**บทที่ 12 การใช้ React กับ REST API**

**ติดตั้ง Web Server และเตรียม Endpoint**

const express = require('express');

const bodyParser = require('body-parser');

const app = express();

const uuid = require('uuid');

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({extended: false}));

app.use((req, res, next) => {

res.setHeader("Access-Control-Allow-Origin", "\*");

res.setHeader(

"Access-Control-Allow-Headers",

"Origin, X-Requested-With, Content-Type, Accept, Authorization"

);

res.setHeader(

"Access-Control-Allow-Methods",

"GET, POST, PUT, DELETE"

)

next();

})

app.use(express.json());

let students = [

{'id':'1', 'name': 'Jeerawuth', 'email':'jeerawuth@hotmail.com'},

{'id':'2', 'name': 'Worrawan', 'email':'worrawan99@gmail.com'},

{'id':'3', 'name': 'Sombat', 'email':'sombatlovelove@me.com'},

{'id':'4', 'name': 'Suwichan', 'email':'suwichang99@hotmail.com'},

{'id':'5', 'name': 'Romyen', 'email':'romyen@gmail.com'},

{'id':'6', 'name': 'Natipong', 'email':'natipong@hotmail.com'}

];

app.get('/api/students', function(req, res){

if(students.length > 0) {

res.send(students);

} else {

res.status(400).send('Not found any student');

}

});

app.get('/api/students/:id', function(req, res){

const id = req.params.id;

const student = students.find(item => item.id === id);

if(student) {

res.send(student);

} else {

res.status(400).send(`Not found student for id ${id}`);

}

});

app.post('/api/students/', function(req, res){

const studentName = req.body.name;

const studentEmail = req.body.email;

if (studentName.length <= 0) {

res.status(400).send('Error cannot add student!');

} else {

const student = {

'id': uuid(),

'name': studentName,

'email': studentEmail

}

students.push(student);

res.send(student);

}

});

app.delete('/api/students/:id', function(req, res){

const id = req.params.id;

const student = students.find(item => item.id === id);

if (student) {

const index = students.indexOf(student);

students.splice(index, 1);

res.send(student);

} else {

res.status(400).send('Error cannot delete student!');

}

});

app.put('/api/students/:id', function(req, res){

const id = req.params.id;

const studentName = req.body.name;

const studentEmail = req.body.email;

if (studentName.length < 1) {

res.status(400).send('Error cannot update student!');

} else {

let student = students.find(item => item.id === id);

if (student) {

student.name = studentName;

student.email = studentEmail;

res.send(student);

} else {

res.status(400).send('Cannot find student to update');

}

}

});

const port = process.env.port || 3001;

app.listen(port, function(){

console.log('Listening on port', port);

});

**ใช้ thunk เป็น middleware**

import React from 'react';

import { createStore, applyMiddleware } from 'redux';

import reducer from './reducer';

import { Provider } from 'react-redux';

import thunk from 'redux-thunk';

function App() {

const store = createStore(reducer, applyMiddleware(thunk));

return (

<Provider store={store}>

<div className="container">

</div>

</Provider>

);

}

export default App;

**action creator กับ thunk**

**กรณีที่ 1 action creator สำหรับใช้งานทั่วไป**

const add = (data) => {

return {

type: 'ADD\_STUDENT',

playload: data

}

}

**กรณีที่ 2 action creator สำหรับใช้งานกับ thunk**

const addAsync = (data) => {

return (dispatch) => {

setTimeout(() => {

dispatch({

type: 'ADD\_STUDENT',

playload: data

});

}, 2000);

}

}

**ใช้งาน HTTP Request อย่างง่ายๆ ด้วย axios**

import axios from 'axios';

axios.get('http://localhost:3001/api/students/')

.then((response)=>{

console.log(response);

})

.catch((err)=>{

console.log(err);

});

**ไฟล์ App.js**

import React from 'react';

import { BrowserRouter, Route, Switch } from 'react-router-dom';

import 'bootstrap/dist/css/bootstrap.min.css';

import { createStore, applyMiddleware } from 'redux';

import reducer from './reducer';

import { Provider } from 'react-redux';

import StudentLists from './Student-lists';

import AddStudent from './Add-student';

import EditStudent from './Edit-student';

import Header from './Header';

import PageNotFound from './Page-not-found';

import thunk from 'redux-thunk';

function App() {

const store = createStore(reducer, applyMiddleware(thunk));

return (

<Provider store={store}>

<div className="container">

<BrowserRouter>

<Header />

<Switch>

<Route exact path="/edit/:id" component={EditStudent} />

<Route exact path="/add" component={AddStudent} />

<Route exact path="/" component={StudentLists} />

<Route component={PageNotFound} />

</Switch>

</BrowserRouter>

</div>

</Provider>

);

}

export default App;

**ไฟล์ reducer.js**

const initialState = {

students: []

