

The background features a grid of light gray squares. From the left edge, several wide, overlapping rays of color (orange, red, green, blue, and gray) extend towards the center. On the right side, there are faint, stylized line graphs in blue and orange, and scattered binary code (0s and 1s) in a light gray font.

Chapter 5

JSON

JavaScript Object Notation

Theerayut Thongkrau

➡ การเปิดเผยข้อมูลด้วย Web API

- Web API (Application Programming Interface) คือ การเผยแพร่ข้อมูลที่อยู่ในรูปแบบมาตรฐาน ให้บุคคลภายนอกสามารถเข้าถึงได้
- Web Site มนุษย์เป็นผู้อ่าน แต่ Web API แอปพลิเคชันเป็นผู้อ่าน
- การเผยแพร่ข้อมูลในแบบ Web API เป็นวิธีที่ช่วยให้นักพัฒนาโปรแกรมสามารถนำข้อมูลไปใช้ประมวลผล หรือแสดงผล ในวิธีการของตนเองได้
- รูปแบบมาตรฐานในปัจจุบัน ได้แก่ XML และ JSON

➡ ผู้ให้บริการ และ ผู้ใช้บริการ

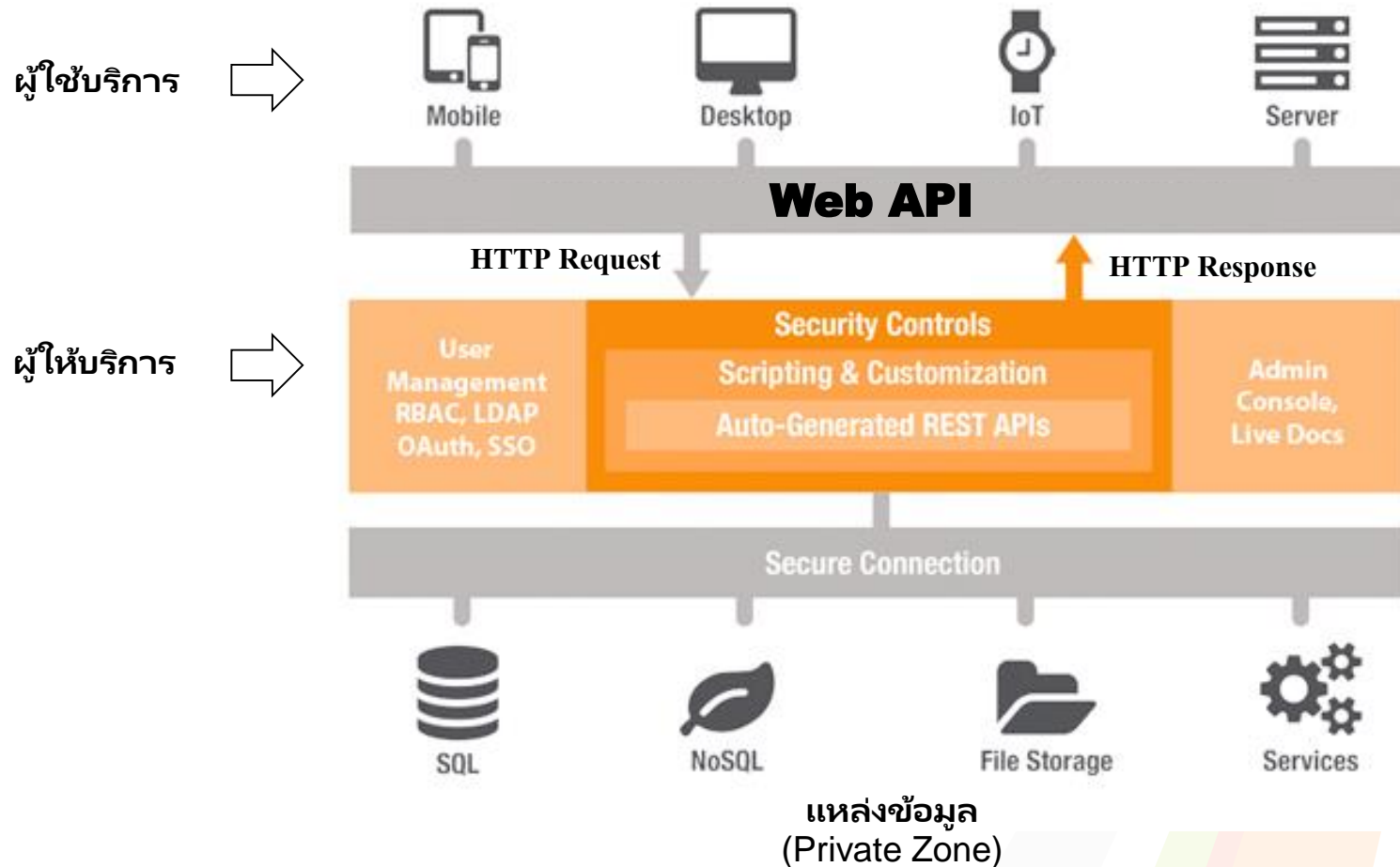
- ผู้ให้บริการ API

- Twitter API
- Facebook API ให้บริการข้อมูลเพื่อให้แอปพลิเคชันอื่นๆนำไปวิเคราะห์ เช่น การตลาด กระแสความนิยม
- Google Maps ให้บริการข้อมูลเกี่ยวกับแผนที่
- รัฐบาลไทย ให้ข้อมูลเกี่ยวกับประเทศไทย

