

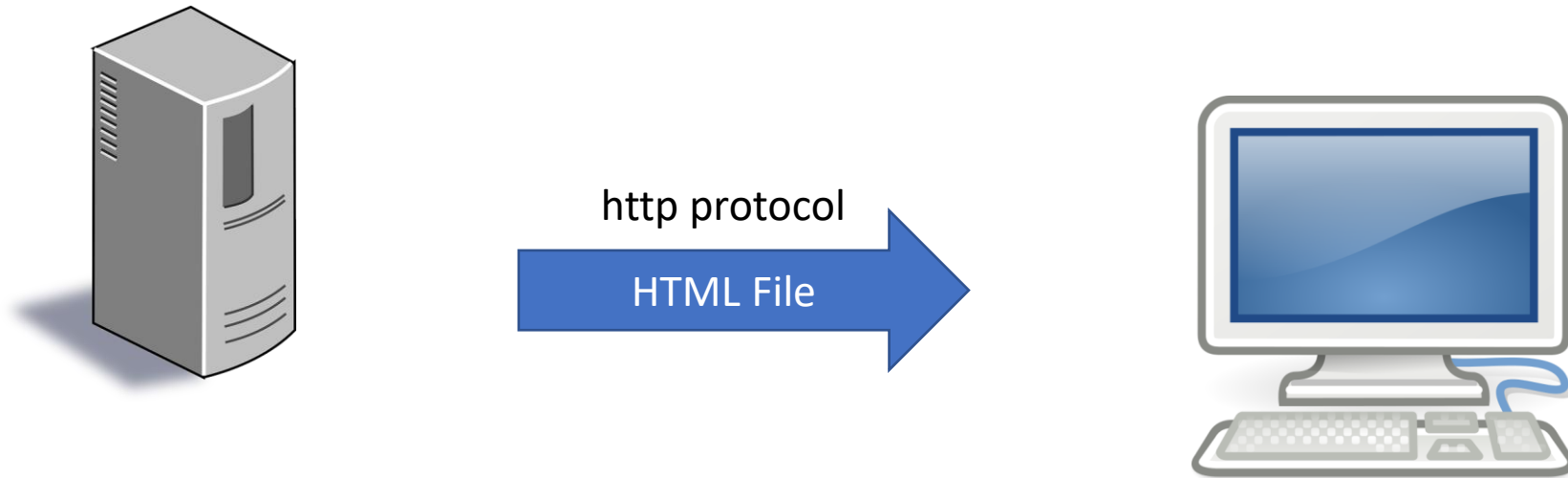
REST API

Fetch function

- ใช้ในการดึง **resource** มาใช้ในการแสดงผล
- การอ้างอิง **resource** ทำได้สองรูปแบบ
 - Relative path
 - อ้างอิงกับตำแหน่ง ไฟล์ในปัจจุบัน
 - ขึ้นด้วย ชื่อ folder, ./ หรือ .//
 - Absolute path
 - เรียกผ่าน ชื่อที่ชัดเจน
 - เช่น C://....
 - เช่น http:// ...

https การเรียกผ่าน http protocol

- Hypertext Transfer protocol



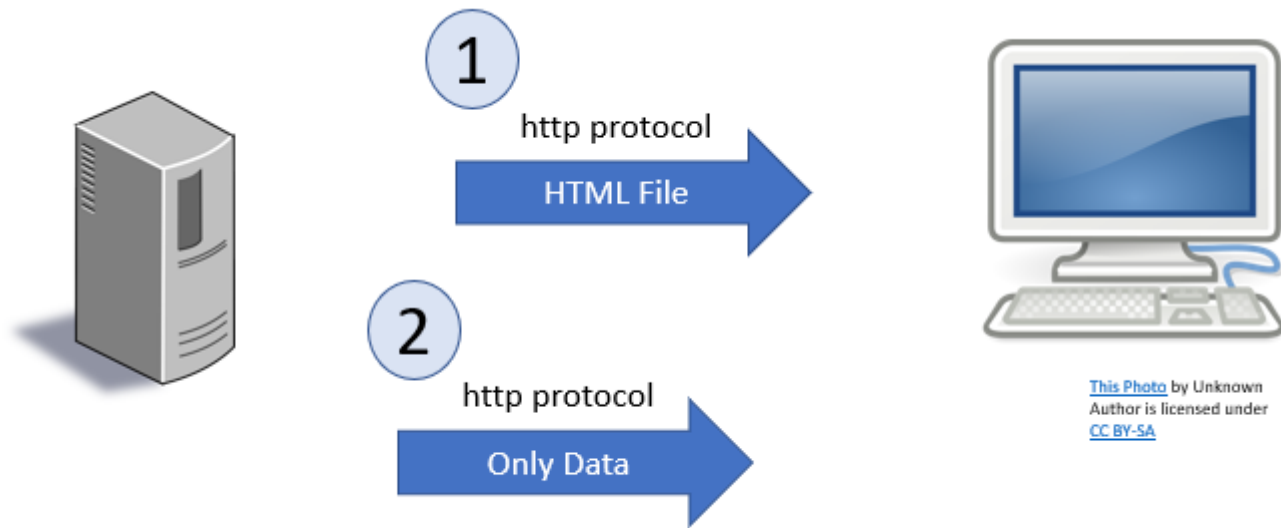
[This Photo](#) by Unknown
Author is licensed under
[CC BY-SA](#)

Dynamic page

- เมื่อเราเรียก url เดิม
 - เนื้อหาจะมีการเปลี่ยนแปลงไปเรื่อยๆ
- เช่นการเขียน php
 - เรียก url เดิม ข้อมูลจะถูกโหลดทั้งหน้า
 - แต่จริงๆ แล้วมีแค่บางเนื้อหาที่ถูกเพิ่มเข้ามา

ลดปริมาณการส่งข้อมูล

- Single Page application



การที่ส่งแต่ข้อมูล

- ทำให้การส่งข้อมูลมีขนาดเล็กลง
- ใช้ **Java Script** เพื่อจัดการส่วนของข้อมูลที่ถูกส่งออกมา

API

- Application Programming Interface
- ที่ที่เป็นประตูในการ รับส่งข้อมูล
- เราไม่ต้องสนใจว่ามันทำงานอย่างไร
- สนแค่ขอข้อมูลไปอย่างไร และจะได้อะไรกลับมา
- บางครั้งเรียกว่า **Endpoint**

Data Format

- รูปแบบของข้อมูล
 - SOAP, Json

Json

- รูปแบบที่ใช้กันเป็นหลัก
- ลักษณะคล้าย **object** ของ **java script**

```
{  
  "name":"John",  
  "age":30,  
  "car":null  
}
```

HTTP Endpoint

- เรียกข้อมูลผ่าน **http protocol**
- ปกติเป็น **text** ธรรมดา แต่ใส่ในรูปแบบที่กำหนดไว้
- **Protocol**
 - คือรูปแบบที่กำหนดไว้ว่าใครจะส่งอะไรอย่างไร

HTTP Protocol

- an application-level protocol for distributed, collaborative, hypermedia information systems
- Basic Feature
 - Connectionless: Do not care will they receive data or not
 - Media independent: any types of data can be sent
 - Stateless: no state, does not know what has been called

HTTP methods

- วิธีการที่จะส่งข้อมูลผ่าน URL
- GET methods
 - ใช้ส่งข้อมูลผ่านตัว URL
 - ส่งเฉพาะข้อมูลที่ไม่กระทบข้อมูลอื่น
 - ใช้ Query เป็นหลัก
- URI
 - Universal Resource Identifier
 - url (Universal Resource Locator)

GET methods

- Data send via url
- Data attached to some part of the application
- <http://www.camt.cmu.ac.th>
 - Only url is send

