



TodoList Application





TodoList Management

Todo class

```
static runningId = 1
//properties
-id //unique id
-description //todo detail

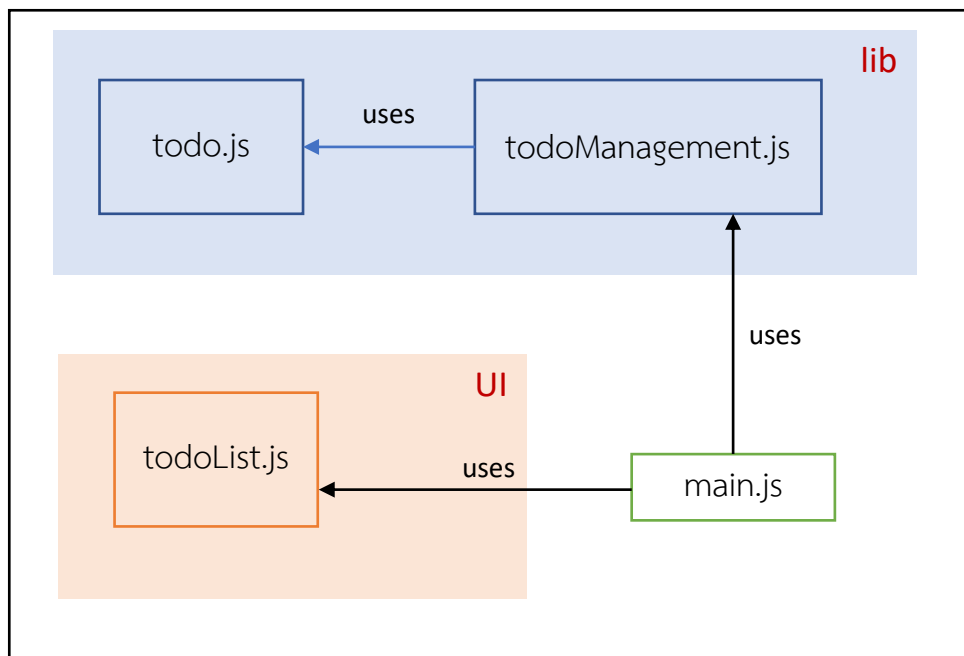
//methods
//returns {id, description} object
-getTodo()
//allows to edit description
-setDescription(newDescription)
```

TodoList Management function

```
- todos //stores all todo in array
//adds a new todo to the end of todos array, returns
the new length of the array
- function addTodo(desc)
//returns a todo object that its id equals to searchId
- function findTodo(searchId)
//returns an index of todo that its id equals to
searchId
- function findIndexTodo(searchId)
//removes a todo that its id equals to removeId
- function removeTodo(removeId)
//returns todos array
- function getTodos()
```

Document Object Modeling

(DOM) TodoList Project Diagram



1. ไฟล์ `./lib/todo.js` เพื่อจัดการกับ property และฟังก์ชันของแต่ละ todo
2. ไฟล์ `./lib/todoManagement.js` เพื่อจัดการโครงสร้างข้อมูลและฟังก์ชันของ todo ทั้งหมด
3. ไฟล์ `./UI/todoListUI.js` เพื่อจัดการ DOM ของเอกสาร HTML
4. ไฟล์ `./main.js` สำหรับทดสอบไฟล์ต่าง ๆ ที่เกี่ยวข้อง

```

<html lang="en">

<head>
  <title>Todo List Application</title>
  <style>
    #listTodo {
      border-style: solid;
      border-width: 1pt;
      border-color: gray;
      padding: 10pt;
    }
    .todoItem {
      display: flex;
      align-items: center;
    }
    button {
      margin-right: 5pt;
      height: 25pt;
    }
    p {
      margin-right: 8pt;
    }
    input {
      height: 20pt;
    }
  </style>
</head>

