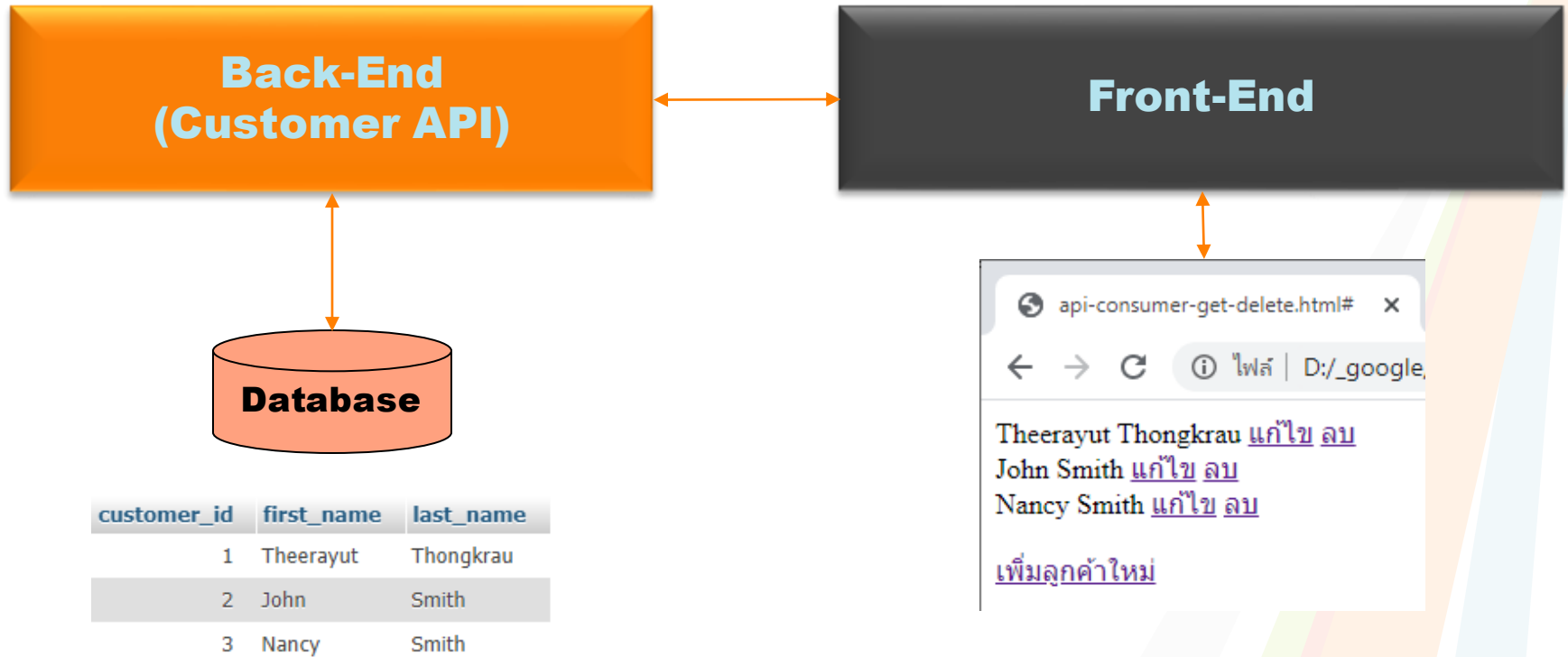




การสร้าง Front-End เรียก API ที่เชื่อมต่อกับฐานข้อมูล

ผศ.ดร.ธีระยุทธ ทองเครือ
หลักสูตรวิศวกรรมซอฟต์แวร์
คณะเทคโนโลยีสารสนเทศและการสื่อสาร
มหาวิทยาลัยพะเยา

ภาพรวมระบบ



➔ Back-End: Customer API

การทำงาน	Request Method	Request URL
แสดงข้อมูลทั้งหมด	GET	http://localhost:8080/customer
แสดงข้อมูลตามคีย์หลัก	GET	http://localhost:8080/customer/{id}
เพิ่มข้อมูลใหม่	POST	http://localhost:8080/customer
ลบข้อมูล	DELETE	http://localhost:8080/customer/{id}
แก้ไขข้อมูล	PUT	http://localhost:8080/customer