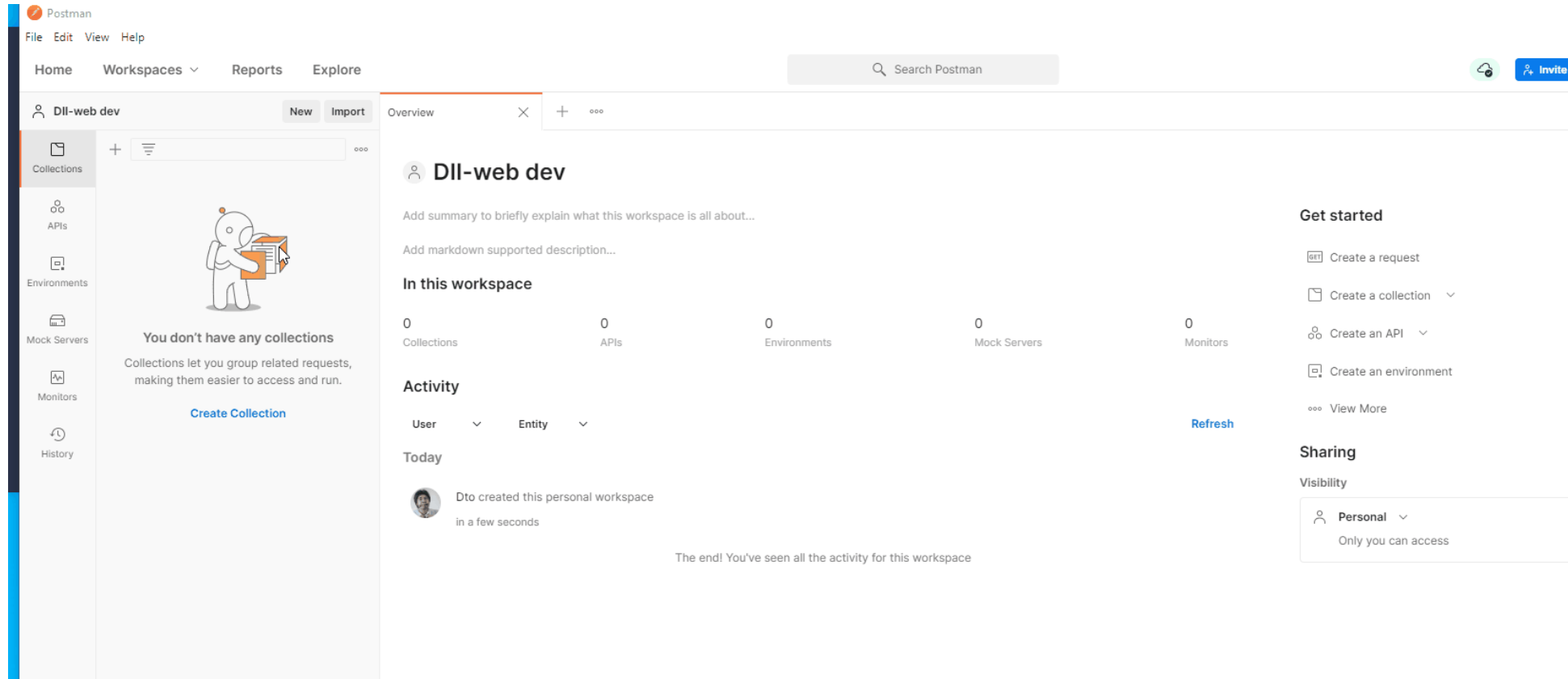
REST API

- Representational state transfer
- Application Programming Interface
- **Format** ของการเรียกข้อมูลผ่าน **http protocol**
 - ผ่าน **url**
 - รับข้อมูลเป็น **json**

เพื่อไม่ให้เป็นการเสียเวลา

- เปิดโปรแกรม **postman** กัน
- **Postman** เป็นโปรแกรมสำหรับการทดลองเรียกข้อมูลผ่าน **REST API**
- เพื่อดูหน้าตาเบื้องต้นของข้อมูล

ทดลองเรียกข้อมูลผ่าน endpoint



- เสร็จแล้วกดปุ่ม **send** แล้วรอดูผล

ผลลัพธ์ที่ได้

```
{
  "id": 1,
  "studentId": "SE-001",
  "name": "Prayuth",
  "surname": "The minister",
  "gpa": 3.59,
  "image": "https://firebasestorage.googleapis.com/v0/b/test-331-lab05.appspot.com/o/tu.jpg?alt=media&token=f16b5c1c-fbea-4d98-9fa3-732b69a1bae8",
  "penAmount": 15,
  "description": "The great man ever!!!!"
}
```

ทดลองนำข้อมูลมาใส่

- จากหน้าเดิมที่เคยทำคร่าวก่อน
- เพิ่ม **code** เพื่อแสดงผลนักเรียน หนึ่งในที่ด้านล่างของ **table** รูปแบบนี้ เพื่อตั้งต้นตำแหน่งที่ต้องการนำข้อมูลมาแสดง
- ลบ **onload** เก่าทิ้งด้วย
- อาจจะ **commit code** เก่าไว้เผื่อมาดูเป็นตัวอย่างในอนาคต

เพิ่มส่วน output

```
        </tbody>
    </table>
</div>
<div id="sinigle_student_result" class="px-3 m-5">
    <div class="row">
        <div class="col-3"><b>Id:</b></div>
        <div class="col-3" id="id">id</div>
    </div>
    <div class="row">
        <div class="col-3"><b>Student id:</b></div>
        <div class="col-3" id="studentId">id</div>
    </div>
    <div class="row">
        <div class="col-3"><b>Name:</b></div>
        <div class="col-3" id="name">name and surname</div>
    </div><div class="row">
        <div class="col-3"><b>GPA:</b></div>
        <div class="col-3" id="gpa">gpa</div>
    </div>
    <div class="row">
        <div class="col-3"><b>Id:</b></div>
        <div class="col">
            
        </div>
    </div>
</div>
</div>
```

เพิ่ม function การเพิ่มข้อมูล

```
function addStudentData(student){  
  let idElem = document.getElementById('id')  
  idElem.innerHTML = student.id  
  let studentIdElem = document.getElementById('studentId')  
  studentIdElem.innerHTML = student.studentId  
  let nameElem = document.getElementById('name')  
  nameElem.innerHTML = `${student.name} ${student.surname}`  
  let gpaElem = document.getElementById('gpa')  
  gpaElem.innerHTML = student.gpa  
  let profileElem = document.getElementById('image')  
  profileElem.setAttribute('src', student.image)  
}
```

```
</tbody>  
</table>  
</div>  
<div id="single_student_result" class="px-3 m-5">  
  <div class="row">  
    <div class="col-3"><b>Id:</b></div>  
    <div class="col-3" id="id">id</div>  
  </div>  
  <div class="row">  
    <div class="col-3"><b>Student id:</b></div>  
    <div class="col-3" id="studentId">id</div>  
  </div>  
  <div class="row">  
    <div class="col-3"><b>Name:</b></div>  
    <div class="col-3" id="name">name and surname</div>  
  </div><div class="row">  
    <div class="col-3"><b>GPA:</b></div>  
    <div class="col-3" id="gpa">gpa</div>  
  </div>  
  <div class="row">  
    <div class="col-3"><b>Id:</b></div>  
    <div class="col">  
        
    </div>  
  </div>  
</div>
```

เพิ่ม onLoadFunction

```
function onLoad(){  
  fetch('https://dv-student-backend-2019.appspot.com/student')  
  .then((response) => {  
    return response.json()  
  }).then(data => {  
    addStudentData(data)  
  })  
}
```

คู่มือ

ลองดู endpoint ที่รับ ค่าเยอะๆ

- สร้าง request ใหม่ใน postman
- ใส่ endpoint
 - <https://dv-student-backend-2019.appspot.com/students>

output

```
[
  {
    "id": 1,
    "studentId": "SE-001",
    "name": "Prayuth",
    "surname": "The minister",
    "gpa": 3.59,
    "image": "https://firebasestorage.googleapis.com/v0/b/test-331-lab05.appspot.com/o/tu.jpg?alt=media&token=f16b5c1c-fbea-4d98-9fa3-732b69a1bae8",
    "penAmount": 15,
    "description": "The great man ever!!!"
  },
  {
    "id": 2,
    "studentId": "SE-002",
    "name": "Cherprang ",
    "surname": "BNK48",
    "gpa": 4.01,
    "image": "https://firebasestorage.googleapis.com/v0/b/test-331-lab05.appspot.com/o/cherprang.png?alt=media&token=2e6a41f3-3bf0-4e42-ac6f-8b7516e24d92",
    "penAmount": 2,
    "description": "Code for Thailand"
  },
  {
    "id": 3,
    "studentId": "SE-003",
    "name": "Nobi",
    "surname": "Nobita",
    "gpa": 1.77,
    "image": "https://firebasestorage.googleapis.com/v0/b/test-331-lab05.appspot.com/o/nobita.jpg?alt=media&token=16e30fb0-9904-470f-b868-c35601df8326",
    "penAmount": 0,
    "description": "Welcome to Olympic"
  }
]
```