```

```

<body>
  <div>
    <div id="addTodo">
      <h2>Add Your Todo:</h2>
      <input placeholder="Add Todo" />
      <button id="addBtn">Add</button>
    </div>
    <h1>### Your Todo list ###</h1>
    <div id="listTodo"></div>
    <div id="summaryTodo">
      <p id="done">Number of Done:</p>
      <p id="notDone">Number of Not Done:</p>
    </div>
  </div>
  <script src="./main.js" type="module"></script>
</body>
</html>

```

ปรับปรุงไฟล์ ./lib/todo.js เพื่อจัดการกับ property และฟังก์ชันของแต่ละ todo

1. ให้เพิ่ม property done (Boolean) และปรับ constructor ให้รับ id, description, done เป็นพารามิเตอร์ เพื่อสร้าง todo object โดยถ้าค่า id เป็น undefined ให้ค่า id มีค่าเป็น running Id สำหรับ done ให้มีค่า default เป็น false
2. ให้เพิ่ม property done และเพิ่ม setDone method เพื่อกำหนดค่าให้กับ done

ปรับปรุงไฟล์ ./lib/todoManagement.js เพื่อจัดการโครงสร้างข้อมูลและฟังก์ชันของ todo ทั้งหมด

1. ให้แก้ไขฟังก์ชัน addTodo(desc) ให้ return todo.id ของรายการ todo ที่เพิ่มแทนการ return length ของ array
2. ให้เพิ่มฟังก์ชัน getNumberOfDone เพื่อ return จำนวนของ Todo ที่อยู่ในสถานะ Done
3. ให้เพิ่มฟังก์ชัน getNumberOfNotDone เพื่อ return จำนวนของ Todo ที่อยู่ในสถานะ Not Done
4. ให้เพิ่มฟังก์ชัน clearTodo เพื่อ reset ค่า todo ให้เป็น empty array

สร้างไฟล์ ./UI/todoListUI.js เพื่อจัดการ dom ของเอกสาร HTML

1. ให้เพิ่มฟังก์ชันที่ชื่อ showTodoItem(newId, newDescription) โดยทำการรับค่า todo id และ description เป็นพารามิเตอร์ เพื่อทำการแสดงรายการ todo ภายใต้ <div id="listTodo"></div> ของเอกสาร HTML โดยแต่ละรายการ todo ให้มีโครงสร้าง tags เรียงลำดับ ดังนี้

```
<div class="todoItem" id="newId">
  <p>newDescription</p>
  <button>Not done</button>
  <button>remove</button>
</div>
```

2. ให้เพิ่มฟังก์ชัน showNumberOfDone(numberOfDone) เพื่อแสดงเลขจำนวนของ Todo ที่อยู่ในสถานะ Done ต่อท้ายข้อความภายใน <p id="done">Number of Done:</p>
3. ให้เพิ่มฟังก์ชัน showNumberOfNotDone(numberOfNotDone) เพื่อแสดงเลขจำนวนของ Todo ที่อยู่ในสถานะ Not Done ต่อท้ายข้อความภายใน <p id="notDone">Number of Not Done:</p>

Add Your Todo:

Your Todo list

Watch Movies

Visit Grandmother

Coding

Number of Done:0

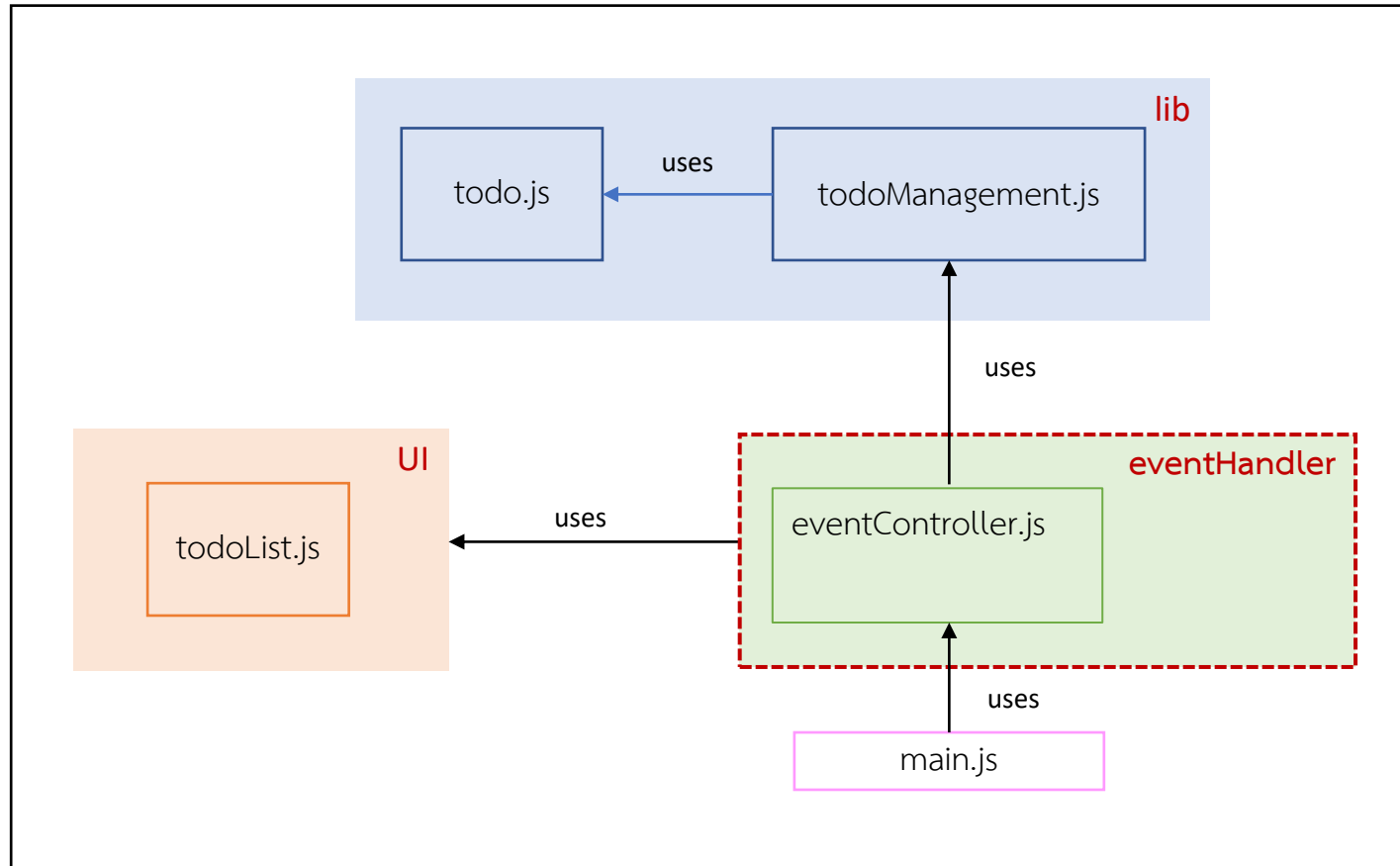
Number of Not Done:3

สร้างไฟล์ ./main.js

1. ให้ทดสอบเรียกใช้ addTodo() และรับ id ที่ return จากฟังก์ชัน เพื่อส่งต่อให้ฟังก์ชัน showTodoItem() เพื่อทำการสร้าง tags แสดงผลใน HTML File
2. ให้ทดสอบเรียกใช้ getNumberOfDone() และ getNumberOfNotDone() และรับค่าจำนวนของ Todo ที่อยู่ในสถานะ Done และ NotDone เพื่อส่งต่อฟังก์ชัน showNumberOfDone() และ showNumberOfNotDone() เพื่อแสดงผลใน HTML File