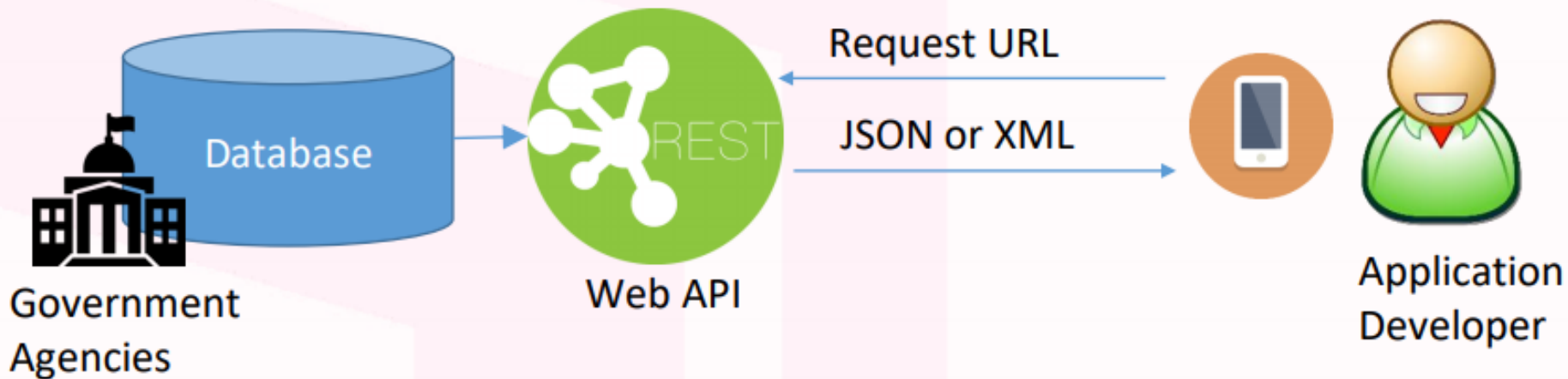
- ผู้ใช้ API

- Mobile Application
- Web Application
- IoT Application

➡ สถาปัตยกรรมระบบที่มี Web API



➡ Web API ชุดข้อมูลเปิดภาครัฐ



JSON

- รูปแบบข้อความมาตรฐานสำหรับการแทนโครงสร้างข้อมูล
- ใช้ในการส่งข้อมูลระหว่างไคลเอนต์และเซิร์ฟเวอร์
 - Web Browser <-> Server
 - Mobile Application <-> Server
 - IoT Application <-> Server

➡ JSON กับภาษา JavaScript

- JSON มีรูปแบบไวยากรณ์คล้ายคลึงกับ object และ array ของ JavaScript

```
<html>
<body>
<script>
  let person = {
    id: 69,
    fullname: 'John Smith',
    weight: 72.5,
    option: ['move', 'stop', 'slow']
  }
  console.log(person.weight * 0.5)
</script>
</body>
</html>
```

Object ของ
JavaScript

➡ JSON Syntax

- ข้อมูลที่อยู่ในรูปแบบ JSON อยู่ในรูปแบบลำดับของคู่ key และ value

"key1": "value1", "key2": "value2", ...

- key คือ คำอธิบายข้อมูลจะต้องขึ้นต้นและปิดท้ายด้วยเครื่องหมาย " เสมอ
- หากมีหลายคู่ key และ value จะคั่นด้วยเครื่องหมาย comma
- รายการ key และ value เป็นรายการแบบไม่สนใจลำดับ