สามารถจัดการเหมือน list ได้เลย

- แต่ต้องหลังจากรับค่าผ่าน **promise** นะ
- ปรับส่วนของ **addStudentToTable** เพื่อให้แสดงผลให้ตรงกับ **object** ที่เรารับมา

```
function addStudentToTable(index, student) {  
  const tableBody = document.getElementById('tableBody')  
  let row = document.createElement('tr')  
  let cell = document.createElement('th')  
  cell.setAttribute('score', 'row')  
  cell.innerHTML = index  
  row.appendChild(cell)  
  cell = document.createElement('td')  
  cell.innerHTML = student.name  
  row.appendChild(cell)  
  cell = document.createElement('td')  
  // cell.innerHTML = student.username  
  let img = document.createElement('img')  
  img.setAttribute('src', student.imageLink )  
  cell.appendChild(img)  
  row.appendChild(cell)  
  cell = document.createElement('td')  
  cell.innerHTML = student.gender  
  row.appendChild(cell)  
  tableBody.appendChild(row)  
}
```



```
function addStudentToTable(index, student) {  
  const tableBody = document.getElementById('tableBody')  
  let row = document.createElement('tr')  
  let cell = document.createElement('th')  
  cell.setAttribute('score', 'row')  
  cell.innerHTML = index  
  row.appendChild(cell)  
  cell = document.createElement('td')  
  cell.innerHTML = `${student.name} ${student.surname}`  
  row.appendChild(cell)  
  cell = document.createElement('td')  
  let img = document.createElement('img')  
  img.setAttribute('src', student.image)  
  cell.appendChild(img)  
  row.appendChild(cell)  
  cell = document.createElement('td')  
  cell.innerHTML = student.gender  
  row.appendChild(cell)  
  tableBody.appendChild(row)  
}
```

ปรับส่วน onLoad

```
function onLoad(){  
  fetch('https://dv-student-backend-2019.appspot.com/students')  
  .then((response) => {  
    return response.json()  
  }).then(data => {  
    addStudentList(data)  
  })  
}
```

ภาพออกมาไม่เท่ากัน

- ปรับภาพด้วย java script

```
function addStudentToTable(index, student) {  
    const tableBody = document.getElementById('tableBody')  
    let row = document.createElement('tr')  
    let cell = document.createElement('th')  
    cell.setAttribute('score', 'row')  
    cell.innerHTML = index  
    row.appendChild(cell)  
    cell = document.createElement('td')  
    cell.innerHTML = `${student.name} ${student.surname}`  
    row.appendChild(cell)  
    cell = document.createElement('td')  
    let img = document.createElement('img')  
    img.setAttribute('src', student.image)  
    img.height = 200  
    cell.appendChild(img)  
    row.appendChild(cell)  
    cell = document.createElement('td')  
    cell.innerHTML = student.gender  
    row.appendChild(cell)  
    tableBody.appendChild(row)  
}
```

เพิ่ม class เพื่อความหรูหรา

```
function addStudentToTable(index, student) {  
  const tableBody = document.getElementById('tableBody')  
  let row = document.createElement('tr')  
  let cell = document.createElement('th')  
  cell.setAttribute('score', 'row')  
  cell.innerHTML = index  
  row.appendChild(cell)  
  cell = document.createElement('td')  
  cell.innerHTML = `${student.name} ${student.surname}`  
  row.appendChild(cell)  
  cell = document.createElement('td')  
  let img = document.createElement('img')  
  img.setAttribute('src', student.image )  
  img.height = 200  
  img.classList.add('img-thumbnail')  
  cell.appendChild(img)  
  row.appendChild(cell)  
  cell = document.createElement('td')  
  cell.innerHTML = student.gender  
  row.appendChild(cell)  
  tableBody.appendChild(row)  
}
```

Class override attribute

- เนื่องจาก **class** ไปแก้ไขค่าที่ระบุไว้ใน **attribute**
- ลองแก้ปัญหาดู

Query parameter

- <http://www.camt.cmu.ac.th?name=hello>
 - Data ที่ส่งอยู่ข้างหลัง “?”
 - The key and value is sent
 - In this case key is name, value is hello
 - Server รับข้อมูลแล้วส่งกลับมา
- For more data
 - <http://www.camt.cmu.ac.th?name=hello&surname=xxx>
 - & ใช้เมื่อต้องการส่งข้อมูลมากกว่า 1 ตัว

ลองเปิด postman

- [https://dv-student-backend-2019.appspot.com/studentQuery?id=\[id number\]](https://dv-student-backend-2019.appspot.com/studentQuery?id=[id number])
- <https://dv-student-backend-2019.appspot.com/studentQuery?id=2>
 - แล้วดูผลลัพธ์

Path variable

- <http://www.camt.cmu.ac.th/hello/name>
 - **Server** จะทราบเองว่าตรงไหนที่เป็น ตัวแปรที่ส่งเข้ามา
 - ต้องศึกษาจาก **api document**
- สามารถรับค่าที่แตกต่างกันได้ ในตำแหน่งของ **path** นั้น
 - <http://www.camt.cmu.ac.th/x/y>
 - Change the value in the position of the path

ลองเปิด postman

- [https://dv-student-backend-2019.appspot.com/student/\[id number\]](https://dv-student-backend-2019.appspot.com/student/[id number])
- <https://dv-student-backend-2019.appspot.com/student/3>
-
- แล้วดูผลลัพธ์

ทดลองรับ id แล้วกัน

- ลองลบ **function on load** ก่อน เพื่อจะได้ไม่ต้อง load อะไรมา
- เพิ่ม การ **click** ลงในปุ่ม
 - ตรวจสอบ id ของ **element** ที่ต้องใช้

```
<form class="col-12 col-lg-auto mb-3 mb-lg-0 me-lg-3">  
  <input type="search" class="form-control" placeholder="Search..." aria-label="Search" id="inputText">  
</form>  
<button type="button" class="btn btn-light text-dark me-2" id="searchButton">Search</button>
```

- เพิ่ม **listener** สำหรับการ **click**

```
document.getElementById('searchButton').addEventListener('click', () =>{  
  let id = document.getElementById('inputText').value  
  console.log(id)  
  fetch(`https://dv-student-backend-2019.appspot.com/student/${id}`)  
    .then(response => {  
      return response.json()  
    }).then(student => {  
      addStudentData(student)  
    })  
})
```

การบ้าน

- ลองทำแบบถ้ามีการแสดง **list** ของ นักเรียน เมื่อกดที่ชื่อของนักเรียนคนใดแล้ว ให้ด้านล่างแสดงรายละเอียดของนักเรียนคนนั้น

POST methods

- The ข้อมูลไม่ได้ถูกส่งไปจาก url
- แต่จะถูกเก็บไว้อยู่ที่ request object
- Format มีได้หลายรูปแบบ



POST request

- รูปแบบสามารถเป็นรูปภาพ ข้อความ หรือ **json object** ที่ส่งไปให้ **server**
- การกำหนดรูปแบบต้องทำที่ **header** ของ **html**
- For example
 - application/json
 - text/css
 - application/msword

POST request

- Body
 - Where the data is stored

The diagram shows an HTTP response structure with three main sections: Version, Headers, and Entity Body (Content). Blue arrows and brackets are used to group and label these sections. The Version section points to 'HTTP/1.1 201 Created', where '201' is labeled as the Status Code. The Headers section groups a list of header fields. The Entity Body (Content) section groups a JSON object representing contact data.

```
HTTP/1.1 201 Created
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Location: http://localhost:8081/api/contacts/6
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-SourceFiles: =?UTF-8?B?
QzpcQ29udGFjdE1hbmFnZXJcQyNcQ29udGFjdE1hb
X-Powered-By: ASP.NET
Date: Sat, 22 Dec 2012 21:31:04 GMT
Content-Length: 175

{
  "ContactId":6,
  "Name":"Jane User",
  "Address":"1 Any Street",
  "City":"Any City","State":"WA",
  "Zip":"00000",
  "Email":"janeuser@example.com",
  "Twitter":null,
  "Self":"/api/contacts/1"
}
```

Version: HTTP/1.1 201 Created (Status Code: 201)

Headers:

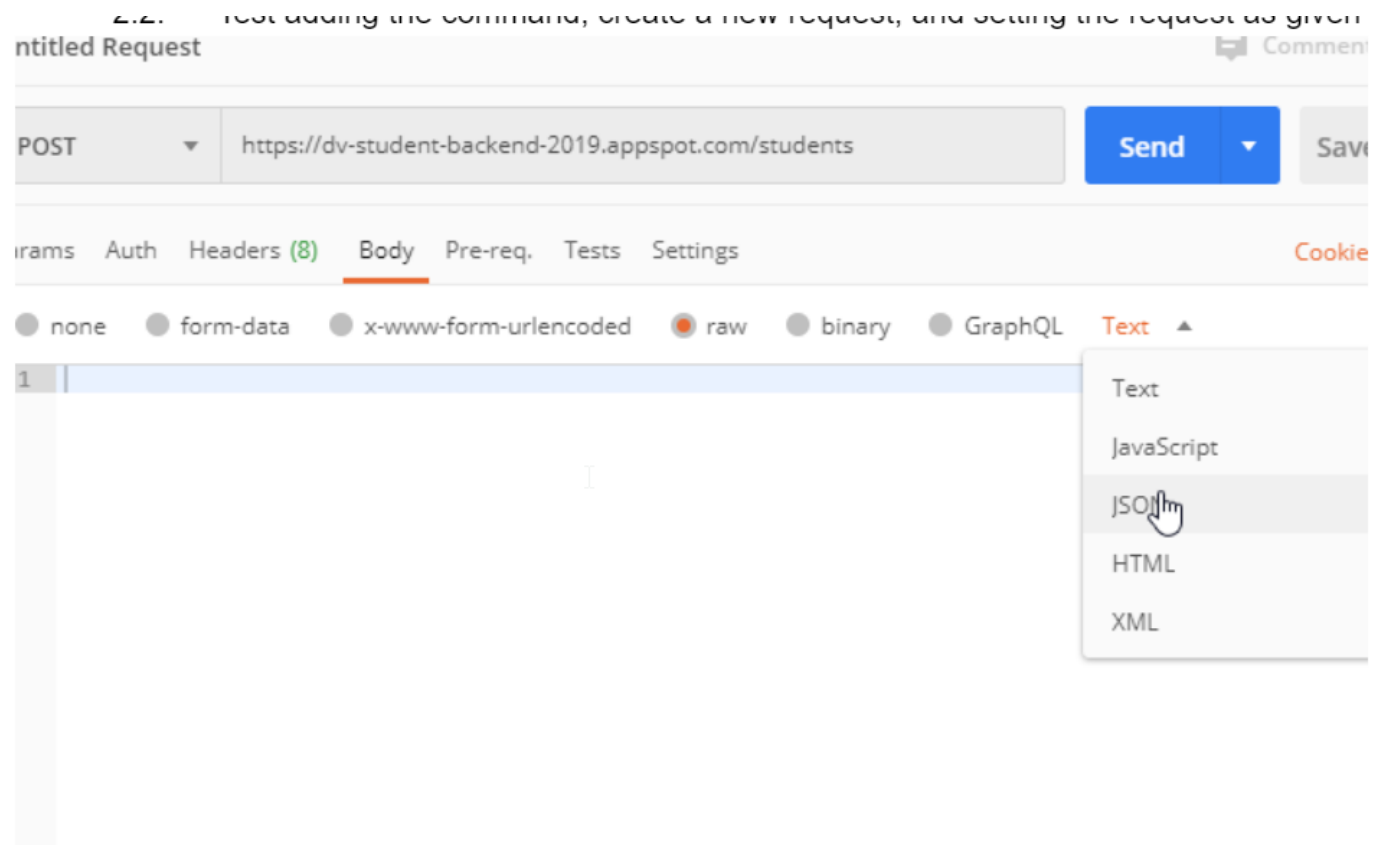
- Cache-Control: no-cache
- Pragma: no-cache
- Content-Type: application/json; charset=utf-8
- Expires: -1
- Location: <http://localhost:8081/api/contacts/6>
- Server: Microsoft-IIS/8.0
- X-AspNet-Version: 4.0.30319
- X-SourceFiles: =?UTF-8?B?
- QzpcQ29udGFjdE1hbmFnZXJcQyNcQ29udGFjdE1hb
- X-Powered-By: ASP.NET
- Date: Sat, 22 Dec 2012 21:31:04 GMT
- Content-Length: 175

Entity Body (Content):

```
{
  "ContactId":6,
  "Name":"Jane User",
  "Address":"1 Any Street",
  "City":"Any City","State":"WA",
  "Zip":"00000",
  "Email":"janeuser@example.com",
  "Twitter":null,
  "Self":"/api/contacts/1"
}
```

ทดลองส่ง post request

- ส่ง post request ตามนี้



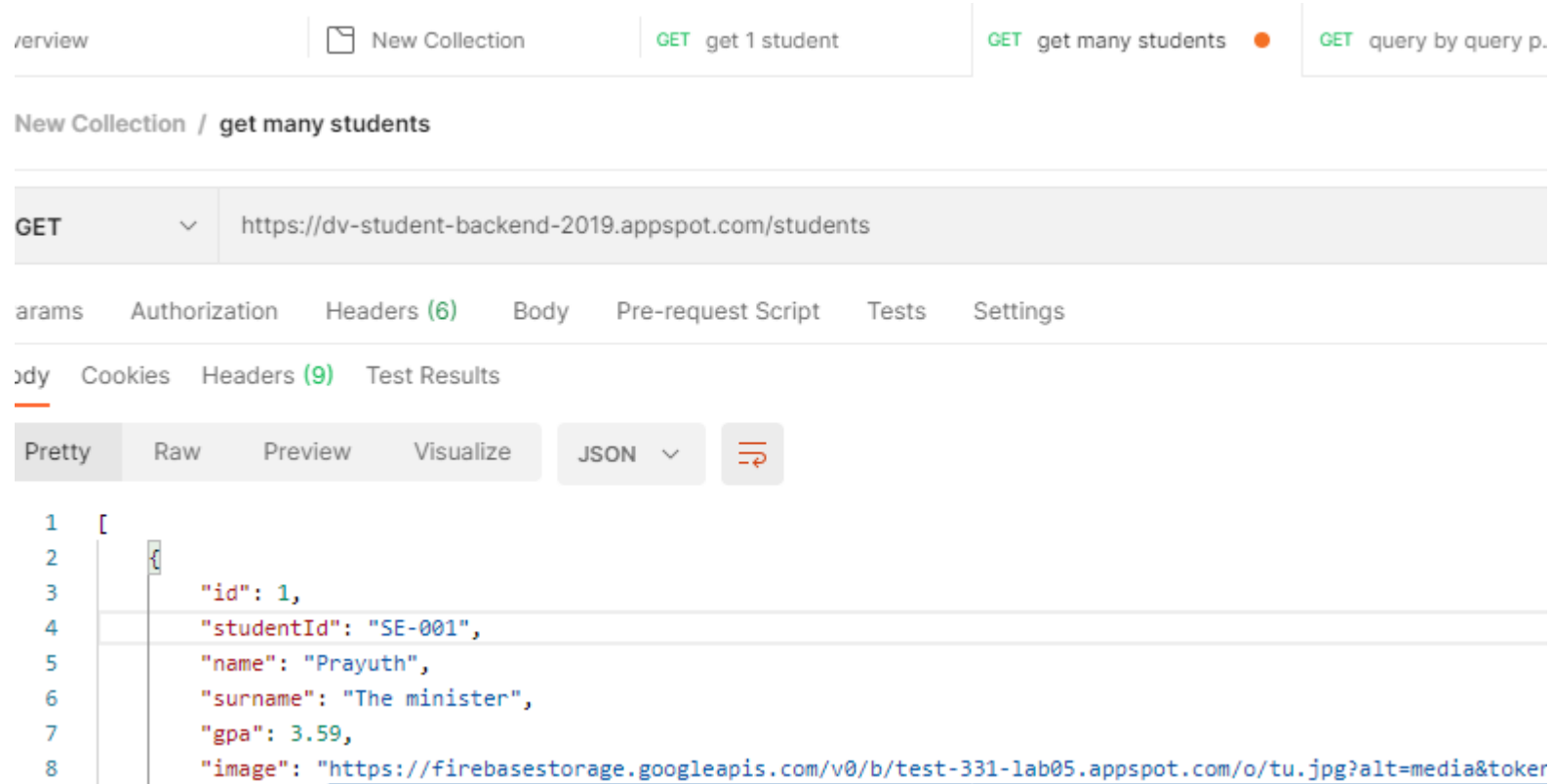
แล้วกรอกข้อมูลประมาณนี้

- รายละเอียดต่าง ๆ สามารถ ปรับได้เองตามความเหมาะสม

```
{
  .... "studentId": "66521100",
  .... "name": "Hello",
  .... "surname": "Kitty",
  .... "gpa": 2.00,
  .... "image": "https://www.appdisqus.com/wp-content/uploads/2021/03/hello-kitty.jpg"
}
```

- แล้วกดส่ง แล้วดูที่ผลลัพธ์
 - มีอะไรที่เพิ่มออกมา?

