
.type-string {
  color: green;
}
.callback-function {
  color: gray;
}
.collapse:after {
  content: "...";
}
.collapsed > .collapse:after {
  content: "...";
}
.ellipsis:after {
  content: "...";
}
.collapsible {
  margin-left: 2em;
}
```

Preview

```
{
  boy: "guy",
  number: 249,
  anobject: {
    whoa: "nuts",
    + anarray: [ ... ],
    more: "stuff"
  },
  awesome: true,
  bogus: false,
  meaning: null,
  japanese: "えいごはいい、",
  link: "http://jsonview.com",
  notLink: "http://jsonview.com is great"
}
```

Validate and view JSON documents

JSONView port for Chrome.

Original firefox extension is here:
<http://benhollis.net/software/jsonview/>

Notes:

- JSON is validated using a client-side javascript implementation of JSONLint (<http://github.com/zaach/jsonlint>)
- this extension displays JSON text compliant with rfc 4627 (<http://www.ietf.org/rfc/rfc4627.txt>).
- JSONP (<http://en.wikipedia.org/wiki/JSON#JSONP>) is supported

Website

Report Abuse

Version: 0.0.32.2

Updated: October 31, 2013

Size: 94.03KB

Language: English

Fotor Banner Maker
No photoshop, easy to make & beautiful

busuu.com
the language learning community

The Jackal of Javascript



การสร้าง Web API ที่ให้ผลลัพธ์เป็น JSON



- ❖ Web API หรือ Web Services Provider นอกจากจะเปิดข้อมูลให้กับระบบอื่นที่ต้องการใช้ข้อมูลร่วมกันในรูปแบบ XML แล้ว ยังสามารถให้บริการในรูปแบบของ JSON ได้ด้วย
- ❖ การสร้าง Web API ในรูปแบบ JSON สามารถทำได้ 2 วิธี
 - **แบบที่ 1 ต่อ String:** นำผลลัพธ์ที่ได้จากการประมวลผลจัดให้อยู่ในรูปแบบ String ของ JSON
 - **แบบที่ 2 POJO -> JSON:** นำผลลัพธ์ที่ได้จากการประมวลผลเก็บลง Object ของ JavaBean แล้วใช้ Library Jackson แปลงเป็นรูปแบบ JSON

แบบที่ 1: ต่อ String

`response.setContentType("application/json; charset=utf-8");` // ให้ส่ง response กลับเป็น JSON

`PrintWriter out = response.getWriter();`

```
try {
    Class.forName("com.mysql.jdbc.Driver");
    String dbURL = "jdbc:mysql://localhost/blueshop?characterEncoding=utf-8";
    Connection con = DriverManager.getConnection(dbURL, "root", "");
    Statement statement = con.createStatement();
    ResultSet resultSet = statement.executeQuery("select * from product");
```

```
String json = "{\n\"results\":[";
while (resultSet.next()) {
    json += "{\n\"id\": " + resultSet.getInt("pid") + ",";
    json += "\"name\": \"" + resultSet.getString("pname") + "\",";
    json += "\"pdetail\": \"" + resultSet.getString("pdetail") + "\",";
    json += "\"price\": " + resultSet.getInt("price") + "}";
    if (!resultSet.isLast()) json += ",";
}
json += "]}";
out.println(json);
} catch (ClassNotFoundException e) {
    System.err.println("Error loading driver: " + e);
} catch (SQLException e) {
    System.err.println("Error database connection: " + e);
}
```

ผลลัพธ์

```
{
  "results": [
    {
      "id": 1,
      "name": "Centrum",
      "detail": "วิตามินรวมจาก A ถึง Zinc",
      "price": 350
    },
    {
      "id": 2,
      "name": "Caltrate",
      "detail": "บำรุงกระดูก เสริมวิตามินดี",
      "price": 760
    }
  ]
}
```



แบบที่ 2: POJO -> JSON



- ❖ สร้างคลาส JavaBean ที่เป็นไปตามโครงสร้างของ JSON ที่ต้องการ
- ❖ สร้าง Object JavaBean และเก็บข้อมูลลงไปใน Object
- ❖ ใช้เมธอดของ ObjectMapper แปลง Object เป็น String ของ JSON



คลาส JavaBeans



```
class ProductList {  
    private List<Product> results = new ArrayList<Product>();  
    public List<Product> getResults() { return results; }  
    public void setResults(List<Product> results) { this.results = results; }  
}
```

```
class Product {  
    private int id;  
    private String name;  
    private String detail;  
    private int price;  
    public int getId() {return id;}  
    public void setId(int id) {this.id = id;}  
    public String getName() {return name;}  
    public void setName(String name) {this.name = name;}  
    public String getDetail() {return detail;}  
    public void setDetail(String detail) {this.detail = detail;}  
    public int getPrice() {return price;}  
    public void setPrice(int price) {this.price = price;}  
}
```




ใช้ Library แปลง Object เป็น JSON



```
ObjectMapper mapper = new ObjectMapper(); // ประกาศ Object ที่ใช้สำหรับแปลง POJO เป็น JSON
ProductList productList = new ProductList(); // ประกาศ Object เก็บรายการจากฐานข้อมูล
response.setContentType("application/json; charset=utf-8"); // ให้ส่ง response กลับเป็น JSON
PrintWriter out = response.getWriter();
try {
    Class.forName("com.mysql.jdbc.Driver");
    String dbURL = "jdbc:mysql://localhost/blueshop?characterEncoding=utf-8";
    Connection con = DriverManager.getConnection(dbURL, "root", "");
    Statement statement = con.createStatement();
    ResultSet resultSet = statement.executeQuery("select * from product");
    while (resultSet.next()) {
        Product product = new Product();
        product.setId(resultSet.getInt("pid"));
        product.setName(resultSet.getString("pname"));
        product.setDetail(resultSet.getString("pdetail"));
        product.setPrice(resultSet.getInt("price"));
        productList.getResults().add(product);
    }
} catch (ClassNotFoundException e) {
    System.err.println("Error loading driver: " + e);
} catch (SQLException e) {
    System.err.println("Error database connection: " + e);
}
mapper.configure(SerializationFeature.INDENT_OUTPUT, true); // กำหนดให้ผลลัพธ์มีการเยื้อง
mapper.setSerializationInclusion(Include.NON_EMPTY); // หากมีค่าว่างใน attribute ใดใน object ให้ตัด key นั้นออก
mapper.writeValue(out, productList); // เริ่มแปลงเป็น JSON ที่นี่
```




Assignment#9



- ❖ สร้าง Web API ที่ให้เฉพาะข้อมูลบางส่วนจากฐานข้อมูล ได้แก่ firstname, lastname, dept_name
- ❖ ให้บริการข้อมูลในรูปแบบ JSON
- ❖ ใช้ไฟล์ xcompany.sql ในการสร้างตารางและเพิ่มข้อมูล
- ❖ ตาราง department และ employee มีโครงสร้างและข้อมูลดังนี้

ตาราง employee

emp_id	firstname	lastname	position	dept_id
1	บุญมี	มากเหลือ	โปรแกรมเมอร์	1
2	สมัย	สมภาร	นักบัญชี	3
3	บุญช่วย	กาโว	ผู้จัดการ	2
4	สมใจ	บุญหลาย	Tester	1
5	ยี่หวา	เร้าร้อน	ธุรการ	2

ตาราง department

dept_id	dept_name
1	ฝ่ายสารสนเทศ
2	ฝ่ายทรัพยากรบุคคล
3	ฝ่ายบัญชี