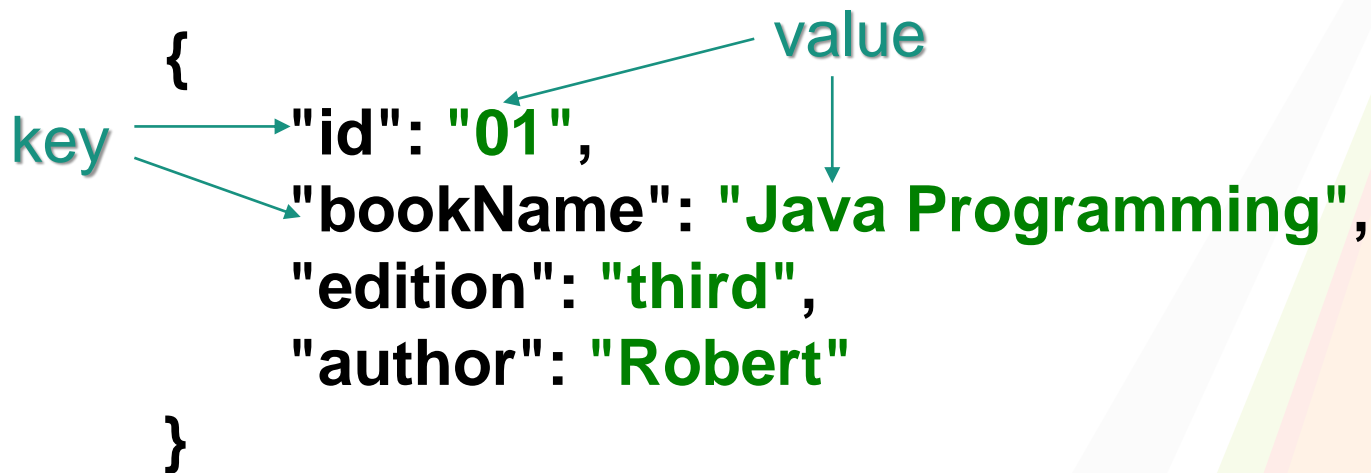
➡ ชนิดข้อมูลของ JSON

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

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

➡ JSON Object

- JSON Object คือ โครงสร้างข้อมูลประกอบด้วยรายการของคู่ key และ value ในเครื่องหมาย { และ }



```
{  
  "id": "01",  
  "bookName": "Java Programming",  
  "edition": "third",  
  "author": "Robert"  
}
```

The diagram illustrates the structure of a JSON object. A teal arrow labeled "key" points to the opening curly brace "{". Another teal arrow labeled "value" points to the string "01" in the "id" pair. A third teal arrow points from the "value" label to the string "Java Programming" in the "bookName" pair. The JSON object is enclosed in curly braces, and the keys and values are separated by colons and commas.

➡ JSON Array

- เป็นรายการแบบมีลำดับ ภายใต้เครื่องหมาย [....]
- สิ่งที่แตกต่างกันจาก JSON Object คือ สมาชิกของอาร์เรย์จะไม่มีชื่อ key ขณะที่ JSON Object สมาชิกจะมีชื่อ key เสมอ
- แต่ละสมาชิกจะคั่นด้วย ,

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

➡ Object ที่มี Array

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

manager และ staff มีหลายคนจึง
กำหนดให้ value มีค่าเป็นอาร์เรย์

➡ Array ဖွဲ့ Object

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

➡ Object ซ่อน Object

object ของ address ที่
ซ่อนอยู่ และภายใน
object ยังมี object
geolocation อีกด้วย

```
{  
  "fullName": "John",  
  "lastName": "Smith",  
  "address" : {  
    "zipCode": "40000",  
    "country": "Thailand",  
    "geolocation" : {  
      "lat": 16.4331702,  
      "lng": 102.8407473  
    }  
  }  
}
```

➡ การแปลง JavaScript Object เป็น String JSON

- Object ในภาษา JavaScript สามารถแปลงเป็น String ได้ด้วยคำสั่ง

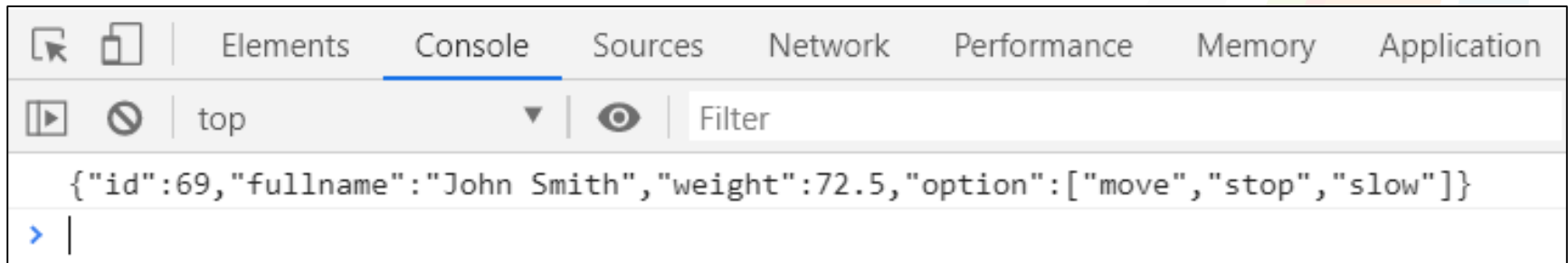
`JSON.stringify ([ตัวแปรที่เป็นobject หรือ array])`

- จะใช้เมื่อต้องการส่งข้อมูลไปยัง server

➡ ตัวอย่าง

```
<html><body>
<script>
  let person = {
    id: 69,
    fullname: 'John Smith',
    weight: 72.5,
    option: ['move', 'stop', 'slow']
  }

  // แปลงจาก object เป็น string รูปแบบ JSON
  let jsonString = JSON.stringify(person)
  console.log(jsonString)
</script>
</body></html>
```



