

นาย ศุภวิชญ์ แก่นนาค

## lab5 engse202

ຝັ່ງ backend api

1.check สถานะ pm2 ว่า api ยังทำงานอยู่หรือไม่

```
supawit@backend2-66543210031:~$ pm2 status
```

| id | name        | mode | ↺ | status | cpu | memory |
|----|-------------|------|---|--------|-----|--------|
| 0  | backend-api | fork | 1 | online | 0%  | 55.7mb |

## 2. ใช้คำสั่งสร้างใบรับรอง ssl

[illegible]

### 3.กรอกข้อมูลตามตัวอย่าง

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

```
Country Name (2 letter code) [AU]:TH
State or Province Name (full name) [Some-State]:chiang-mai
Locality Name (eg, city) []:moung
Organization Name (eg, company) [Internet Widgits Pty Ltd]:se-rmutl
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:10.211.55.6
Email Address []:admin@se-rmutl.net
```

#### 4.ทำการ gen รหัส ของ openssl

```
supawit@backend2-66543210031:/var/www/back-end$ sudo openssl dhparam -out /etc/nginx/dhparam.pem 4096
Generating DH parameters, 4096 bit long safe prime
```

นาย ศกวิชญ์ แก่นนาค

## 5. ทำการตรวจสอบ selfsign.key และ selfsigned

```
supawit@backend2-66543210031:/var/www/back-end$ sudo ls -l /etc/ssl/private/nginx-selfsigned.key
-rw----- 1 root root 1704 Feb  7 04:29 /etc/ssl/private/nginx-selfsigned.key
supawit@backend2-66543210031:/var/www/back-end$ sudo ls -l /etc/ssl/certs/nginx-selfsigned.crt
-rw-r--r-- 1 root root 1399 Feb  7 04:33 /etc/ssl/certs/nginx-selfsigned.crt
```

## 6. ทำการเข้าไปแก้ไข เพิ่มในไฟล์ self-signed.conf

```
supawit@backend2-66543210031:/var/www/back-end$ sudo nano /etc/nginx/snippets/self-signed.conf
```

## 7. ข้างในไฟล์ self-sign.conf

```
GNU nano 6.2 /etc/nginx/snippets/self-signed.conf
ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;
ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;
```

## 8. ทำการแก้ไขไฟล์ ssl-params.conf

```
supawit@backend2-66543210031:/var/www/back-end$ sudo nano /etc/nginx/snippets/ssl-params.conf
```

## 9. ข้างในไฟล์ ไฟล์ ssl-params.conf แก้ไขตามภาพ

```
GNU nano 6.2 /etc/nginx/snippets/ssl-params.conf
ssl_protocols TLSv1.3;
ssl_prefer_server_ciphers on;
ssl_dhparam /etc/nginx/dhparam.pem;
ssl_ciphers ECDH+AESGCM:EDH+AESGCM;
ssl_ecdh_curve secp384r1;
ssl_session_timeout 10m;
ssl_session_cache shared:SSL:10m;
ssl_session_tickets off;
ssl_stapling on;
ssl_stapling_verify on;
resolver 8.8.8.8 8.8.4.4 valid=300s;
resolver_timeout 5s;
# Disable strict transport security for now. You can uncomment the following
# line if you understand the implications.
#add_header Strict-Transport-Security "max-age=63072000; includeSubDomains; preload";
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;
add_header X-XSS-Protection "1; mode=block";
```

## 10. เข้า path /etc/nginx/sites-available เพื่อแก้ไข apimoviebackend

```
supawit@backend2-66543210031:/etc/nginx/sites-available$ sudo nano apimoviebackend
```

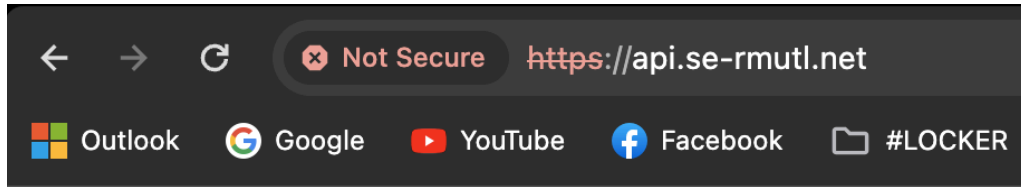
## 11. ข้างในไฟล์ apimoviebackend

```
GNU nano 6.2 apimoviebackend
server {
    listen 443 ssl;
    listen [::]:443 ssl;
    include snippets/self-signed.conf;
    include snippets/ssl-params.conf;
    server_name api.se-rmutl.net;

    location / {
        proxy_pass http://127.0.0.1:4000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

นาย ศุภวิชญ์ แก่นนาค

## 12.ทดสอบเช็ค api ในเว็บ



# Welcome to API Back-end Ver. 1.0.0

**ฝั่ง front-end**

### 13.เช็ค pm2 status ของ front end