Event Handling

(Event Handlers) TodoList Project Diagram



1. ไฟล์ `./lib/todo.js` เพื่อจัดการกับ property และ ฟังก์ชันของแต่ละ todo
2. ไฟล์ `./lib/todoManagement.js` เพื่อจัดการ โครงสร้างข้อมูลและฟังก์ชันของ todo ทั้งหมด
3. ไฟล์ `./UI/todoListUI.js` เพื่อจัดการ DOM ของ เอกสาร HTML
4. ไฟล์ `./eventHandler/eventController.js` เพื่อ จัดการกับ event ที่เกิดขึ้นบน DOM ของเอกสาร HTML
5. ไฟล์ `./main.js` เริ่มต้นทำงาน

ปรับปรุงไฟล์ todoManagement.js เพื่อจัดการโครงสร้างข้อมูลและฟังก์ชันของ todo ทั้งหมด

1. ให้เพิ่มฟังก์ชัน setItemToDone(doneld) เพื่อค้นหา todo ที่มีค่าตรงกับ doneld จากนั้นให้เรียกฟังก์ชัน setDone() เพื่อกำหนดค่า done ให้เป็น true

ปรับปรุงไฟล์ todoList.js เพื่อจัดการ DOM ของเอกสาร HTML

1. ให้เพิ่มฟังก์ชัน removeTodoItem(removeld) เพื่อลบ <div> ของรายการ Todo นั้นออกไปจาก HTML

สร้างไฟล์ eventController.js เพื่อสร้าง event handle functions ดังนี้

1. ให้สร้างฟังก์ชัน addTodoHandler() เพื่อจัดการเมื่อผู้ใช้กดปุ่ม add ให้ตรวจสอบว่าไม่เป็นค่า empty string จึงเพิ่มรายการ todo นั้นที่ todos array และให้แสดง todo ภายใต้ <div id="listTodo"></div> ของเอกสาร HTML จากนั้นค้นหาปุ่ม Not Done และ Remove ของ todo ที่ add เพื่อลงทะเบียน event handler function notDoneButtonHandler() และ removeButtonHandler() ให้กับปุ่ม Not Done และ Remove
2. ให้สร้างฟังก์ชัน notDoneButtonHandler() เพื่อจัดการเมื่อผู้ใช้กดปุ่ม Not Done ข้อความจะเปลี่ยนเป็น done โดยแสดง background เป็นสีเขียว และตัวอักษรเป็นสีขาว จากนั้นให้ทำการเรียกใช้ฟังก์ชัน setItemToDone (ไฟล์ todoManagement.js) โดยทำการส่ง todo id ของรายการ todo นั้น เพื่อให้เปลี่ยนค่า property done เป็น true
3. ให้สร้างฟังก์ชัน removeButtonHandler() เพื่อจัดการเมื่อผู้ใช้กดปุ่ม remove จะทำการเรียกใช้ฟังก์ชัน removeTodoItem(removeld) (ไฟล์ todoList.js) และฟังก์ชัน removeTodo(removeld) (ไฟล์ todoManagement.js) เพื่อลบรายการ Todo นั้นออกจาก DOM HTML และจาก todos array

ท้ายฟังก์ชันในข้อ 1 - ข้อ 3 ให้ update จำนวน todo ที่มีสถานะ Done และ Not Done โดยการเรียกใช้ฟังก์ชัน showNumberOfDone() และ showNumberOfNotDone()