➡ การแปลง String เป็น Object ของ JavaScript

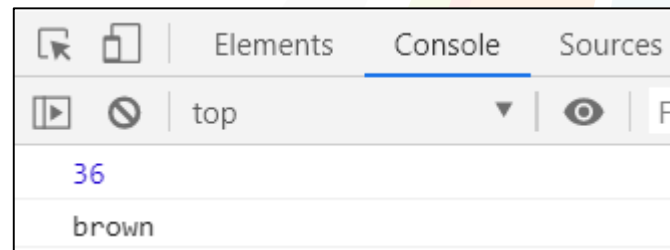
- String ที่อยู่ในรูปแบบ JSON สามารถแปลงเป็น object เพื่อให้นำไปใช้งานต่อด้วยคำสั่ง `JSON.parse()`

```
<html>
<body>
<script>
  // string ในรูปแบบ JSON ก่อนถูกแปลง
  let jsonString = '{"height":1.9, "age":36, "eyeColor":"brown"}'

  // ใช้ JSON.parse() เพื่อแปลงเป็น Object
  let person = JSON.parse(jsonString)

  // นำ object ไปใช้งาน
  console.log(person.age)
  console.log(person.eyeColor)

</script>
</body></html>
```

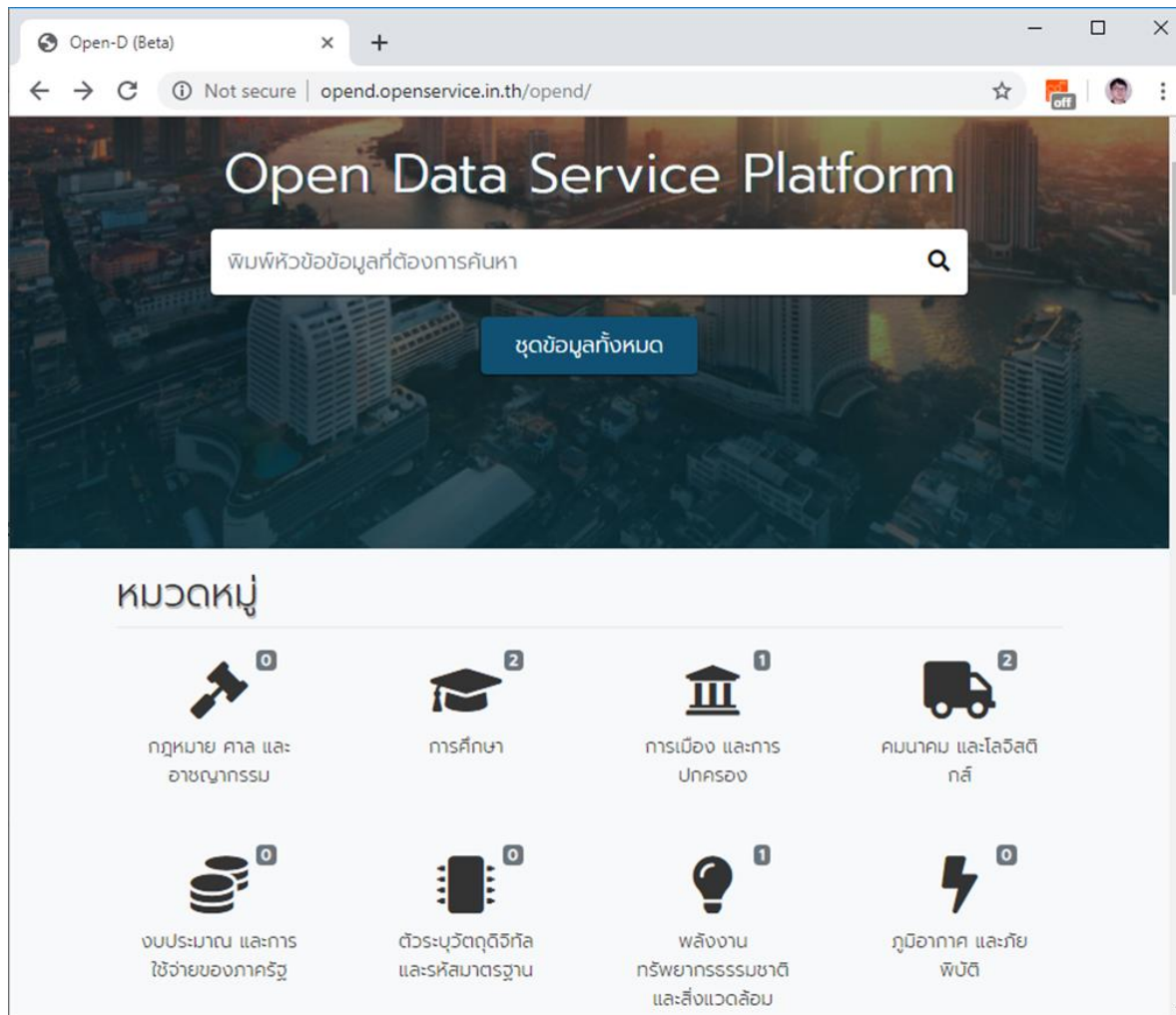


➡ การดึงข้อมูล JSON จาก Web API

- ศึกษารูปแบบของ URL สำหรับขอข้อมูล JSON จากคู่มือของ Web API
 - บาง Web API จะต้องสมัครสมาชิก เพื่อนำ token แนบไปกับ URL
- ใช้โปรแกรมทดสอบดึงข้อมูล เช่น Postman
- เขียนคำสั่งดึงข้อมูลผ่าน HTTP โดยภาษาโปรแกรมใดๆ เช่น ภาษา JavaScript ใช้ Fetch API หรือ XMLHttpRequest
- ใช้ Library หรือคำสั่งช่วยแปลง JSON ให้เป็น object แล้วนำไปประมวลผล หรือแสดงผล

