

{ REST }

ดร.ชัยวัฒน์ ศิระวัฒนานนท์
01204271 วิศวกรรมคอมพิวเตอร์เบื้องต้น
มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตเฉลิมพระเกียรติ จังหวัดสกลนคร
คณะวิทยาศาสตร์และวิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์



Web Service

CRUD

SOAP

JSON

REST

HTTP

RESTful

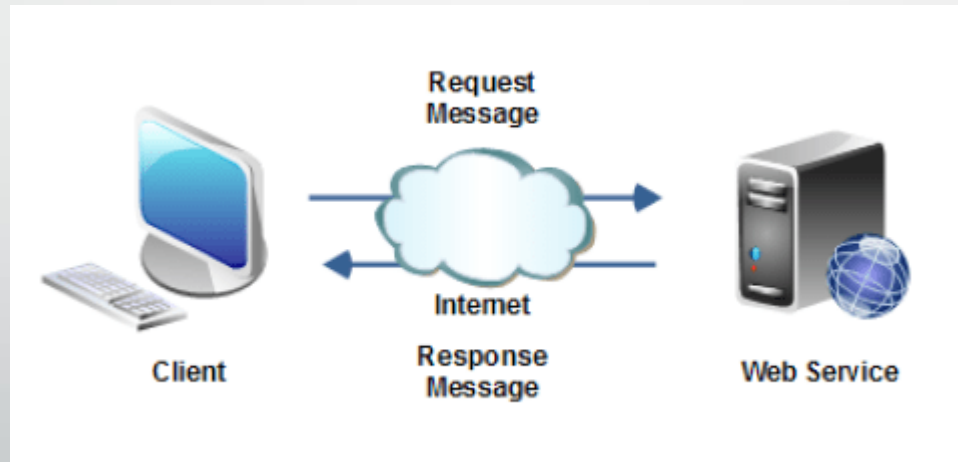
XML

API

What is Web Services ?

Web service (เว็บเซอร์วิส) คือการให้บริการแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์ในโลก internet ซึ่งถูกออกแบบมาตอนแรกเพื่อใช้ระหว่าง human-to-machine ผ่าน **HTTP** ต้องถูกพัฒนามาใช้กับ machine-to-machine ในรูปแบบ **XML** และ **JSON** format

ในทางเทคนิค “web service” เป็นมาตรฐานในการเชื่อมต่อ web-base application ที่วิ่งอยู่บน internet protocol โดยอาศัย XML, SOAP, WSDL และ UDDI



What is Rest ?

REST ย่อมาจาก Representational state transfer

REST เป็น “รูปแบบสถาปัตยกรรม” ที่ใช้ประโยชน์จากเทคโนโลยี Web protocol เพื่อใช้ในการสร้าง Web Service.



<https://medium.com/@iamgique/restful-api-%E0%B8%81%E0%B8%B1%E0%B8%9A-rest-api-%E0%B8%95%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B8%81%E0%B8%B1%E0%B8%99%E0%B8%99%E0%B8%Bo%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B8%A2%E0%B8%B1%E0%B8%87-2c70c42990e3>

Architectural properties

ข้อกำหนดของ REST architectural style ทั้ง 7 ประกอบไปด้วย

Performance : ประสิทธิภาพในการโต้ตอบ

Scalability : มีความสามารถในการปรับขยายได้ทำให้สามารถรองรับส่วนประกอบจำนวนมาก

Simplicity : ความเรียบง่ายของอินเทอร์เฟซที่เหมือนกัน

Modifiability : ความสามารถในการปรับเปลี่ยนส่วนประกอบเพื่อตอบสนองความต้องการที่เปลี่ยนแปลง (แม้ในขณะที่แอปพลิเคชันทำงานอยู่)

Visibility : การมองเห็นการสื่อสารระหว่าง Component ผ่าน Service agents

Portability : ง่ายต่อการโยกย้ายในส่วนของ Program และ Data

Reliability : ความน่าเชื่อถือในการต้านทานความล้มเหลวในระบบ

RESTful Web Services (RWS)