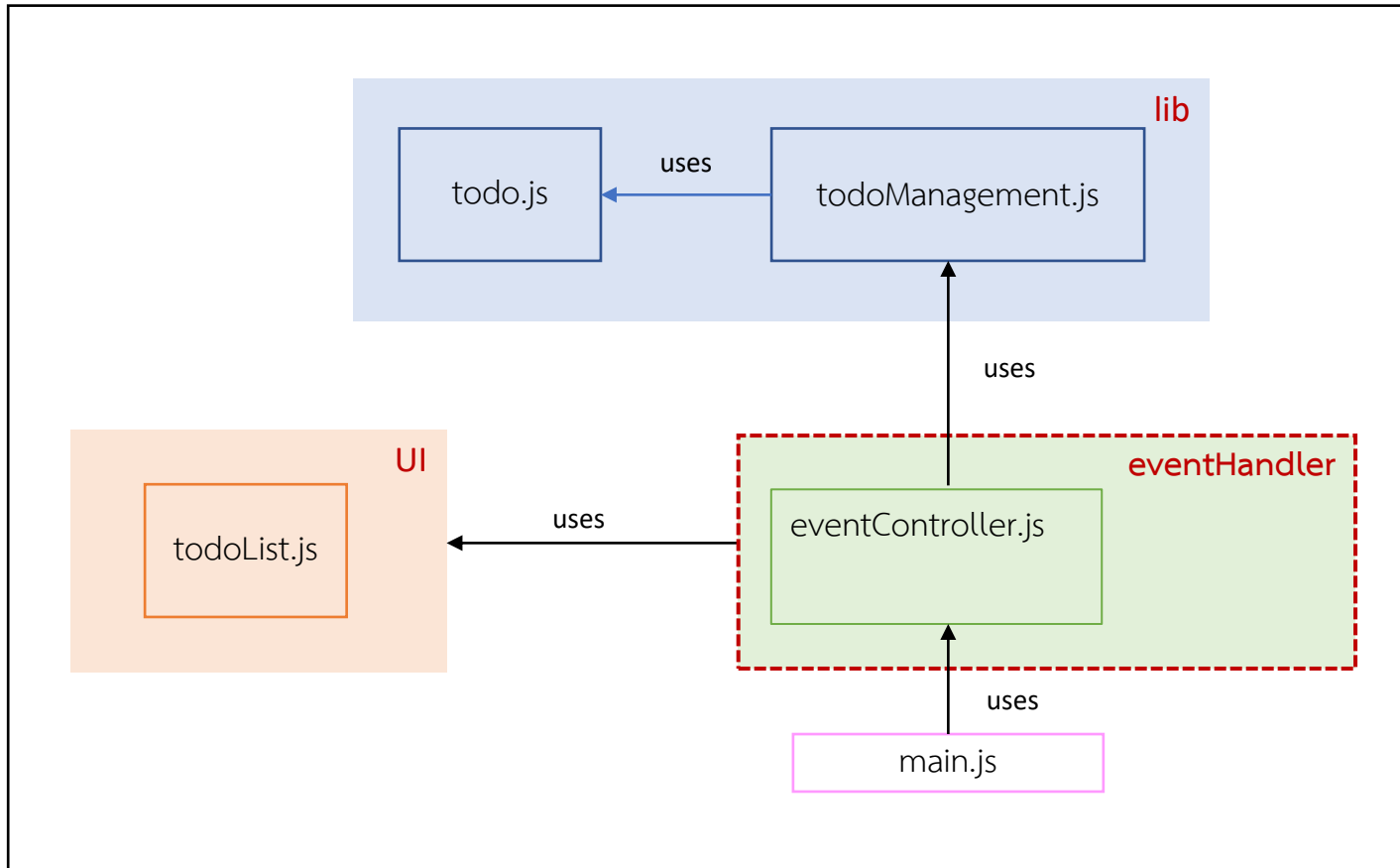
แทนที่ code ไฟล์ main.js ด้วยคำสั่ง ดังนี้

1. ให้ทำการค้นหาปุ่ม Add Todo และลงทะเบียน addTodoHandler() ให้กับปุ่ม Add Todo



Client-Side Storages

(Client Storages) TodoList Project Diagram



1. ไฟล์ `./lib/todo.js` เพื่อจัดการกับ property และ ฟังก์ชันของแต่ละ todo
2. ไฟล์ `./lib/todoManagement.js` เพื่อจัดการ โครงสร้างข้อมูลและฟังก์ชันของ todo ทั้งหมด
3. ไฟล์ `./UI/todoListUI.js` เพื่อจัดการ DOM ของ เอกสาร HTML
4. ไฟล์ `./eventHandler/eventController.js` เพื่อ จัดการกับ event ที่เกิดขึ้นบน DOM ของเอกสาร HTML และบันทึกข้อมูลของผู้ใช้ไว้ใน client-side storage
5. ไฟล์ `./main.js` เริ่มต้นทำงาน