➡ Web API ข้อมูลรัฐบาล

<http://opend.openservice.in.th/opend/>



เลือกข้อมูลที่ต้องการ

Open-D (Beta)

Not secure | opend.openservice.in.th/opend/search_ds/dataset/9

หมวดหมู่

- กฎหมาย ศาล และ
อาชญากรรม 0
- การศึกษา 2
- การเมือง และการปกครอง 1
- คมนาคม และโลจิสติกส์ 2
- งบประมาณ และการใช้จ่าย
ของภาครัฐ 0
- ตัวระบุวัตถุดิจิทัล และรหัส
มาตรฐาน 0
- พลังงาน
ทรัพยากรธรรมชาติ และสิ่ง
แวดล้อม 1
- ภูมิอากาศ และภัยพิบัติ 0
- วิทยาศาสตร์เทคโนโลยี
และนวัตกรรม (วทน.) 0
- ศาสนา ศิลปะ และ
วัฒนธรรม 17

รายการชุดข้อมูล

จำนวนหนี้สินเฉลี่ยต่อครัวเรือน จำแนกตามภาคและจังหวัด พ.ศ. 2539 - 2556

Download [CSV](#) [RDF](#)

เกี่ยวกับข้อมูลชุดนี้

จำนวนหนี้สินเฉลี่ยต่อครัวเรือน จำแนกตามภาคและจังหวัด พ.ศ. 2539 - 2556 (ที่มา: การสำรวจภาวะเศรษฐกิจและสังคมของครัวเรือน สำนักงานสถิติแห่งชาติ กระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร รวบรวมโดย: สำนักสถิติพยากรณ์ สำนักงานสถิติแห่งชาติ) ชุดข้อมูล download จาก Data.go.th เพื่อวัตถุประสงค์ในการสาธิตระบบ Open-D

ผู้เผยแพร่ชุดข้อมูล: [โครงการ Open-D](#)

ปรับปรุงข้อมูลล่าสุด: วันที่ 16/09/2018

สร้างวันที่: วันที่ 01/03/2018

ชื่อผู้ติดต่อ: โครงการ Open-D

อีเมลผู้ติดต่อ: open-d@nectec.or.th

ความถี่ในการปรับปรุงข้อมูล:

หมวดหมู่: เศรษฐกิจ การเงิน และอุตสาหกรรม

ป้ายกำกับ: หนี้สิน สถิติ

ลิขสิทธิ์: Open Government License - Thailand

[API](#)

เลือก API

➡ ระบุเงื่อนไขข้อมูลที่ต้องการ

Open-D (Beta) x Open-D (Beta) x +

Not secure | opend.openservice.in.th/search_virtuoso/after/index.jsp?language=th&dsname=vir_...

Open-D (Beta) ขุดข้อมูล

จำนวนหนี้สินเฉลี่ยต่อครัวเรือน จำแนกตามภาคและจังหวัด พ.ศ. 2539 - 2556

ค้นหาขั้นสูง

ค้นหาขั้นสูง :

1. กำหนดเงื่อนไข และทดสอบดึงข้อมูล

jeonhai province Contains ขอนแก่น

jeonhai

รายงาน Reset

ค้นหา

2. คลิกเพื่อดู URL สำหรับดึงข้อมูล JSON

Share | Get API

Label Property

records (10)

ลำดับ	province	year	debt_value
54	ขอนแก่น	2539	52988
131	ขอนแก่น	2541	111650
208	ขอนแก่น	2543	64250
285	ขอนแก่น	2545	96173

ขอ URL สำหรับดึงข้อมูล

คัดลอก URL และ
ทดสอบเปิดบน browser

Open-D (Beta)

Not secure | opend.openservice.in.th/search_virtuoso/after/index.jsp?language=th&dsname=vir_...

Open-D (Beta) ชัดข้อมูล

Get API : copy this url

Copy URL

`http://opend.openservice.in.th/search_virtuoso/api/dataset/query?dsname=vir_9_1519461425&path=vir_9_1519461425&property=col_1&operator=CONTAINS&valueLiteral=%E0%B8%82%E0%B8%AD%E0%B8%99%E0%B9%81%E0%B8%81%E0%B9%88%E0%B8%99&loadAll=1&type=json&limit=100&offset=0`