}

const reducer = (state = initialState, action) => {

const allStudents = [...state.students];

switch(action.type) {

case 'GET\_STUDENT\_LISTS':

const allStudentsState = {

...state,

students: action.playload

}

return allStudentsState;

case 'DEL\_STUDENT':

const newState = {

...state,

students: state.students.filter(item => item.id !== action.playload)

}

return newState;

case 'ADD\_STUDENT':

const addedState = {

...state,

students: [action.playload, ...state.students]

}

return addedState;

case 'EDIT\_STUDENT':

const indexForEdit = allStudents.findIndex((item) => {

return item.id === action.playload.id;

});

console.log('index for editing',indexForEdit);

allStudents[indexForEdit] = {

id: action.playload.id,

name: action.playload.name,

email: action.playload.email

}

const editedState = {

...state,

students: allStudents

}

return editedState;

default:

break;

}

return state;

}

export default reducer;

**ไฟล์ action.js**

import axios from 'axios';

export const delStudent = (id) => {

return (dispatch) => {

axios.delete(`http://localhost:3001/api/students/${id}`)

.then((res) => {

dispatch({

type: 'DEL\_STUDENT',

playload: res.data.id

});

});

}

}

export const addStudent = (data) => {

return (dispatch) => {

axios.post('http://localhost:3001/api/students/', data)

.then((res)=>{

dispatch({

type: 'ADD\_STUDENT',

playload: res.data

});

});

}

}

export const editStudent = (data) => {

return (dispatch) => {

axios.put(`http://localhost:3001/api/students/${data.id}`, data)

.then((res)=>{

dispatch({

type: 'EDIT\_STUDENT',

playload: res.data

});

});

}

}

export const getStudent = (id) => {

return (dispatch) => {

axios.get(`https://localhost:3001/api/students/${id}`)

.then(()=>{

dispatch({

type: 'GET\_STUDENT',

playload: id

});

});

}

}

export const getStudentLists = () => {

return (dispatch) => {

axios.get('http://localhost:3001/api/students/')

.then((res)=>{

dispatch({

type: 'GET\_STUDENT\_LISTS',

playload: res.data

});

});

}

}

**ไฟล์ Header.js**

import React, { Component } from 'react'

import { Link } from 'react-router-dom';

class Header extends Component {

state = {

collapse: false

}

render() {

return (

<nav className="navbar navbar-expand-lg navbar-light bg-light mx-auto text-center">

<Link className="navbar-brand" to="/">Home</Link>

<button className="navbar-toggler"

type="button" data-toggle="collapse"

data-target="#navbarNav" aria-controls="navbarNav"

aria-expanded="true" aria-label="Toggle navigation">

<span className="navbar-toggler-icon"

onClick={() => {

this.setState({ collapse: !this.state.collapse} );

}}

></span>

</button>

<div className={

this.state.collapse?

'collapse navbar-collapse show text-center':

'collapse navbar-collapse'

}>

<ul className="navbar-nav">

<li className="nav-item active">

<Link className="nav-link" to="/">All Students</Link>

</li>

<li className="nav-item">

<Link className="nav-link" to="/add">Add Student</Link>

</li>

</ul>

</div>

</nav>

)

}

}

export default Header;

**ไฟล์ Student.js**

import React, { Component } from 'react';

import { connect } from 'react-redux';

import { Link } from 'react-router-dom';

import \* as action from './action';

class Student extends Component {

render() {

const delFn = this.props.deleteStudentAtStore;

return (

<div className="card">

<div className="card-header">

name: {this.props.data.name}

<button

className="btn btn-outline-danger btn-sm float-right ml-1"

onClick={delFn.bind(this, this.props.data.id)}

>ลบ</button>

<Link to={`/edit/${this.props.data.id}`}>

<button

className="btn btn-success btn-sm float-right"

>แก้ไข</button>

</Link>

</div>

<div className="card-body">

email: {this.props.data.email}

</div>

</div>

)

}

}

const mapDispatchToProps = dispatch => {

return {

deleteStudentAtStore : (id) => {

return dispatch(action.delStudent(id));

}

}

}

export default connect(null, mapDispatchToProps)(Student);

**ไฟล์ Student-lists.js**

import React, { Component } from 'react';

import Student from './Student';

import { connect } from 'react-redux';

import \* as action from './action';

class StudentLists extends Component {

render() {

const allStudents = this.props.studentsFromStore;

let lists = (

<div className="col-12 mx-auto">

<div className="alert-info text-center pt-5 pb-5">ไม่พบข้อมูลนักเรียน</div>

</div>

);

if(allStudents.length !== 0) {

lists = allStudents.map(item =>

(

<div className="col-12 col-sm-6 col-lg-4 mt-3" key={item.id}>

<Student data={item} />

</div>

)

);

}

return (

<div className="row">

{lists}

</div>

)