แล้วลอง search โดยดูข้อมูลทั้งหมดจาก postman อีกครั้ง



ลองปรับ code ให้แสดงผลลัพธ์ทั้งหมดอีกครั้งเมื่อ โหลด

```
function onLoad(){  
  fetch('https://dv-student-backend-2019.appspot.com/students')  
  .then((response) => {  
    return response.json()  
  }).then(data => {  
    addStudentList(data)  
  })  
}
```

ลองเพิ่มข้อมูลจากเว็บแบบ ดิบๆ

```
function addStudentToDB(student){
  fetch('https://dv-student-backend-2019.appspot.com/students',{
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body:JSON.stringify(student)
  }).then(response => {
    return response.json()
  }).then(data => {
    console.log('success',data)
  })
}
```

- ปรับปรุง **onload function** เพื่อเพิ่มข้อมูลทุกครั้งที่เปิดหน้าจอใหม่
 - ข้อมูลนักศึกษา ปรับเอาตามใจชอบ

```
function onLoad(){
  student = {
    name:"John",
    surname:"Doe",
    studentId: "6512345",
    gpa: "4.00",
    image: "asset/images/default.jpg"
  }
  addStudentToDB(student)
}
```

ดูผลลัพธ์ใน console

```
LIVE RELOAD ENABLED.
success ▼ Object
  description: null
  gpa: 4
  id: 4
  image: "asset/images/default.jpg"
  name: "John"
  penAmount: null
  studentId: "6512345"
  surname: "Doe"
  ▶ __proto__: Object
```

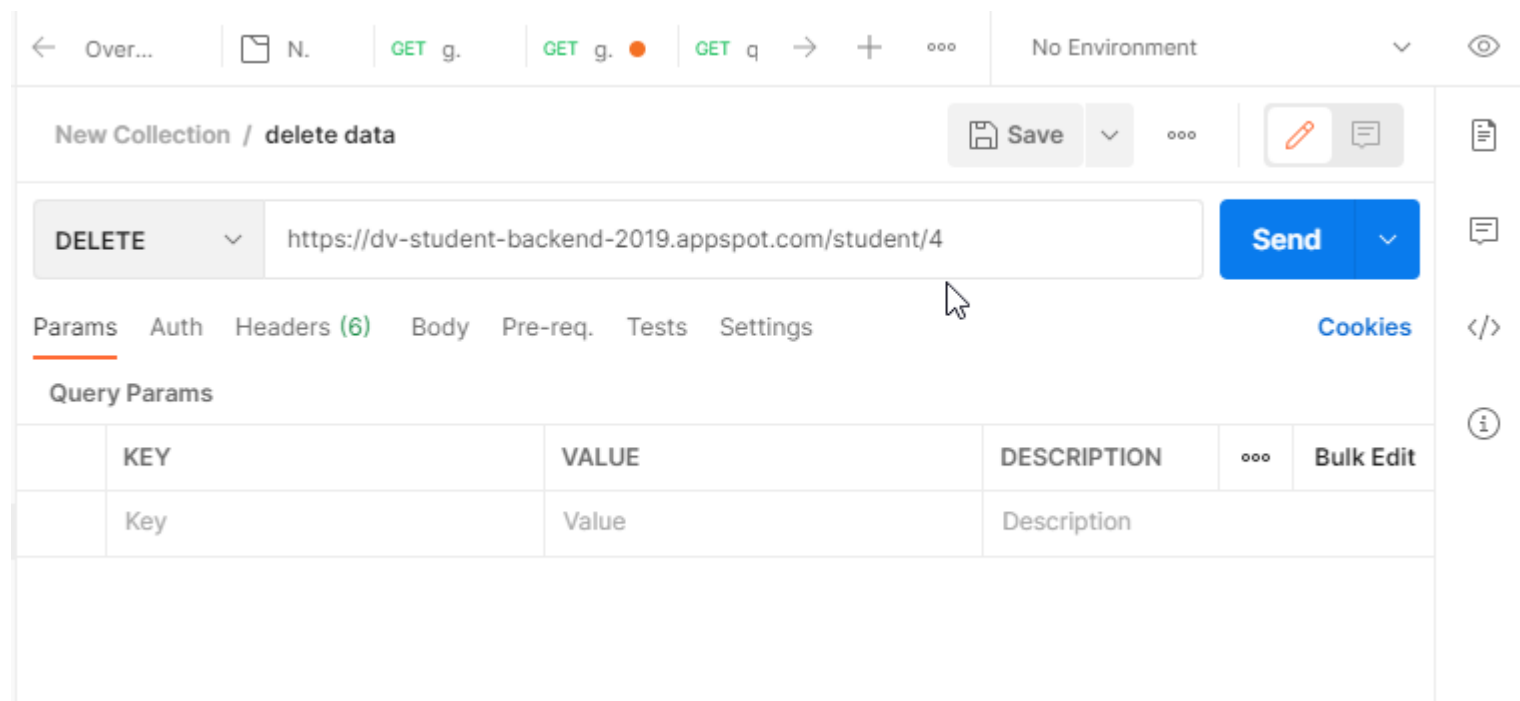
- แล้วดูใน **postman** ว่า ผลลัพธ์เข้าไปใน **db** รึยัง

เพิ่มได้ก็ ลบได้

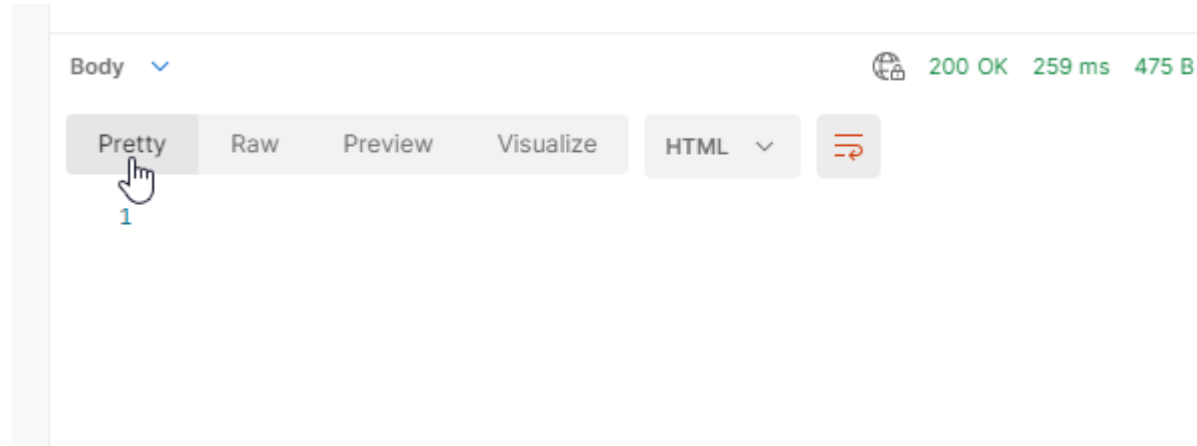
- DELETE method
 - REST api ใช้ method ในการส่งข้อมูลว่าต้องการจะทำอะไร
 - DELETE เพื่อบอกว่าจะทำอย่างไร
- ปกติ DELETE จะส่งแบบเดียวกับ GET
 - ต่างกันที่ methods
- ปกติการลบข้อมูลจากฐานข้อมูลจะใช้การลบโดยอ้างอิงจาก id ของข้อมูล