```
supawit@backend1-66543210031:~$ pm2 status
```

| id | name      | namespace | version | mode | pid  | uptime | ⚡ | status | cpu | mem    | user    | watching |
|----|-----------|-----------|---------|------|------|--------|---|--------|-----|--------|---------|----------|
| 0  | front-end | default   | N/A     | fork | 6125 | 4D     | 0 | online | 0%  | 76.7mb | supawit | disabled |

14. ใช้คำสั่ง `sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/nginx-selfsigned.key -out /etc/ssl/certs/nginx-selfsigned.crt`  
สร้างใบรับรอง ssl

```
[sudo] password for supawit:  
.....  
.....  
.....  
.....  
-----  
  
You are about to be asked to enter information that will be incorporated  
into your certificate request.  
  
What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.  
  
-----  
Country Name (2 letter code) [AU]:TH  
State or Province Name (full name) [Some-State]:chiang-mai  
Locality Name (eg, city) []:mueang  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:se-remutl  
Organizational Unit Name (eg, section) []:  
Common Name (e.g. server FQDN or YOUR name) []:10.211.55.5  
Email Address []:admin@se-remutl.net
```

15. ใช้คำสั่ง `sudo openssl dhparam -out /etc/nginx/dhparam.pem 4096` เพื่อ genkey openssl

```
supawit@backend1-66543210931:/var/www/front-end$ sudo openssl dhparam -out /etc/nginx/dhparams.pem 4096
Generating DH parameters, 4096 bit long safe prime
```

นาย ศกวิชญ์ แก่นนาค

16. ใช้คำสั่ง `sudo nano /etc/nginx/snippets/self-signed.conf`

```
supawit@backend1-66543210031:/var/www/front-end$ sudo nano /etc/nginx/snippets/self-signed.conf
```

17. ข้างในไฟล์ `self.sign.conf` เขียนโค้ดตามตัวอย่าง

```
GNU nano 6.2 /etc/nginx/snippets/self-signed.conf *
ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;
ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;
```

18. ใช้คำสั่ง `sudo nano /etc/nginx/snippets/ssl-params.conf`

```
supawit@backend1-66543210031:/var/www/front-end$ sudo nano /etc/nginx/snippets/ssl-params.conf
```

19. ข้างในไฟล์ `ssl-params` เขียนโค้ดตามตัวอย่าง

```
GNU nano 6.2 /etc/nginx/snippet
ssl_protocols TLSv1.3;
ssl_prefer_server_ciphers on;
ssl_dhparam /etc/nginx/dhparam.pem;
ssl_ciphers EECDH+AESGCM:EDH+AESGCM;
ssl_ecdh_curve secp384r1;
ssl_session_timeout 10m;
ssl_session_cache shared:SSL:10m;
ssl_session_tickets off;
ssl_stapling on;
ssl_stapling_verify on;
resolver 8.8.8.8 8.8.4.4 valid=300s;
resolver_timeout 5s;
# Disable strict transport security for now. You can uncomment the following
# line if you understand the implications.
#add_header Strict-Transport-Security "max-age=63072000; includeSubDomains; preload";
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;
add_header X-XSS-Protection "1; mode=block";
```

20. ใช้คำสั่ง `cd /etc/nginx/sites-available`

```
supawit@backend1-66543210031:~$ cd /etc/nginx/sites-available/
```

21. ใช้คำสั่ง `nano` แก้ไขไฟล์ `frontendmovie`

```
supawit@backend1-66543210031:/etc/nginx/sites-available$ sudo nano frontendmovie
```

นาย ศกวิชญ์ แก่นนาค

22. เข้า Terminal ของคอมพิวเตอร์ imac พิมพ์ sudo nano /etc/hosts แล้วเพิ่ม ip กับ domain

10.211.55.5 movie.se-rmutl.net

23. ให้ cd /etc/nginx/sites-available/ และแก้ไขไฟล์ frontendmovie