}

componentDidMount() {

this.props.getAllStudents();

}

}

const mapStateToProps = state => {

return {

studentsFromStore : state.students

}

}

const mapDispatchToProps = dispatch => {

return {

getAllStudents : () => {

return dispatch(action.getStudentLists());

}

}

}

export default connect(mapStateToProps, mapDispatchToProps)(StudentLists);

**ไฟล์ Add-student.js**

import React, { Component } from 'react';

import { connect } from 'react-redux';

import \* as action from './action';

class AddStudent extends Component {

state = {

name: '',

email: ''

}

onChangeData = (event) => {

this.setState(

{

[event.target.name] : event.target.value

}

)

}

onSubmitStudentForm = (data, event) => {

event.preventDefault();

const newData = {

name: data.name,

email: data.email

}

this.props.addStudentAtStore(newData);

this.props.history.push('/');

}

render() {

return (

<div className="row mt-3 mb-5 ml-5 mr-5">

<div className="card col-12 col-sm-8 col-lg-6 mx-auto">

<div className="card-body">

<form onSubmit={this.onSubmitStudentForm.bind(this, this.state)}>

<div className="block-4 text-center">

<img

className="img img-thumbnail mt-5 mb-3"

src="http://lorempixel.com/150/150/sports/" alt="โลโก้"

/>

</div>

<div className="form-group">

<label>Name</label>

<input

type="text"

className="form-control"

name="name"

value={this.state.name}

onChange={this.onChangeData} />

</div>

<div className="form-group">

<label>Email</label>

<input

type="email"

className="form-control"

name="email"

value={this.state.email}

onChange={this.onChangeData} />

</div>

<div className="form-group text-center pt-3">

<button

onClick={() => this.props.history.push('/')}

className="btn btn-outline-success btn-sm mr-1">ยกเลิก</button>

<button

type="submit"

className="btn btn-sm btn-primary ml-1">เพิ่ม</button>

</div>

</form>

</div>

</div>

</div>

)

}

}

const mapDispatchToProps = dispatch => {

return {

addStudentAtStore : (newStudentData) => {

return dispatch(action.addStudent(newStudentData));

}

}

}

export default connect(null, mapDispatchToProps)(AddStudent);

**ไฟล์ Edit-student.js**

import React, { Component } from 'react';

import { connect } from 'react-redux';

import \* as action from './action';

class EditStudent extends Component {

state = {

id: '',

name: '',

email: ''

}

onChangeData = (event) => {

this.setState(

{

[event.target.name]: event.target.value

}

)

}

componentDidMount() {

const userId = this.props.match.params.id;

const student = this.getCurrentStudent(userId);

if(student) {

this.setState({

id: student.id,

name: student.name,

email: student.email

})

} else {

this.props.history.push('/');

}

}

getCurrentStudent(id) {

const allStudents = this.props.studentsFromStore;

const [student] = allStudents.filter((item) => {

return item.id.toString() === id.toString();

});

return student;

}

onSubmitStudentForm = (data, event) => {

event.preventDefault();

const newData = {

id: data.id,

name: data.name,

email: data.email

}

this.props.editStudentAtStore(newData);

this.props.history.push('/');

}

render() {

return (

<div className="row mt-3 mb-5 ml-5 mr-5">

<div className="card col-12 col-sm-8 col-lg-6 mx-auto">

<div className="card-body">

<form onSubmit={this.onSubmitStudentForm.bind(this, this.state)}>

<div className="form-group">

<label>Name</label>

<input

type="text"

className="form-control"

name="name"

value={this.state.name}

onChange={this.onChangeData} />

</div>

<div className="form-group">

<label>Email</label>

<input

type="email"

className="form-control"

name="email"

value={this.state.email}

onChange={this.onChangeData} />

</div>

<div className="form-group text-center pt-3">

<button

onClick={() => this.props.history.push('/')}

className="btn btn-outline-success mr-1">ยกเลิก</button>

<button

type="submit"

className="btn btn-primary ml-1">แก้ไข</button>

</div>

</form>

</div>

</div>

</div>

)

}

}

const mapStateToProps = state => {

return {

studentsFromStore : state.students

}

}

const mapDispatchToProps = dispatch => {

return {

editStudentAtStore : (data) => {

return dispatch(action.editStudent(data))

}

}

}

export default connect(mapStateToProps, mapDispatchToProps)(EditStudent);

**ไฟล์ Page-not-found.js**

import React, { Component } from 'react'

class PageNotFound extends Component {

render() {

return (

<div className="row mt-5 mb-5">

<div className="col-10 text-center p-5 mx-auto">

<h1 className="display-4 mb-4 pb-3">

<span className="text-danger">ERROR 404</span> Page Not Found!!

</h1>

<p className="lead">ขออภัยในความไม่สะดวก</p>

</div>

</div>

)

}

}

export default PageNotFound;