➔ การดึงข้อมูลจาก API ผ่าน HTTP GET

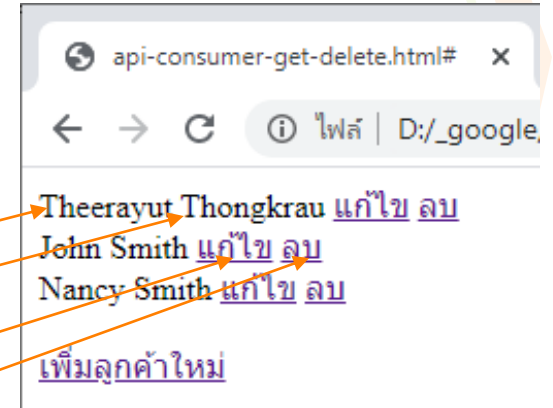
```
<html>
<head>
<script>
```

```
async function loadCustomerList() {
  let response = await fetch('http://localhost:8080/customer');
  let rawData = await response.text();
  let data = JSON.parse(rawData);

  let customerTab = document.getElementById("customerList");

  data.forEach(temp => { // ลูปแสดงรายการลูกค้า พร้อมกับ link แก้ไข และ ลบ
    let div = document.createElement("div"); // สร้างแคว้นใหม่
    customerTab.appendChild(div);
    div.innerHTML = `
      ${temp.firstName} ${temp.lastName}
      <a href="#" onclick="loadEditForm(${temp.customerId})">แก้ไข</a>
      <a href="#" onclick="deleteCustomer(${temp.customerId})">ลบ</a> `;
  });
}
```

เรียก API เพื่อ
ดึงข้อมูล



```
async function deleteCustomer(customerId) {
  if (confirm(`ยืนยันลบลูกค้ารหัส ${customerId} ?`)) {
    let response = await fetch(`http://localhost:8080/customer/${customerId}`, {method: 'DELETE'});
    document.getElementById("customerList").innerHTML = ""; // เคลียร์รายการข้อมูลเดิม
    loadCustomerList(); // โหลดข้อมูลใหม่ เพราะมีข้อมูลถูกลบไป
  }
}
```

```
function loadEditForm(customerId) {
  localStorage.setItem('customerId', customerId); // เก็บรหัสลูกค้าที่ต้องการลบสำหรับใช้ในหน้าถัดไป
  window.location = "api-consumer-put.html"; // โหลดหน้าแบบฟอร์มแก้ไข
}
</script>
</head>
<body onload="loadCustomerList()">
  <div id="customerList"></div><br>
  <div><a href="api-consumer-post.html">เพิ่มลูกค้าใหม่ </a></div>
</body></html>
```

➔ การลบข้อมูลด้วย API ผ่าน HTTP DELETE

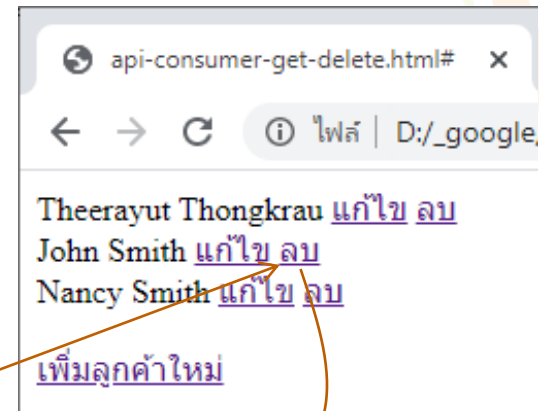
```
<html>
<head>
  <script>
    async function loadCustomerList() {
      let response = await fetch('http://localhost:8080/customer');
      let rawData = await response.text();
      let data = JSON.parse(rawData);

      let customerTab = document.getElementById("customerList");

      data.forEach(temp => { // ลูปแสดงรายการลูกค้า พร้อมกับ link แก้ไข และ ลบ
        let div = document.createElement("div"); // สร้างแถวใหม่
        customerTab.appendChild(div);
        div.innerHTML = `
          ${temp.firstName} ${temp.lastName}
          <a href="#" onclick="loadEditForm(${temp.customerId})">แก้ไข </a>
          <a href="#" onclick="deleteCustomer(${temp.customerId})">ลบ </a> `;
      });
    }

    async function deleteCustomer(customerId) {
      if (confirm(`ยืนยันลบลูกค้ารหัส ${customerId} ?`)) {
        let response = await fetch(`http://localhost:8080/customer/${customerId}`, {method: 'DELETE'});
        document.getElementById("customerList").innerHTML = ""; // เคลียร์รายการข้อมูลเดิม
        loadCustomerList(); // โหลดข้อมูลใหม่ เพราะมีข้อมูลถูกลบไป
      }
    }

    function loadEditForm(customerId) {
      localStorage.setItem('customerId', customerId); // เก็บรหัสลูกค้าที่ต้องการลบสำหรับใช้ในหน้าถัดไป
      window.location = "api-consumer-put.html"; // โหลดหน้าแบบฟอร์มแก้ไข
    }
  </script>
</head>
<body onload="loadCustomerList()">
  <div id="customerList"></div><br>
  <div><a href="api-consumer-post.html">เพิ่มลูกค้าใหม่ </a></div>
</body></html>
```