```
GNU nano 6.2                                     fro
# Virtual Host configuration for example.com
#
# You can move that to a different file under sites-available/ and symlink that
# to sites-enabled/ to enable it.
#
server {
    #listen 80;
    #listen [::]:80;
    listen 443 ssl;
    listen [::]:443 ssl;
    include snippets/self-signed.conf;
    include snippets/ssl-params.conf;
    server_name movie.se-rmutl.net;

    location / {
        proxy_pass http://127.0.0.1:3001; # !!! - change to your app port
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

24. ทดสอบเข้า เว็บไซต์

| Movie Id | Title | Genre                  | Director      | Release |
|----------|-------|------------------------|---------------|---------|
| 1        | 1234  | 1234                   | 1234          | 1234    |
| 2        | Joker | psychological thriller | Todd Phillips | 2019    |
| 3        | test  | test                   | test          | 2111    |

### ส่วน vscode โค้ด หน้า movieservices ของ front end

```
const env = require('../components/env');
const config = require('../components/apiConfig')[env];
const token = config.bearer_token;
export async function getAllMovies() {
  try {
    const response = await fetch(config.API_URL + '/api/movie/all', {
      headers: {
        'Authorization': `Bearer ${token}`
      }
    });
    return await response.json();
  } catch (error) {
    return [];
  }
}

export async function createMovie(data) {
  try {
    const response = await fetch(config.API_URL + '/api/movie/insert', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
        'Authorization': `Bearer ${token}`
      },
      body: JSON.stringify(data)
    });
    return await response.json();
  } catch (error) {
    return { error: 'Failed to create movie' };
  }
}

export async function getMovieSearch(search_text) {
  try {
    const response = await fetch(config.API_URL
+ '/api/movie/search?search_text=${search_text}', {
      headers: {
        'Authorization': `Bearer ${token}`
      }
    });
    return await response.json();
  } catch (error) {
    return [];
  }
}
```

## นาย ศกวิชญ์ แก่นนาค

### ส่วนของ back-end

#### โค้ดหน้า index.js ของ api

```
const hapi = require("@hapi/hapi");
const env = require("../env.js");
const Movies = require("../respository/movie");
const AuthBearer = require("hapi-auth-bearer-token"); //----- AuthBearer -----
const express = require("express");
const app = express();
const path = require("path");
bodyParser = require("body-parser");
//-----
const api_port = 4000;
const web_port = 4001;
//----- hapi -----
console.log("Running Environment: " + env);
const init = async () => {
  const server = hapi.Server({
    port: api_port,
    host: "0.0.0.0",
    routes: {
      "cors": true,
      //cors: {
      //  origin: ["Access-Control-Allow-Origin", "https://api.se-rmutl.net/"],
      //  headers: ["Accept", "Content-Type"],
      //  additionalHeaders: ["X-Requested-With"],
      // },
    },
  });
  //-----
  await server.register(require("@hapi/inert"));
  //----- AuthBearer -----
  await server.register(AuthBearer);
  server.auth.strategy("simple", "bearer-access-token", {
    allowQueryToken: true, // optional, false by default
    validate: async (request, token, h) => {
      // here is where you validate your token
      // comparing with token from your database for example
      const isValid = token === "1234567890";
      const credentials = { token };
      const artifacts = { test: "info" };
      return { isValid, credentials, artifacts };
    },
  },
```

## นาย ศกวิชญ์ แก่นนาค

```
});  
server.auth.default("simple");  
//----- AuthBearer -----  
/*  
  server.route({  
    method: "GET",  
    path: "/",  
    config: {  
      cors: {  
        origin: ['*'],  
        additionalHeaders: ['cache-control', 'x-requested-width']  
      }  
    },  
    handler: () => {  
      return '<h3> Welcome to API Back-end Ver. 1.0.0</h3>';  
    }  
  });  
*/  
server.route({  
  method: "GET",  
  path: "/",  
  config: {  
    cors: {  
      origin: ["*"],  
      additionalHeaders: ["cache-control", "x-requested-width"],  
      credentials: true,  
    },  
  },  
  handler: () => {  
    return "<h3> Welcome to API Back-end Ver. 1.0.0</h3>";  
  },  
});  
//API: http://localhost:3001/api/movie/all  
server.route({  
  method: "GET",  
  path: "/api/movie/all",  
  config: {  
    cors: {  
      origin: ["*"],  
      additionalHeaders: ["cache-control", "x-requested-width"],  
      credentials: true,  
    },  
  },  
});
```