JSON RDF Add Sparql Close

ภาษา: TH

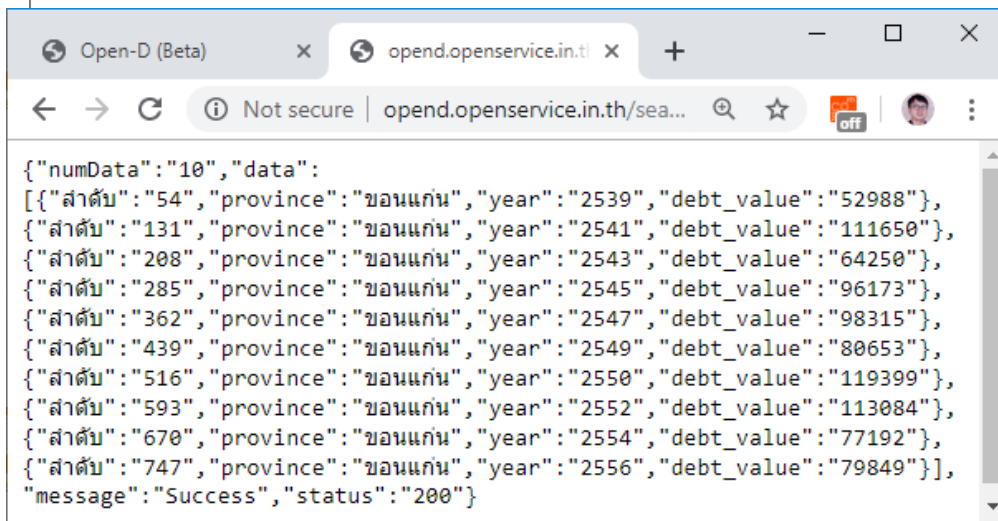
Share | Get API

Label Property

records (10)

ลำดับ	province	year	debt_value
54	ขอนแก่น	2539	52988
131	ขอนแก่น	2541	111650
208	ขอนแก่น	2543	64250
285	ขอนแก่น	2545	96173
362	ขอนแก่น	2547	98315

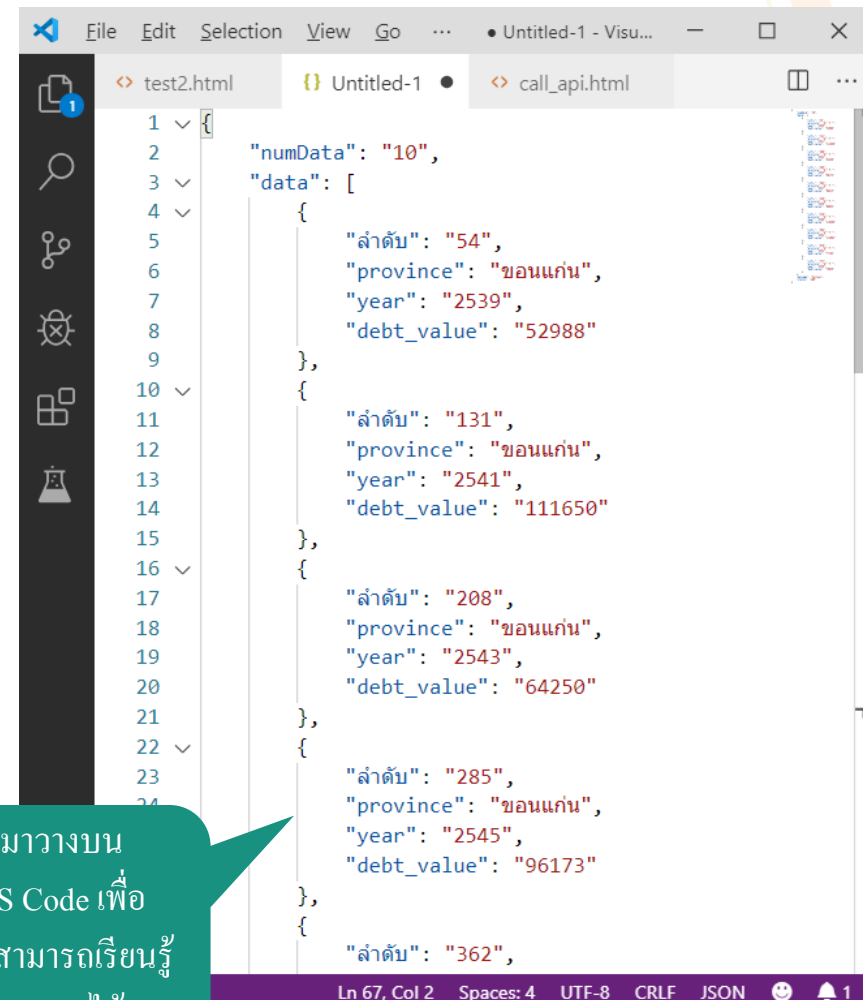
➡ ทดสอบเปิดดูข้อมูล



Open-D (Beta) | opend.openservice.in.th

```
{ "numData": "10", "data": [
  { "ลำดับ": "54", "province": "ขอนแก่น", "year": "2539", "debt_value": "52988" },
  { "ลำดับ": "131", "province": "ขอนแก่น", "year": "2541", "debt_value": "111650" },
  { "ลำดับ": "208", "province": "ขอนแก่น", "year": "2543", "debt_value": "64250" },
  { "ลำดับ": "285", "province": "ขอนแก่น", "year": "2545", "debt_value": "96173" },
  { "ลำดับ": "362", "province": "ขอนแก่น", "year": "2547", "debt_value": "98315" },
  { "ลำดับ": "439", "province": "ขอนแก่น", "year": "2549", "debt_value": "80653" },
  { "ลำดับ": "516", "province": "ขอนแก่น", "year": "2550", "debt_value": "119399" },
  { "ลำดับ": "593", "province": "ขอนแก่น", "year": "2552", "debt_value": "113084" },
  { "ลำดับ": "670", "province": "ขอนแก่น", "year": "2554", "debt_value": "77192" },
  { "ลำดับ": "747", "province": "ขอนแก่น", "year": "2556", "debt_value": "79849" } ],
  "message": "Success", "status": "200" }
```