ลองทำใน postman

- สร้าง **request** ใหม่
- เลขตัวสุดท้ายใส่ **id** ของข้อมูลที่เราเพิ่มไปเมื่อีก



ดูผลลัพธ์



ลองเพิ่มข้อมูลเข้าไปใหม่

- แล้วลองเขียน code เพื่อ ลบ

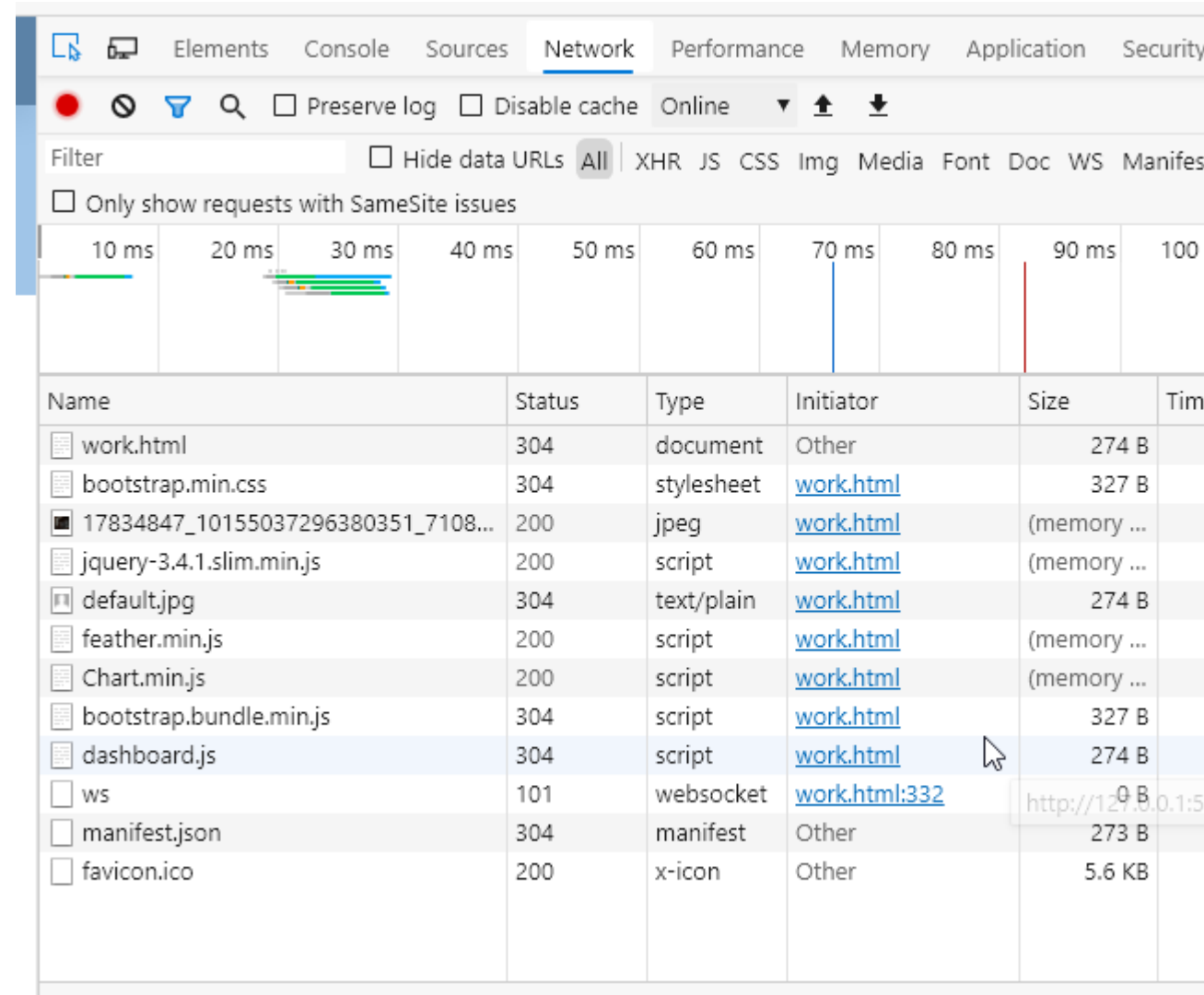
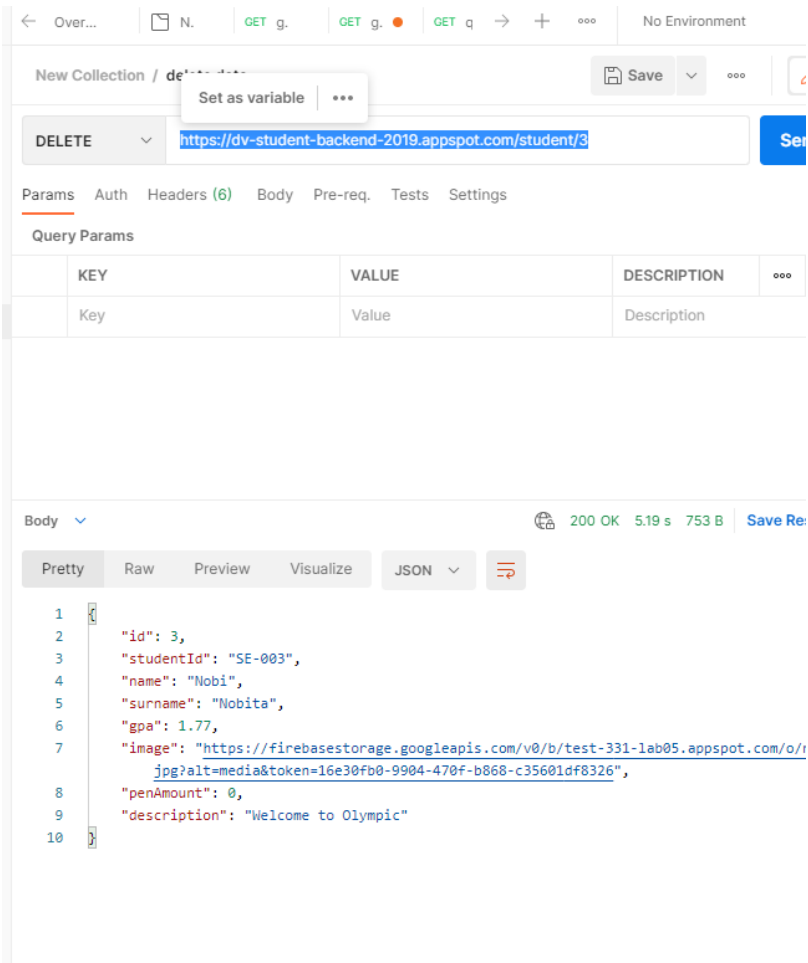
```
function deleteStudent(id) {  
  fetch(`https://dv-student-backend-2019.appspot.com/student/${id}`, {  
    method: 'DELETE'  
  }).then(response => {  
    if (response.status === 200){  
      return response.json()  
    }else{  
      throw Error(response.statusText)  
    }  
  }).then(data =>{  
    alert(`student name ${data.name} is now deleted`)  
  }).catch( error => {  
    alert('your input student id is not in the database')  
  })  
}
```

- แล้วแก้ไข load
ลองโหลดสองครั้ง

```
function onLoad(){  
  deleteStudent(4)  
}
```


Response Code

- ค่าที่แสดงว่าผลการเรียกข้อมูลเป็นอย่างไร



Class of response code

- Informational responses (100–199),
- Successful responses (200–299),
- Redirects (300–399),
- Client errors (400–499),
- Server errors (500–599).

Important code

- 200 – OK
- 304 – Not Modified
- 401 – unauthorize
- 403 - Forbidden
- 404 – not found
- 405 – Method Not Allowed
- 500 - Internal server error
- https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

การทำ Error Handler

- เขียนโปรแกรมเพื่อบอกว่าถ้าเกิดมีความไม่ถูกต้องเกิดขึ้นจะทำอย่างไร

- Error**

- สิ่งที่ทำงานได้ไม่ตรงกับ requirement
- สิ่งที่เราคิดอยู่แล้วว่าต้องเป็น error
- สามารถโยน **error** ออกมาเพื่อให้ทำงานอย่างอื่นต่อไป
- โยนได้หลาย **error** ใน code

```
function deleteStudent() {  
  let studentId = document.getElementById('studentIdDeleteForm').value;  
  fetch('https://dv-student-backend-2019.appspot.com/student/' + studentId,  
    method: 'DELETE'  
  ).then((response) => {  
    if (response.status === 200) {  
      return response.json();  
    }  
    // throw Error(response.statusText);  
  }).then((data) => {  
    alert('student name ' + data.name + ' is now deleted');  
    showAllStudents();  
  }).catch((error) => {  
    // the error is occurred here because the json()  
    // cannot return the value  
    alert('your input student id is not in the database');  
  });  
}
```

โยนมาต้องมีคนรับ

- `catch`
 - รับ `error` ที่ถูกโยนมาแล้วทำงานอย่างอย่างหนึ่ง
 - `Code` หลัง `throw` จะไม่ทำงาน

ลองเพิ่มข้อมูลจากเว็บ

- ปรับ **html** เพื่อให้รับค่าที่เกี่ยวข้อง
- ลองเลือก **form component** ที่ชอบ
- Floating Label [Floating labels · Bootstrap v5.0 \(getbootstrap.com\)](https://getbootstrap.com/docs/5.0/forms/floating-labels/)
- ลบตัวใน **form** เก่าเพื่อปรับค่าให้ตรงกับข้อมูลที่แสดง
- ใส่ **id** ด้วยนะ

```
<div class="d-flex justify-content-center my-3">
  <form class="row gy-2 gx-3 align-items-center">
    <div class="form-floating">
      <input type="text" class="form-control" id="nameInput" placeholder="ชื่อ">
      <label for="nameInput">Name</label>
    </div>
    <div class="form-floating">
      <input type="text" class="form-control" id="surnameInput" placeholder="นามสกุล">
      <label for="surnameInput">Surname</label>
    </div>
    <div class="form-floating">
      <input type="text" class="form-control" id="studentIdInput" placeholder="รหัสนักศึกษา">
      <label for="studentIdInput">รหัสนักศึกษา</label>
    </div>
    <div class="form-floating">
      <input type="text" class="form-control" id="gpaInput" placeholder="gpa">
      <label for="gpaInput">GPA</label>
    </div>
    <div class="form-floating">
      <input type="text" class="form-control" id="imageLinkInput" placeholder="image Link">
      <label for="imageLinkInput">Link รูป</label>
    </div>
    <div class="col-auto">
      <button type="button" class="btn btn-primary" id="addButton">Submit</button>
    </div>
  </form>
</div>
```