- เป็น Web Service ที่ใช้ REST architectural style เป็นที่รู้จักกันในชื่อ RESTful web services (RWS)
- อนุญาตให้ระบบ Request และเข้าถึง Resource บนเว็บโดยใช้ชุดคำสั่งตามที่กำหนดไว้
- การโต้ตอบของระบบที่ใช้ REST จะอยู่บนพื้นฐานของ Hypertext Transfer Protocol (HTTP)
- Request จะส่งคำขอไปยัง URI ที่กำหนด และล้วงเอา response กลับมาเป็น Payload ในแบบ HTML, XML, JSON หรือ format อื่น ๆ

JSON

```
{  
  "ID": "1",  
  "Name": "M Vaqqas",  
  "Email": "m.vaqqas@gmail.com",  
  "Country": "India"  
}
```

XML

```
<Person>  
<ID>1</ID>  
<Name>M Vaqqas</Name>  
<Email>m.vaqqas@gmail.com</Email>  
<Country>India</Country>  
</Person>
```

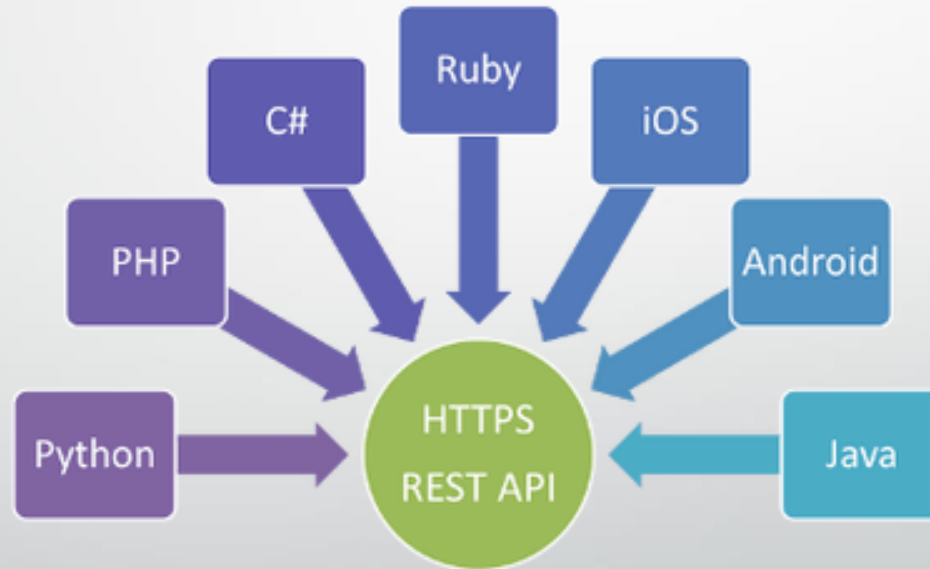
Architectural constraints of RESTful API

1. Client-Server architecture: Client ไม่จำเป็นต้องรู้อะไรเกี่ยวกับ Business logic ภายใน ไม่มีหน้าที่เกี่ยวกับการจัดเก็บข้อมูล ส่วน Server มีหน้าที่เก็บ Resource และไม่จำเป็นต้องรู้อะไรเกี่ยวกับ UI Frontend หรือสถานะของผู้เรียก
2. Stateless: Client ส่ง Request และรับ Response จาก Server แล้วจบการทำงาน
3. Cacheability: สามารถกำหนดการ cache response
4. Layered system: Server สามารถขยายระบบได้ เช่นการทำ Load balance
5. Code on demand (optional): สามารถปรับแต่งการทำงานของไคลเอนต์ได้ ตัวอย่างเช่น ทำ client-side scripts ใน JavaScript
6. Uniform interface: มีรูปแบบ API ที่เรียกใช้ได้ง่าย ไม่ซับซ้อน

REST VS RESTful

REST เป็นกระบวนทางสถาปัตยกรรม หรือ architectural paradigm

RESTful เป็นการใช้อ้างอิงถึงสถาปัตยกรรม หรือ using that paradigm



API

การเขียนเว็บเซอร์วิส หรือ เว็บ API (Application Programming Interface) เพื่อให้เทคโนโลยีแพลตฟอร์มต่างๆสามารถแลกเปลี่ยนข้อมูลกันได้ ไม่ว่าจะเป็น OS ,Web ,Mobile Application นั้นมี 2 มาตรฐานที่ได้รับความนิยม ได้แก่