ข้อมูลที่ได้ในรูปแบบ JSON



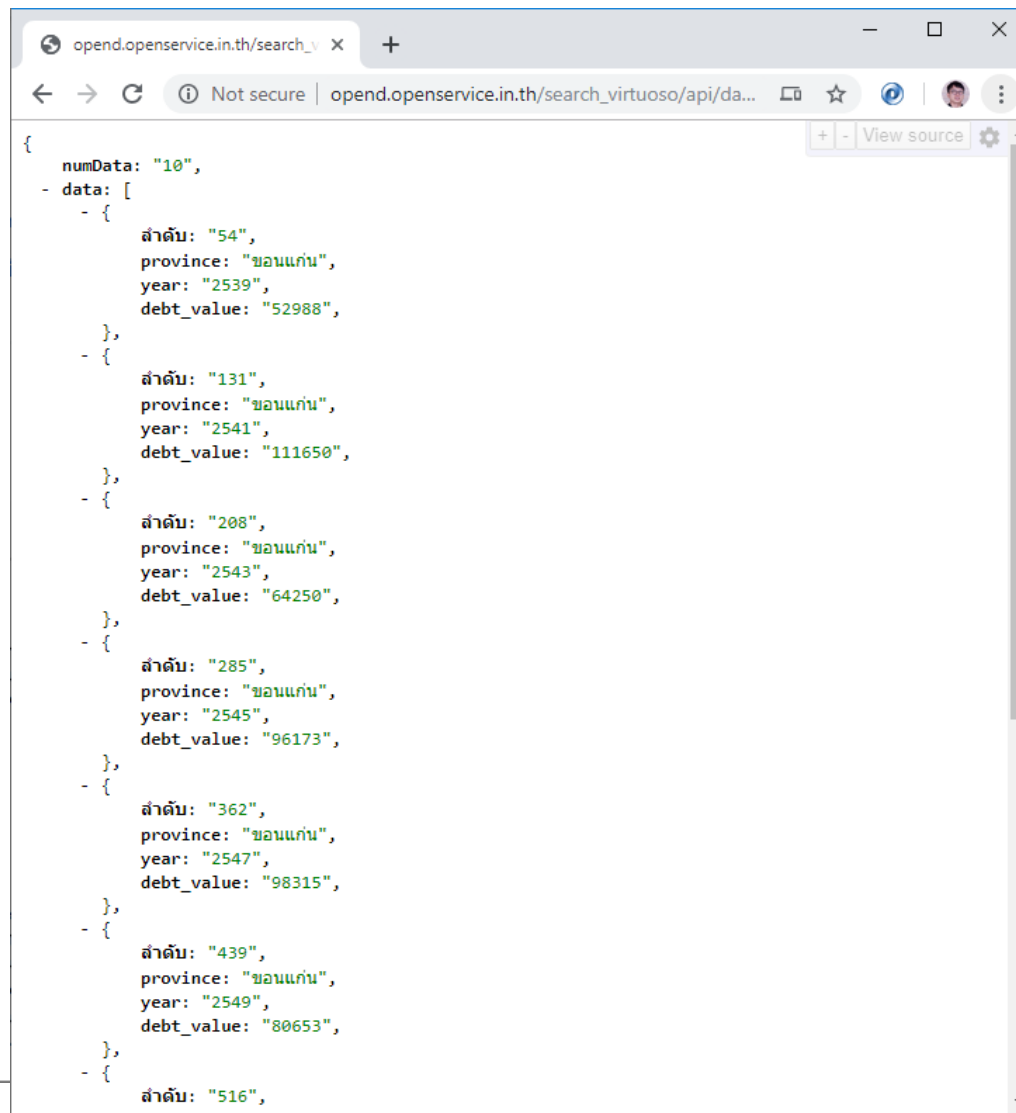
test2.html | Untitled-1 | call_api.html

```
1 {
2   "numData": "10",
3   "data": [
4     {
5       "ลำดับ": "54",
6       "province": "ขอนแก่น",
7       "year": "2539",
8       "debt_value": "52988"
9     },
10    {
11      "ลำดับ": "131",
12      "province": "ขอนแก่น",
13      "year": "2541",
14      "debt_value": "111650"
15    },
16    {
17      "ลำดับ": "208",
18      "province": "ขอนแก่น",
19      "year": "2543",
20      "debt_value": "64250"
21    },
22    {
23      "ลำดับ": "285",
24      "province": "ขอนแก่น",
25      "year": "2545",
26      "debt_value": "96173"
27    },
28    {
29      "ลำดับ": "362",
30      "province": "ขอนแก่น",
31      "year": "2547",
32      "debt_value": "98315"
33    },
34    {
35      "ลำดับ": "439",
36      "province": "ขอนแก่น",
37      "year": "2549",
38      "debt_value": "80653"
39    },
40    {
41      "ลำดับ": "516",
42      "province": "ขอนแก่น",
43      "year": "2550",
44      "debt_value": "119399"
45    },
46    {
47      "ลำดับ": "593",
48      "province": "ขอนแก่น",
49      "year": "2552",
50      "debt_value": "113084"
51    },
52    {
53      "ลำดับ": "670",
54      "province": "ขอนแก่น",
55      "year": "2554",
56      "debt_value": "77192"
57    },
58    {
59      "ลำดับ": "747",
60      "province": "ขอนแก่น",
61      "year": "2556",
62      "debt_value": "79849"
63    }
64  ]
65 }
```