เพิ่ม code เพื่อรับข้อมูลมาพิมพ์ เมื่อกดปุ่ม addButton

- ลบคำสั่งใน onLoad ก่อนนะครับ

```
function onAddStudentClick(){  
  let student = {}  
  student.name = document.getElementById('nameInput').value  
  student.surname = document.getElementById('surnameInput').value  
  student.studentId = document.getElementById('studentIdInput').value  
  student.gpa = document.getElementById('gpaInput').value  
  student.image = document.getElementById('imageLinkInput').value  
  addStudentToDB(student)  
}
```

- ลอง map กับปุ่ม AddButton เพิ่มข้อมูลแล้วลองดูใน postman ว่ามีข้อมูลเข้าไหม?

ถ้าสำเร็จให้แสดงค่าในตารางสิ

- เขียน **code** แสดงค่าในตารางอีกครั้ง

```
function showAllStudents(){
  fetch('https://dv-student-backend-2019.appspot.com/students')
    .then((response) => {
      return response.json()
    }).then(data => {
      addStudentList(data)
    })
}
```

- แล้วปรับ **code** addStudentToDb

```
function addStudentToDB(student) {
  fetch('https://dv-student-backend-2019.appspot.com/students', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(student)
  }).then(response => {
    if (response.status === 200) {
      return response.json()
    } else {
      throw Error(response.statusText)
    }
  }).then(data => {
    console.log('success', data)
    showAllStudents()
  }).catch(error => {
    return null
  })
}
```

ลองทำดู

- ให้เอาข้อมูลที่เพิ่งรับเข้าไปออกมาแสดงผลด้วย

เพิ่มปุ่มลบ

- ให้กดปุ่มลบจาก ตาราง

- แก้ไข html

```
<div id="output" class="px-5">
  <table class="table table-hover">
    <thead>
      <tr>
        <th scope="col">#</th>
        <th scope="col">ชื่อ</th>
        <th scope="col">รหัส</th>
        <th scope="col">ลบ</th>
      </tr>
    </thead>
    <tbody id="tableBody">
    </tbody>
  </table>
</div>
```



- แก้ onload ให้ดึงข้อมูลทั้งหมดขึ้นมาด้วย

เลือกปุ่มกัน

- เอาปุ่มแดงนะคับ

```
function addStudentToTable(index, student) {  
  const tableBody = document.getElementById('tableBody')  
  let row = document.createElement('tr')  
  let cell = document.createElement('th')  
  cell.setAttribute('score', 'row')  
  cell.innerHTML = index  
  row.appendChild(cell)  
  cell = document.createElement('td')  
  cell.innerHTML = `${student.name} ${student.surname}`  
  row.appendChild(cell)  
  cell = document.createElement('td')  
  let img = document.createElement('img')  
  img.setAttribute('src', student.image)  
  // img.classList.add('img-thumbnail')  
  img.height = 200  
  
  cell.appendChild(img)  
  row.appendChild(cell)  
  cell = document.createElement('td')  
  cell.innerHTML = student.gender  
  row.appendChild(cell)  
  cell = document.createElement('td')  
  let button = document.createElement('button')  
  button.classList.add('btn')  
  button.classList.add('btn-danger')  
  button.setAttribute('type', 'button')  
  button.innerText = 'delete'  
  cell.appendChild(button)  
  row.appendChild(cell)  
  row.appendChild(cell)  
  tableBody.appendChild(row)  
}
```

ผลลัพธ์

#	ชื่อ	รหัส	เพศ	ลบ
1	Prayuth The minister		undefined	delete
2	Cherprang BNK48		undefined	delete

Add listener กัน

```
cell.appendChild(img)
row.appendChild(cell)
cell = document.createElement('td')
cell.innerHTML = student.gender
row.appendChild(cell)
cell = document.createElement('td')
let button = document.createElement('button')
button.classList.add('btn')
button.classList.add('btn-danger')
button.setAttribute('type', 'button')
button.innerText = 'delete'
button.addEventListener('click', function () { deleteStudent(student.id) })
cell.appendChild(button)
row.appendChild(cell)
row.appendChild(cell)
tableBody.appendChild(row)
```




- ลองกดปุ่มดู ลองกดปุ่มเดิม สองครั้งแล้วดูผลลัพธ์ใน **console** ว่า **response code** คืออะไร

Error ถูกโยนออกมา

- เพราะไปกดอันเดิมที่ถูกลบไปแล้ว
- ทำอย่างไร
 - ลบแล้วก็ต้องโหลดใหม่สิ
 - เพิ่มข้อมูลก่อนนะครับ

```
button.innerText = 'delete'
button.addEventListener('click', function () {
  deleteStudent(student.id)
  showAllStudents()
})
cell.appendChild(button)
row.appendChild(cell)
```

แล้วผลลัพธ์ละ

#	ชื่อ	รหัส	เพศ	ลบ
1	Prayuth The minister		undefined	delete
2	Cherprang BNK48		undefined	delete
1	Prayuth The minister		undefined	delete

เพราะ?

- Dom เก่าก็ยังอยู่
- ลบ dom เก่าทิ้งก่อน

```
function addStudentList(studentList) {  
  let counter = 1  
  const tableBody = document.getElementById('tableBody')  
  tableBody.innerHTML = ''  
  for (student of studentList) {  
    addStudentToTable(counter++, student)  
  }  
}
```

ปัญหา

- บางที ข้อมูลก็หาย
- บางทีก็ไม่หาย
 - เพราะ ใน deleteStudent มี promise
 - บางทีลบไม่เสร็จแต่ showAllStudents ก็เสร็จแล้ว
 - ต้องเรียก showAllStudents ใน promise
 - เพื่อให้มั่นใจว่าการลบเสร็จแล้วจึงต้องเรียก

```
function deleteStudent(id) {  
  fetch(`https://dv-student-backend-2019.appspot.com/student/${id}`, {  
    method: 'DELETE'  
  }).then(response => {  
    if (response.status === 200) {  
      return response.json()  
    } else {  
      throw Error(response.statusText)  
    }  
  }).then(data => {  
    alert(`student name ${data.name} is now deleted`)  
    showAllStudents()  
  })  
  alert('your input student id is not in the database')  
}
```

```
cell.appendChild(img)  
row.appendChild(cell)  
cell = document.createElement('td')  
cell.innerHTML = student.gender  
row.appendChild(cell)  
cell = document.createElement('td')  
let button = document.createElement('button')  
button.classList.add('btn')  
button.classList.add('btn-danger')  
button.setAttribute('type', 'button')  
button.innerText = 'delete'  
button.addEventListener('click', function () {  
  deleteStudent(student.id)  
  showAllStudents()  
})  
cell.appendChild(button)  
row.appendChild(cell)  
row.appendChild(cell)  
tableBody.appendChild(row)
```

สวยๆ

- แต่เดี๋ยวก่อน
 - เราจะลบโดยไม่ถามไม่ได้นะ
 - ใช้ **alert** เพื่อความมั่นใจ

```
button.setAttribute('type', 'button',  
button.innerText = 'delete'  
button.addEventListener('click', function () {  
    let confirm = confirm(`ท่านต้องการลบคุณ ${student.name} หรือไม่`)  
    if (confirm) {  
        deleteStudent(student.id)  
    }  
})  
cell.appendChild(button)  
row.appendChild(cell)
```

Single page application

- ทุกอย่างรวมกันอยู่หน้าเดียว
- (โหลด **html** แค่ครั้งเดียว)
- ทำอย่างไรให้เลือกการแสดงผลได้

Name



Surname

รหัสนักศึกษา

GPA

Link รูป

Submit

#	ชื่อ	รหัส	เพศ	ลบ
1	Prayuth The minister		undefined	delete
2	Nobi Nobita		undefined	delete

Id:

id

Student id:

id

Name:

name and surname

GPA:

GPA

หน้าจอมีสามส่วน

- 1. ส่วน search
- 2. ส่วนเพิ่มข้อมูล
- 3. ส่วนแสดงรายการทั้งหมด
- ต้องเลือกซ่อนข้อมูลที่เราไม่ต้องการแสดง