## นาย ศกวิชญ์ แก่นนาค

```
    },
    handler: async function (request, reply) {
        //var param = request.query;
        //const category_code = param.category_code;
        try {
            const respondedata = await Movies.MovieRepo.getMovieList();
            if (respondedata.error) {
                return respondedata.errMessage;
            } else {
                return respondedata;
            }
        } catch (err) {
            server.log(["error", "home"], err);
            return err;
        }
    },
});

server.route({
    method: "GET",
    path: "/api/movie/search",
    config: {
        cors: {
            origin: ["*"],
            additionalHeaders: ["cache-control", "x-requested-width"],
            credentials: true,
        },
    },
    handler: async function (request, reply) {
        var param = request.query;
        let search_text = param.search_text;
        //const title = param.title;
        console.log("search_text: " + search_text);
        if (search_text === undefined) {
            console.log("search_text is undefined ");
            search_text = "N/A";
            console.log("search_text: " + search_text);
        }
        //if (typeof myVar !== 'undefined')
        //myVar === undefined
        try {
            const respondedata = await Movies.MovieRepo.getMovieSearch(search_text);
            if (respondedata.error) {
```

## นาย ศกวิชญ์ แก่นนาค

```
        return responsedata.errorMessage;
    } else {
        return responsedata;
    }
} catch (err) {
    server.log(["error", "home"], err);
    return err;
}
},
});
server.route({
    method: "POST",
    path: "/api/movie/insert",
    config: {
        payload: {
            multipart: true,
        },
        cors: {
            origin: ["*"],
            additionalHeaders: ["cache-control", "x-requested-width"],
            credentials: true,
        },
    },
},
handler: async function (request, reply) {
    const { title, genre, director, release_year } = request.payload;
    console.log("request.payload: " + JSON.stringify(request.payload));
    /*
        const myObj = JSON.parse(request.payload);
        const title = myObj.title;
        const genre = myObj.genre;
        const director = myObj.director;
        const release_year = myObj.release_year;
    */
    //console.log("myObj.title: " + myObj.title);
    try {
        const responsedata = await Movies.MovieRepo.postMovie(
            title,
            genre,
            director,
            release_year
        );
    }
    if (responsedata.error) {
```

## นาย ศกวิชญ์ แก่นนาค

```
        return responsedata;
    } else {
        return responsedata;
    }
} catch (err) {
    server.log(["error", "home"], err);
    return err;
}
},
});
await server.start();
console.log("API Server running on %s", server.info.uri);
//-----
};
process.on("unhandledRejection", (err) => {
    console.log(err);
    process.exit(1);
});
init();
```

## 25.ใช้ postman get api เมื่อไม่กรอก token

HTTP ENGSE203 / front-end / hello

GET https://api.se-rmutl.net/api/movie/all

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Type Bearer ...

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Heads up! These parameters hold sensitive data. To keep collaborative environment, we recommend using variables

Token Token

Body Cookies Headers (10) Test Results 401 Unauthorized

Pretty Raw Preview Visualize JSON

```
1 {
2   "statusCode": 401,
3   "error": "Unauthorized",
4   "message": "Missing authentication"
5 }
```

## 26.ใช้ postman get api เมื่อกรอก token

HTTP ENGSE203 / front-end / hello

GET https://api.se-rmutl.net/api/movie/all

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Type Bearer ...

Heads up! These parameters hold sensitive data. To keep this data secure in a collaborative environment, we recommend using variables. Learn more about [variables](#).

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Token 1234567890

Body Cookies Headers (13) Test Results 200 OK 13 ms 67

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "title": "1234",
4     "genre": "1234",
5     "director": "1234",
6     "release_year": 1234
7   },
8   {
9     "title": "Joker",
10    "genre": "psychological thriller",
11    "director": "Todd Phillips",
12    "release_year": 2019
13  },
14  {
15    "title": "test",
16    "genre": "test",
17    "director": "test",
18    "release_year": 2111
19  }
20 ]
```

## 26.ทดสอบเข้าเว็บ api

Not Secure https://api.se-rmutl.net

Outlook Google YouTube Facebook #LOCKER #LOCKER2 #ดูค้างคา #มันจะ #ดูจบแล้ว หน้าหลัก | ระบบรับส... เข้าสู่ระบบ

```
{"statusCode":401,"error":"Unauthorized","message":"Missing authentication"}
```