สามารถนำมาวางบน
โปรแกรม VS Code เพื่อ
จัดรูปแบบ ให้สามารถเรียนรู้
โครงสร้าง JSON ได้

➡ Extension ช่วยจัดมุมมอง JSON

- JSONView: <https://jsonview.com/>



The screenshot shows a web browser window with the address bar displaying "opend.openservice.in.th/search_v...". The page content shows a JSON object with the following structure:

```
{
  numData: "10",
  - data: [
    - {
      ลำดับ: "54",
      province: "ขอนแก่น",
      year: "2539",
      debt_value: "52988",
    },
    - {
      ลำดับ: "131",
      province: "ขอนแก่น",
      year: "2541",
      debt_value: "111650",
    },
    - {
      ลำดับ: "208",
      province: "ขอนแก่น",
      year: "2543",
      debt_value: "64250",
    },
    - {
      ลำดับ: "285",
      province: "ขอนแก่น",
      year: "2545",
      debt_value: "96173",
    },
    - {
      ลำดับ: "362",
      province: "ขอนแก่น",
      year: "2547",
      debt_value: "98315",
    },
    - {
      ลำดับ: "439",
      province: "ขอนแก่น",
      year: "2549",
      debt_value: "80653",
    },
    - {
      ลำดับ: "516",

```




การเขียนคำสั่งอ่านข้อมูล JSON

<html>

<head>

<script>

```
async function getDataFromAPI() {  
  let response = await fetch('http://opend.openservice.in.th/sea..')  
  let rawData = await response.text() // อ่านผลลัพธ์  
  let objectData = JSON.parse(rawData) // แปลผลลัพธ์เป็น object  
  let result = document.getElementById('result') // ดึง<ul> เพื่อใช้ในการเพิ่มแท็ก<li>
```

รอให้การดึงข้อมูลจาก
Web API เสร็จก่อน

```
  for (let i = 0; i < objectData.data.length; i++) {  
    let content = 'ปี'+ objectData.data[i].year + ': ' // ดึงข้อมูลจาก object มาจัดรูปแบบ  
    content += objectData.data[i].debt_value + ' บาท'
```

```
    let li = document.createElement('li') // สร้างแท็ก<li>  
    li.innerHTML = content // นำข้อมูลที่จัดแล้วมาไว้ในแท็ก<li>  
    result.appendChild(li) // เพิ่มแท็ก<li> ใหม่
```

```
  }  
}
```

```
getDataFromAPI() // เรียกฟังก์ชัน
```

</script>

</head>

<body>

<h1>จำนวนหนี้สินเฉลี่ยต่อครัวเรือน จ.ขอนแก่น พ.ศ. 2539 - 2556</h1>

<ul id="result">

</body></html>

async ใช้คู่กับ await เพื่อบอกว่าเป็นการทำงานแบบ asynchronous

await คือ การรอให้ประมวลผลเสร็จก่อน

test2.html

File | C:/Users/MyPC/Desktop/t...

จำนวนหนี้สินเฉลี่ยต่อครัวเรือน จ.ขอนแก่น พ.ศ. 2539 - 2556

- ปี 2539: 52988 บาท
- ปี 2541: 111650 บาท
- ปี 2543: 64250 บาท
- ปี 2545: 96173 บาท
- ปี 2547: 98315 บาท
- ปี 2549: 80653 บาท
- ปี 2550: 119399 บาท
- ปี 2552: 113084 บาท
- ปี 2554: 77192 บาท
- ปี 2556: 79849 บาท

➡ Web API สำหรับทดสอบดึงข้อมูล

- โปรแกรมที่แบ่งส่วน front-end และ back-end ออกจากกัน
ฝั่ง front-end จำเป็นต้องออกแบบหน้าจอแสดงผล
- หาก back-end ยังไม่สามารถเตรียม Web API ให้ได้ทันที
front-end อาจต้องจำลอง Web API ขึ้นมาก่อน
- front-end อาจใช้ Fake Online Web API ทดสอบดึงข้อมูลได้

<https://jsonplaceholder.typicode.com/>