SOAP (Simple Object Access Protocol)

↳ ใช้ XML format ส่งข้อมูล

REST (Representational state transfer)

↳ ใช้ XML หรือ JSON format ส่งข้อมูล



การใช้งาน APIs

1. Libraries and frameworks
2. Operating Systems
3. Remote APIs
4. Web APIs

ตัวอย่าง API ที่นิยมในปัจจุบัน

[Google Maps API](#)

[YouTube APIs](#)

[Twitter APIs](#)

[Amazon Product Advertising API](#):



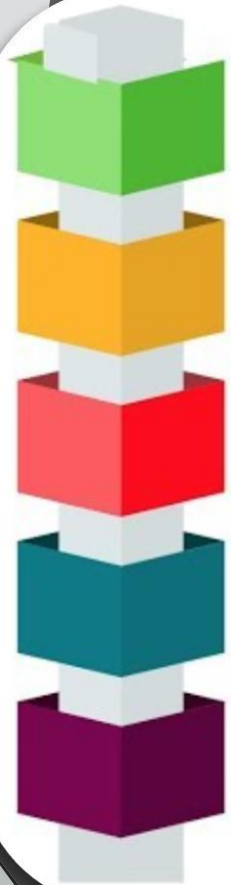
HTTP verbs

ใช้ HTTP Methods ที่มีวิธีการเดียวกับ **CRUD** operation ดังนี้

- **Create** operation ใช้ HTTP method **POST** สำหรับสร้างข้อมูล
- **Read** operation ใช้ HTTP method **GET** สำหรับดึงข้อมูล
- **Update** operation ใช้ HTTP method **PUT** สำหรับแก้ไขข้อมูล
- **Delete** operation ใช้ HTTP method **DELETE** สำหรับลบข้อมูล



HTTP Status Codes



1XX
INFORMATIONAL

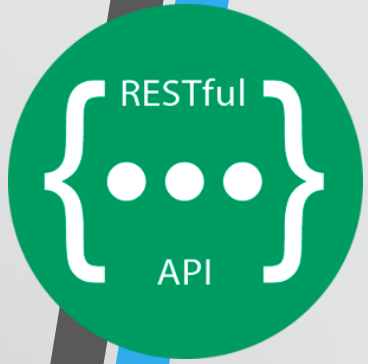
2XX
SUCCESS

3XX
REDIRECTION

4XX
CLIENT ERROR

5XX
SERVER ERROR

GET	200 Ok
Application Error	400 Bad Request
Resource not available	404 Not found
Server Error	500 Internal Server
Authorization	401 Unauthorized
Authorized but not allowed	403 Forbidden
Method Invalid	405 Method not allowed
Content Invalid	406 Not Accepted



How to create a REST API ?

Install

1. Visual Studio Code
2. Node.js
3. Express.js
4. Postman



Node.js คือ Cross Platform Runtime Environment สำหรับฝั่ง Server และเป็น Open Source ซึ่งเขียนด้วยภาษา **JavaScript** ง่ายๆ NodeJS ก็คือ Platform ตัวหนึ่งที่เขียนด้วย JavaScript สำหรับเป็น Web Server (เดิมทีภาษา JavaScript ทำงานฝั่ง Client เป็นหลัก)

<https://nodejs.org/en/>

NodeJs เป็นรูปแบบการทำงานของ JavaScript ที่เป็น **asynchronous event driven** หรือก็คือการพิจารณาทำคำสั่งใดๆ ให้ขึ้นกับเหตุการณ์หรือ Event ที่กำหนด โดยไม่จำเป็นต้องประสานต่อเนื่องกันตามลำดับ

ตัวอย่างการทำงานแบบ synchronous และ asynchronous

แบบ **synchronous**

ส่งข้อมูลไปประมวลผลที่ server -> รอข้อมูลจนกว่าข้อมูลจะถูกส่งกลับ -> ข้อมูลส่งกลับมา
-> นำข้อมูลไปแสดง -> *(ทำงานคำสั่งถัดไป) แบบนี้ตามลำดับไปอย่างต่อเนื่อง