เรียก API เพื่อ
ลบข้อมูล

➔ การแก้ไขข้อมูลด้วย API ผ่าน HTTP PUT

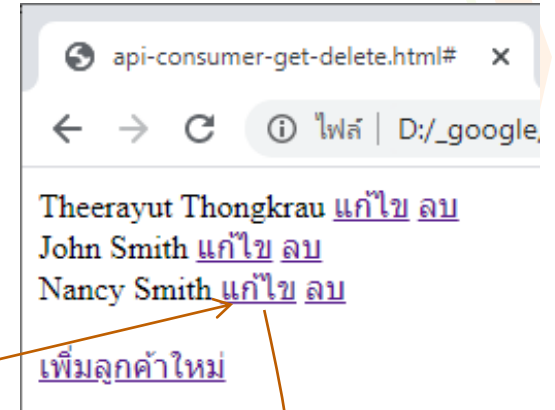
```
<html>
<head>
  <script>
    async function loadCustomerList() {
      let response = await fetch('http://localhost:8080/customer');
      let rawData = await response.text();
      let data = JSON.parse(rawData);

      let customerTab = document.getElementById("customerList");

      data.forEach(temp => { // ลูปแสดงรายการลูกค้า พร้อมกับ link แก้ไข และ ลบ
        let div = document.createElement("div"); // สร้างแถวใหม่
        customerTab.appendChild(div);
        div.innerHTML = ` ${temp.firstName} ${temp.lastName}
          <a href="#" onclick="loadEditForm(${temp.customerId})">แก้ไข</a>
          <a href="#" onclick="deleteCustomer(${temp.customerId})">ลบ</a> `;
      });
    }

    async function deleteCustomer(customerId) {
      if (confirm(`ยืนยันลบลูกค้ารหัส ${customerId} ?`)) {
        let response = await fetch(`http://localhost:8080/customer/${customerId}`, {method: 'DELETE'});
        document.getElementById("customerList").innerHTML = ""; // เคลียร์รายการข้อมูลเดิม
        loadCustomerList(); // โหลดข้อมูลใหม่ เพราะมีข้อมูลถูกลบไป
      }
    }

    function loadEditForm(customerId) {
      localStorage.setItem('customerId', customerId); // เก็บรหัสลูกค้าที่ต้องการลบสำหรับใช้ในหน้าถัดไป
      window.location = "api-consumer-put.html"; // โหลดหน้าแบบฟอร์มแก้ไข
    }
  </script>
</head>
<body onload="loadCustomerList()">
  <div id="customerList"></div><br>
  <div><a href="api-consumer-post.html">เพิ่มลูกค้าใหม่ </a></div>
</body></html>
```



➔ การสร้างแบบฟอร์มแก้ไขข้อมูล

```
<html>
<head>
<script>
  async function loadCustomer() { // ดึงข้อมูลลูกค้า 1 คน
    let customerId = localStorage.getItem('customerId'); // เอารหัสที่เลือกว่าจะลบออกมา
    let response = await fetch(`http://localhost:8080/customer/${customerId}`); // เรียก API ขอข้อมูลลูกค้าตามรหัส
    let rawData = await response.text();
    let customerObject = JSON.parse(rawData);
    document.getElementById("firstName").value = customerObject.firstName; // แสดงข้อมูลในช่องแบบฟอร์ม
    document.getElementById("lastName").value = customerObject.lastName;
  }
}
```