ข้อให้หมด

- เพิ่ม **id** ของแต่ละส่วนเพื่อความสะดวก
- ลอง **list** ก่อน
 - **Id = output** -> แสดงส่วนของตารางข้อมูล
 - **Id = single_student_result** -> แสดงข้อมูลนักศึกษาหนึ่งคน
 - ส่วนที่เป็น **form** ยังไม่มี ใส่ **id** ซะ
 - **addUserDetail**

```
<div class="d-flex justify-content-center my-3" id="addUserDetail">
  <form class="row gy-2 gx-3 align-items-center">
    <div class="form-floating">
      <input type="text" class="form-control" id="nameInput" placeholder="ชื่อ">
      <label for="nameInput">Name</label>
    </div>
    <div class="form-floating">
      <input type="text" class="form-control" id="surnameInput" placeholder="นามสกุล">
      <label for="surnameInput">Surname</label>
    </div>
    <div class="form-floating">
      <input type="text" class="form-control" id="studentIdInput" placeholder="รหัส">
      <label for="studentIdInput">รหัสนักศึกษา</label>
    </div>
    <div class="form-floating">
      <input type="text" class="form-control" id="gpaInput" placeholder="gpa">
      <label for="gpaInput">GPA</label>
    </div>
    <div class="form-floating">
```

Function เพื่อซ่อน

- สร้างตัวแปรก่อน

```
var singleStudentResult = document.getElementById('single_tudent_result')
var listStudentResult = document.getElementById('output')
var addUserDetail = document.getElementById('addUserDetail')
```

- แล้วสร้าง function

```
function hideAll(){
    singleStudentResult.style.display='none'
    listStudentResult.style.display='none'
    addUserDetail.style.display='none'
}
```


ปรับ การ load


- โหลดปุ่มซ่อนให้หมด

```
function onLoad() {  
  |   hideAll()  
}
```

ผลลัพธ์

- Box ไม่หาย !!!

[Overview](#) [Inventory](#) [Customers](#) [Products](#)

กลับมาแล้ว 0 

Name

Surname

รหัสนักศึกษา

GPA

Link รูป

```
▶ <head>...</head>
▼ <body onload="onLoad()">
  ▶ <div class="container">...</div>
... ▼ <div class="d-flex justify-content-center my-3" id="addUserDetail" style="display: none;
"> flex == $0
  ▶ <form class="row gy-2 gx-3 align-items-center">...</form> flex
  </div>
  ...
```

ลอง search หาวิธีดู

เก็บไว้ในกล่อง div อีกกล่อง

```
</div>
<div class="d-flex justify-content-center my-3" id="addUserDetail">
  <form class="row gy-2 gx-3 align-items-center">
    <div class="form-floating">
      <input type="text" class="form-control" id="nameInput" placeholder="ชื่อ">
      <label for="nameInput">Name</label>
    </div>
    <div class="form-floating">
      <input type="text" class="form-control" id="surnameInput" placeholder="นามสกุล">
      <label for="surnameInput">Surname</label>
    </div>
    <div class="form-floating">
      <input type="text" class="form-control" id="studentIdInput" placeholder="รหัสนักศึกษา">
      <label for="studentIdInput">รหัสนักศึกษา</label>
    </div>
    <div class="form-floating">
      <input type="text" class="form-control" id="gpaInput" placeholder="gpa">
      <label for="gpaInput">GPA</label>
    </div>
    <div class="form-floating">
      <input type="text" class="form-control" id="imageLinkInput" placeholder="image Link">
      <label for="imageLinkInput">Link รูป</label>
    </div>
    <div class="col-auto">
      <button type="button" class="btn btn-primary" id="addButton">Submit</button>
    </div>
  </form>
</div>
```



```
</div>
<div id="addUserDetail">
  <div class="d-flex justify-content-center my-3">
    <form class="row gy-2 gx-3 align-items-center">
      <div class="form-floating">
        <input type="text" class="form-control" id="nameInput" placeholder="ชื่อ">
        <label for="nameInput">Name</label>
      </div>
      <div class="form-floating">
        <input type="text" class="form-control" id="surnameInput" placeholder="นามสกุล">
        <label for="surnameInput">Surname</label>
      </div>
      <div class="form-floating">
        <input type="text" class="form-control" id="studentIdInput" placeholder="รหัสนักศึกษา">
        <label for="studentIdInput">รหัสนักศึกษา</label>
      </div>
      <div class="form-floating">
        <input type="text" class="form-control" id="gpaInput" placeholder="gpa">
        <label for="gpaInput">GPA</label>
      </div>
      <div class="form-floating">
        <input type="text" class="form-control" id="imageLinkInput" placeholder="image Link">
        <label for="imageLinkInput">Link รูป</label>
      </div>
      <div class="col-auto">
        <button type="button" class="btn btn-primary" id="addButton">Submit</button>
      </div>
    </form>
  </div>
</div>
```

ปรับส่วนเมนูใหม่

```
div class="container">

!\[\]\(8af806fb1314382d09bc5ec5b767526c\_img.jpg\)  
<use xlink:href="#bootstrap"></use>  
</svg>  
</a>  
  
<ul class="nav col-12 col-lg-auto me-lg-auto mb-2 justify-content-center mb-md-0">  
<li><a href="#" class="nav-link px-2 link-secondary">รายชื่อทั้งหมด</a></li>  
<li><a href="#" class="nav-link px-2 link-dark">ค้นหา</a></li>  
<li><a href="#" class="nav-link px-2 link-dark">เพิ่มนักศึกษา</a></li>  
<li><a href="#" class="nav-link px-2 link-dark">Products</a></li>  
</ul>


```

เพิ่ม id

```
<ul class="nav col-12 col-lg-auto me-lg-auto mb-2 justify-content-center mb-md-0">  
  <li><a href="#" class="nav-link px-2 link-secondary" id="allStudentMenu">รายชื่อทั้งหมด</a></li>  
  <li><a href="#" class="nav-link px-2 link-dark" id="searchMenu">ค้นหา</a></li>  
  <li><a href="#" class="nav-link px-2 link-dark" id="addStudentMenu">เพิ่มนักศึกษา</a></li>  
  <li><a href="#" class="nav-link px-2 link-dark">Products</a></li>  
</ul>
```

ถ้าจะ handle ที่ event เดียวก็ใช้ function โดยตรงได้

- เพิ่ม menu รายชื่อทั้งหมด

```
document.getElementById('allStudentMenu').addEventListener('click', (event) =>{  
    listStudentResult.style.display='block'  
    showAllStudents()  
})
```

เพิ่ม การ click สำหรับ เมนู เพิ่มนักศึกษา

```
document.getElementById('addStudentMenu').addEventListener('click',(event) =>{  
    addUserDetail.style.display='block'  
})
```

- ลองกด เมนู รายชื่อทั้งหมด แล้วกด เมนู เพิ่มนักศึกษา

มันจะยังช้อนกัน

- ดังนั้นต้อง **hide** หน้าอื่นก่อนที่จะเพิ่ม

[รายชื่อทั้งหมด](#) [ค้นหา](#) [เพิ่มนักศึกษา](#) [Products](#)

กดมาแล้ว 0

Name



Surname

รหัสนักศึกษา

GPA

Link รูป

Submit

#	ชื่อ	รหัส	เพศ	ลบ
1	Prayuth The minister		undefined	delete
2	Cherprang BNK48		undefined	delete

เพิ่ม hideAll()

```
document.getElementById('allStudentMenu').addEventListener('click', (event) =>{  
  hideAll()  
  listStudentResult.style.display='block'  
  showAllStudents()  
})  
  
document.getElementById('addStudentMenu').addEventListener('click',(event) =>{  
  hideAll()  
  addUserDetail.style.display='block'  
})
```

เพิ่งนึกออกว่าเรามีช่อง search กับปุ่ม search

- ให้ลองเขียนโปรแกรมเพื่อให้พิมพ์คำแล้วกดปุ่ม **search** แล้วแสดงข้อมูลของนักเรียนคนนั้น

ทำปุ่ม edit ในหน้ารวม

- เมื่อกดปุ่ม **edit** แล้วให้แสดงหน้า **edit** ข้อมูล โดยให้มีข้อมูลเดิมของบุคคลนั้นแสดงออกมา
- **Note** การ **edit** คือการใช้ค่า **id** ค่าเดิม แล้วส่งค่าอื่นๆ ที่เป็นค่าที่ปรับปรุงแล้วไปยัง **api** (backend)
 - **Object** ที่ส่งต้องมี **field id** ด้วย