แบบ **asynchronous**

ส่งข้อมูลไปประมวลผลที่ server -> แล้วรอข้อมูล -> *(ในขณะเดียวกันทำงานคำสั่งถัดไปเลย)
-> ข้อมูลส่งกลับจาก server -> นำข้อมูลไปแสดง

ใน NodeJs จะเรียกรูปแบบการทำงานทั้งสองว่า Blocking และ Non-Blocking ตามลำดับ
ตัวอย่างการใช้ File System module ในรูปแบบ synchronous file read

```
1 const fs = require('fs'); // เรียกใช้ file system module
2 const data = fs.readFileSync('/file.md'); // block หรือ รอจนกว่าจะอ่านไฟล์เรียบร้อย
3 console.log(data);
4 moreWork(); // ทำงานหลังคำสั่ง console.log
```

กรณีแรกนี้คำสั่ง console.log จะทำงานก่อน moreWork()
และในรูปแบบ asynchronous

```
1 const fs = require('fs'); // เรียกใช้ file system module
2 fs.readFile('/file.md', (err, data) => {
3   if (err) throw err;
4   console.log(data);
5 });
6 moreWork(); // will run before console.log
```

กรณีที่สองคำสั่ง fs.readFile() จะเป็นแบบ non-blocking มีการใช้งาน callback function หรือก็คือ กำหนดฟังก์ชัน เมื่อทำการอ่านข้อมูลเรียบร้อยแล้ว หมายความว่าเราไม่จำเป็นต้องรอให้ข้อมูลอ่านเสร็จเรียบร้อย ก็สามารถทำคำสั่ง JavaScript ในลำดับถัดไปซึ่งก็คือ moreWork() ได้เลย



npm (Node Package Manager) คือตัวช่วยที่คอยทำหน้าที่จัดการกับ Package ของ Node ให้เรา อย่างเช่น application/package หรือ module ต่างๆ มีคนเขียนไว้มากมาย เราไม่จำเป็นต้องไปเขียนซ้ำอีกรอบ แค่นำ package นั้นๆของคนอื่นมาใช้งาน ผ่านทาง npm ง่ายๆ เช่น

```
npm install PACKAGE_NAME
```

เพียงแค่นี้ package/module ของเราก็จะถูกติดตั้งไว้ที่โฟลเดอร์ node_modules ภายในโปรเจกต์ของเรา

<https://www.npmjs.com/>

npm is the package manager for javascript.

 **170,065**
total packages

 **83,860,131**
downloads in the last day

 **464,268,262**
downloads in the last week

 **1,908,347,670**
downloads in the last month

packages people 'npm install' a lot



browserify
browser-side require() the node way
10.2.6 published 3 weeks ago by substack

express

express
Fast, unopinionated, minimalist web framework
4.13.1 published 4 weeks ago by dougwilson



pm2
Production process manager for Node.JS appl...
0.14.3 published 4 weeks ago by jshkurti



grunt-cli
The grunt command line interface.
0.1.13 published 2 years ago by tkellen



npm
a package manager for JavaScript
2.13.0 published 4 weeks ago by zkat



karma
Spectacular Test Runner for JavaScript.
0.13.1 published 2 weeks ago by dignifiedquire



bower
The browser package manager
1.4.1 published 4 months ago by sheerun



cordova
Cordova command line interface tool
5.1.1 published 2 months ago by stevegill



coffee-script
Unfancy JavaScript
1.9.3 published 2 months ago by jashkenas



gulp
The streaming build system
3.9.0 published 2 months ago by phated



forever
A simple CLI tool for ensuring that a given nod...
0.14.2 published 4 weeks ago by indexzero