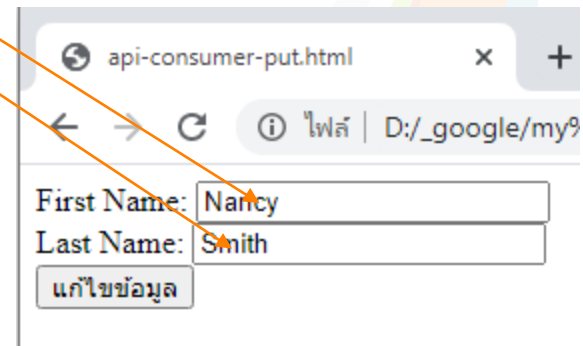
```
async function editCustomer() {
  const customerData = new Object();
  customerData.customerId = localStorage.getItem('customerId'); // ดึงข้อมูลใหม่เก็บลง object
  customerData.firstName = document.getElementById("firstName").value;
  customerData.lastName = document.getElementById("lastName").value;
```

```
let response = await fetch('http://localhost:8080/customer', {
  method: 'PUT',
  headers: {"Content-type": "application/json;charset=UTF-8"},
  body: JSON.stringify(customerData) // ส่งข้อมูลในรูปแบบ JSON ไปให้ API แก้ไขในฐานข้อมูล
});
```

```
window.location = "api-consumer-get-delete.html"; // ย้อนกลับไปหน้ารายการลูกค้า
}
```

```
</script>
</head>
```

```
<body onload="loadCustomer()">
  First Name: <input id="firstName" type="text"><br>
  Last Name: <input id="lastName" type="text"><br>
  <button onclick="editCustomer()">แก้ไขข้อมูล</button>
</body>
</html>
```



เรียก API เพื่อ
แก้ไขข้อมูล

➔ การสร้างแบบฟอร์มเพิ่มข้อมูล

```
<html>
<head>
  <script>

    async function createCustomer() {

      const customerData = new Object();
      customerData.firstName = document.getElementById("firstName").value; // ดึงค่าจากฟอร์มเก็บลง object
      customerData.lastName = document.getElementById("lastName").value;

      let response = await fetch('http://localhost:8080/customer', { ← เรียก API เพื่อเพิ่มข้อมูล
        method: 'POST',
        headers: {"Content-type": "application/json;charset=UTF-8"},
        body: JSON.stringify(customerData) // ส่งข้อมูลในรูปแบบ JSON ไปให้ API เพิ่มในฐานข้อมูล
      });

      window.location = "api-consumer-get-delete.html"; // ย้อนกลับไปหน้ารายการลูกค้า
    }

  </script>
</head>
<body>
  First Name: <input id="firstName" type="text"> <br>
  Last Name: <input id="lastName" type="text"> <br>
  <button onclick="createCustomer()">เพิ่มข้อมูล </button>
</body>
</html>
```



api-consumer-post.html

ไฟล์ | D:/_google/my...

First Name:

Last Name:

เพิ่มข้อมูล



คลาสมาตรฐาน LocalDateTime

```
import java.time.*;  
import java.time.format.DateTimeFormatter;  
import java.time.temporal.ChronoUnit;
```

```
public class DateTimeDemo {
```

```
    public static void main(String[] args) {
```

```
        // ขอรวันและเวลาปัจจุบัน
```

```
        System.out.println(LocalDate.now());  
        System.out.println(LocalDate.now());  
        System.out.println(LocalTime.now());  
        System.out.println(LocalDate.now().getYear());
```

```
        // หาความต่างของวัน
```

```
        LocalDateTime dateTime1 = LocalDateTime.now();  
        LocalDateTime dateTime2 = LocalDateTime.of(2020, 1, 8, 17, 30);  
        System.out.println(ChronoUnit.DAYS.between(dateTime1, dateTime2));
```

```
        // จัดรูปแบบวันเวลา
```

```
        System.out.println(dateTime1.format(DateTimeFormatter.BASIC_ISO_DATE));  
        System.out.println(dateTime1.format(DateTimeFormatter.ofPattern("dd MMMM yyyy")));
```

```
    }
```

```
}
```

ตัวอย่างผลลัพธ์

```
2020-02-08T17:37:10.839  
2020-02-08  
17:37:10.840  
2020  
-31  
20200208  
08 February 2020
```

➔ Assignment#7

- จงสร้าง Front-End เพื่อเรียกใช้ Back-End ที่ได้สร้างไว้แล้วใน Assignment#6 โดยให้ออกแบบ UI ใหม่ที่ไม่เหมือนตัวอย่างด้านล่างนี้