ปรับปรุงไฟล์ todoManagement.js เพื่อจัดการโครงสร้างข้อมูลและฟังก์ชันของ todo ทั้งหมด

- 1. ให้เพิ่มฟังก์ชัน loadTodos (userTodos) เพื่อนำค่า todo ที่เก็บใน local storage เป็นค่าตั้งต้นให้กับ todos array

ปรับปรุงไฟล์ eventController.js เพื่อสร้าง event handle functions ดังนี้

- 1. ให้เพิ่มฟังก์ชัน loadHandler() เพื่อทำการ get ค่า todos จาก local storage และให้แปลง string เป็น array โดยใช้ JSON.parse โดยตรวจสอบว่าไม่เป็น empty array หรือ null หรือ undefined จากนั้นส่งไปเป็นพารามิเตอร์ของฟังก์ชัน loadTodos() จากนั้นให้เรียกฟังก์ชัน getTodos() เพื่อจะ show แต่ละรายการ todos ในเอกสาร HTML โดยการเรียกฟังก์ชัน showTodoItem() และให้ add handler ของปุ่ม Not Done และปุ่ม remove กรณี todo รายการนั้น done แล้ว ให้ใส่ style background เป็นสีเขียว และตัวอักษรเป็นสีขาว จากนั้น update จำนวนรายการ done และ not done โดยเรียกฟังก์ชัน showNumberOfDone() และ showNumberOfNotDone() ตามลำดับ และให้ทำการค้นหาปุ่ม Add Todo และลงทะเบียน addTodoHandler() ให้กับปุ่ม Add Todo
- 2. ให้เพิ่มฟังก์ชัน beforeUnloadHandler(event) โดยให้เรียกฟังก์ชัน preventDefault() เพื่อจะได้สามารถบันทึกค่า todos ปัจจุบัน โดยตั้งชื่อ todos เพิ่มเข้าไปใน local storage ได้ก่อนจะมีการ unload HTML Page และทำการเคลียร์ค่า todo array ให้เป็น empty

แทนที่ code ไฟล์ main.js ด้วยคำสั่ง ดังนี้

- 1. เพิ่ม event เมื่อ window ถูกโหลด โดยเรียกใช้ loadHandler() เพื่อโหลดค่า todos ของ user ที่บันทึกไว้ใน local storage
- 2. เพิ่ม event ก่อน window จะ unload โดยเรียกใช้ beforeUnloadHandler(event) เพื่อบันทึกค่า todos ของ user ใน local storage

Add Your Todo:

Add Todo

Add

When window loading

Your Todo list

Grading Exam

Done

remove

Online Learning

Done

remove

Internship Meeting

Not Done

remove

Project Consulting

Done

remove

Number of Done:3

Number of Not Done:1

Application

- Manifest
- Service Workers
- Storage

Storage

- Local Storage
 - http://127.0.0.1:5501
 - Session Storage
 - IndexedDB
 - Web SQL
 - Cookies

Key	Value
todos	[[{"id":1,"description":"Grading Exam","done":true}, {"id":2,"description":"Online Learning","don...

When window beforeUnloading

```
▼ [{"id": 1, "description": "Grading Exam", "done": true}, {"id": 2, "description": "Online Learning", "done": true}, ...]  
  ▶ 0: {"id": 1, "description": "Grading Exam", "done": true}  
  ▶ 1: {"id": 2, "description": "Online Learning", "done": true}  
  ▶ 2: {"id": 3, "description": "Internship Meeting", "done": false}  
  ▶ 3: {"id": 4, "description": "Project Consulting", "done": true}
```