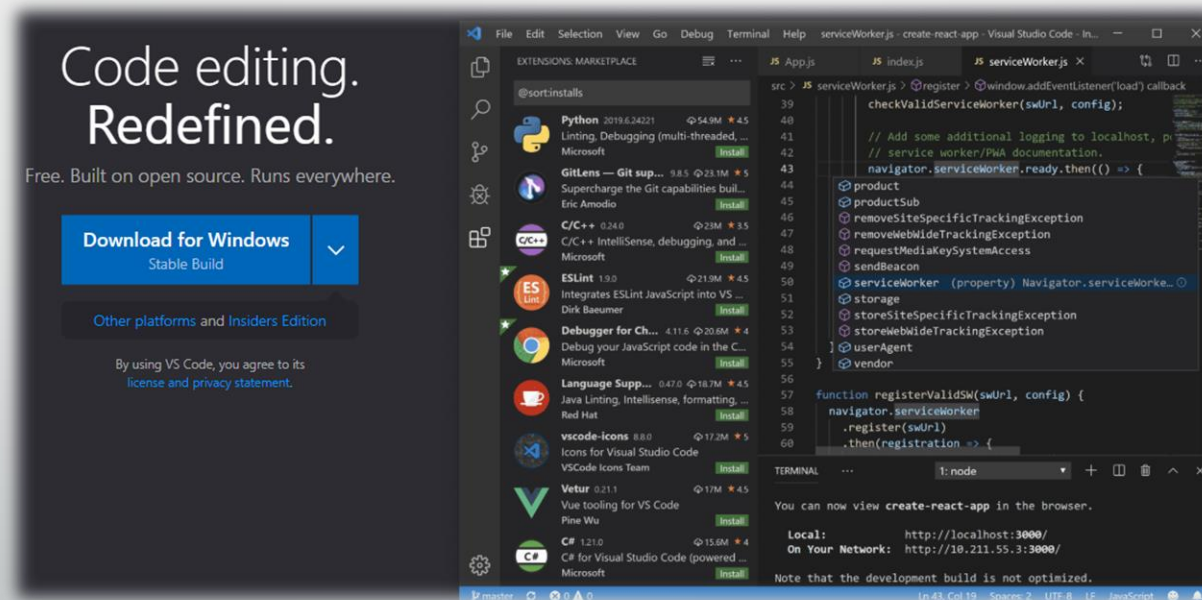


statsd
A simple, lightweight network daemon to coll...
0.7.2 published 11 months ago by pkhzzrd



Visual Studio Code

Visual Studio Code เป็น Editor สมัยใหม่ โดยมีคุณสมบัติเด่น คือ สามารถใช้งานได้ฟรี, Cross-platform และมีตัวจัดการ Extensions หรือส่วนเสริมต่างๆ เพื่อช่วยให้นักพัฒนาเขียนโค้ดได้ง่ายขึ้น

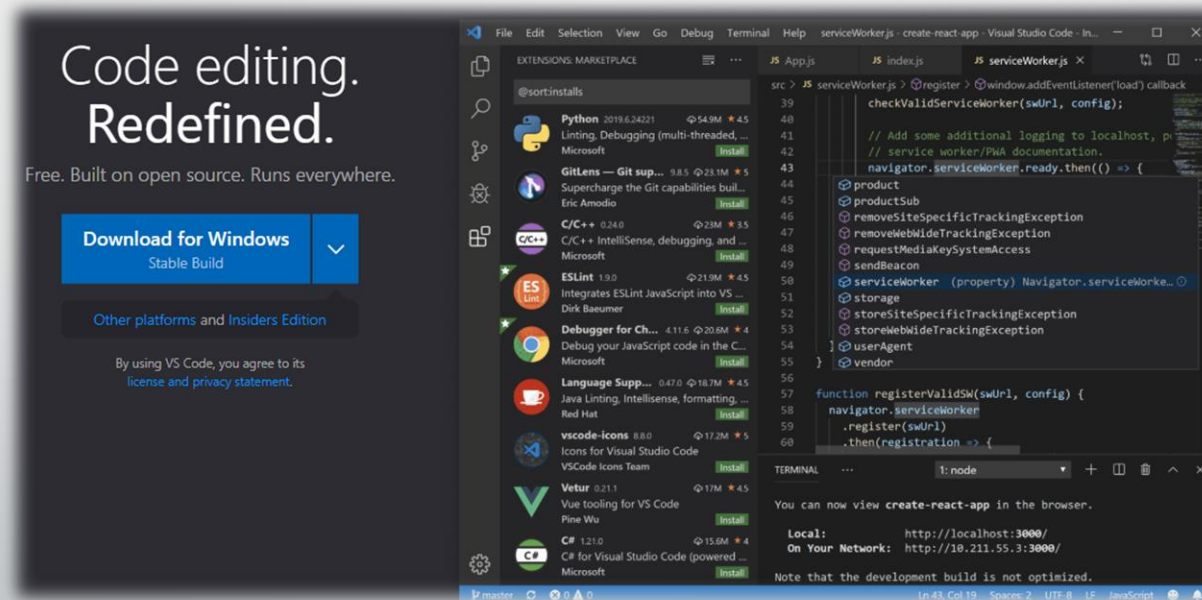


<https://code.visualstudio.com/>

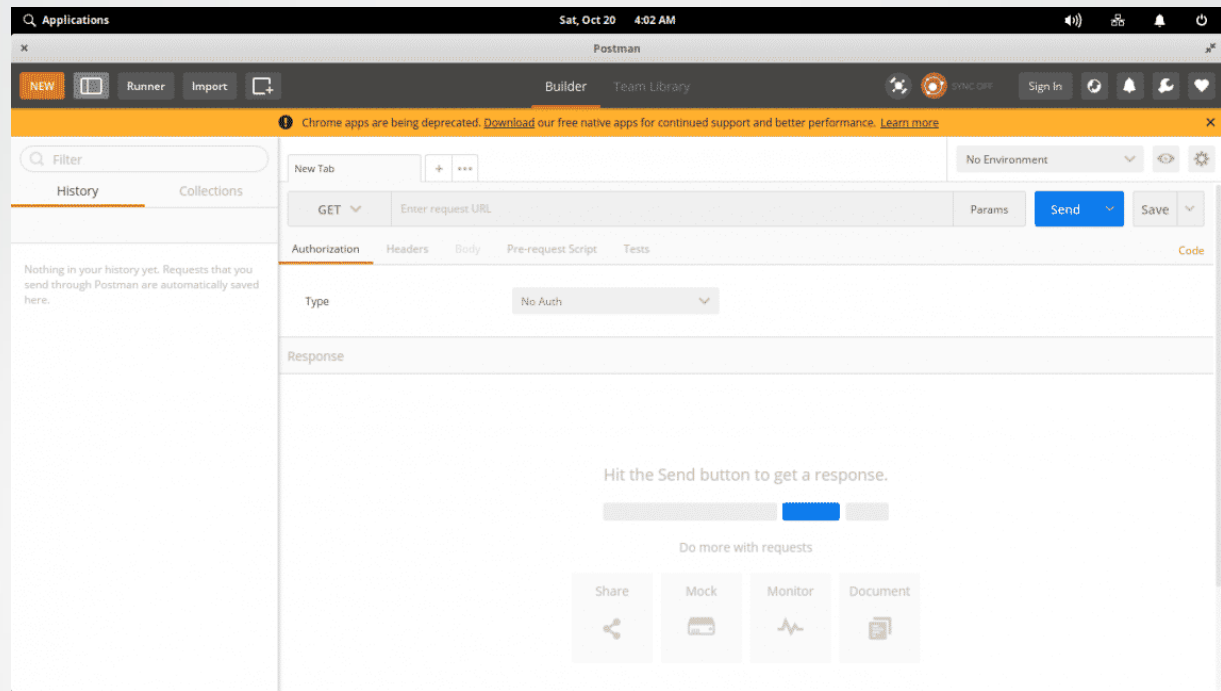


Visual Studio Code

Visual Studio Code เป็น Editor สมัยใหม่ โดยมีคุณสมบัติเด่น คือ สามารถใช้งานได้ฟรี, Cross-platform และมีตัวจัดการ Extensions หรือส่วนเสริมต่างๆ เพื่อช่วยให้นักพัฒนาเขียนโค้ดได้ง่ายขึ้น



<https://code.visualstudio.com/>



หรือใช้ Chrome Extension -> Tabbed Postman - REST Client

