

# เอกสาร API Outlab Module - iHospital System

## ภาพรวมระบบ (Overview)

Outlab Module เป็นส่วนหนึ่งของระบบ iHospital ที่ใช้สำหรับจัดการผลตรวจทางห้องปฏิบัติการจากภายนอก (External Laboratory Results) โดยระบบนี้ช่วยให้สามารถอัปโหลด จัดเก็บ และเรียกดูผลตรวจจากห้องปฏิบัติการภายนอกได้อย่างมีประสิทธิภาพ

## ข้อกำหนดเบื้องต้น (Prerequisites)

- API Base URL: `https://ihospital.kkh.go.th`
- API Version: v2
- Protocol: HTTPS
- Authentication: **IP Whitelist** - ต้องแจ้ง IP Address ให้ผู้ดูแลระบบเพิ่มใน whitelist
- Content-Type: `multipart/form-data` สำหรับการอัปโหลดไฟล์
- File Size Limit: ขนาดไฟล์ตามที่กำหนดในระบบ

## ระบบ Authentication

- ใช้วิธี **IP Whitelist** แทน API Token
- Vendor ต้องแจ้ง IP Address ของเซิร์ฟเวอร์ที่จะเชื่อมต่อ
- ระบบจะอนุญาตเฉพาะ IP ที่อยู่ใน whitelist เท่านั้น
- หาก IP ไม่ได้รับอนุญาต จะได้รับ Error 401

## API Endpoints

### 1. Upload Outlab Result

อัปโหลดไฟล์ผลตรวจทางห้องปฏิบัติการจากภายนอกเข้าสู่ระบบ

#### Endpoint Details

- **Method:** `POST`
- **URL:** `https://ihospital.kkh.go.th/api/v2/interface/lab/outlab/{labNo}`
- **Content-Type:** `multipart/form-data`

#### Parameters

URL Parameters

Parameter	Type	Required	Description
labNo	string	✓	หมายเลขใบสั่งตรวจ (Laboratory Order Number)

Form Data Parameters

Parameter	Type	Required	Description
type	string	✓	ประเภทของผลตรวจ (เช่น "BLOOD", "URINE", "XRAY", "CT", "MRI")
labcode	string	✓	รหัสห้องปฏิบัติการ (เช่น "LAB001", "RAD002")
labname	string	✓	ชื่อห้องปฏิบัติการ (เช่น "ห้องปฏิบัติการกลาง", "แผนกรังสีวิทยา")
file	file	✓	ไฟล์ผลตรวจ (รองรับ PDF, JPG, PNG, DICOM)

Response Codes

✔ Success Response (200 OK)

```
json
{
  "create_date": "2025-08-26 11:48:43",
  "module_name": "outlab",
  "ref": "680284033",
  "ref2": "680284033",
  "hn": "499999998",
  "vn": "S5063826",
  "an": "",
  "path": "68/08/",
  "filename": "680826-ba4c8f2a-9f9c-43e8-9046-384aef753a17.pdf",
  "original_filename": "outlab-api.pdf",
  "file_type": "application/pdf",
  "file_extension": ".pdf",
  "file_size": 2130,
  "storage": "s1",
  "uuid": "02928e1c-b9b6-4699-b99a-e1f1db1ad277"
}
```

✗ Error Responses

Status Code	Scenario	Response Body
401	IP ไม่อยู่ใน Whitelist	<code>{"statusCode": 401, "ip": "10.3.42.128", "message": "You are not authorized to access this route."}</code>
400	ไม่ได้แนบไฟล์	<code>{"statusCode": 400, "message": "No file uploaded."}</code>
500	Lab Number ไม่มีในระบบ	<code>{"statusCode": 500, "message": "Error uploading to storage: Lab No not found in the database."}</code>

Response Fields Description (Success Case)

Field	Type	Description
create_date	string	วันที่สร้างไฟล์ในระบบ
module_name	string	ชื่อโมดูล (คงที่: "outlab")
ref	string	หมายเลขอ้างอิงหลัก
ref2	string	หมายเลขอ้างอิงรอง
hn	string	Hospital Number (HN) ของผู้ป่วย
vn	string	Visit Number (VN) ของการรักษา
an	string	Admission Number (AN) กรณีผู้ป่วยใน
path	string	Path ที่จัดเก็บไฟล์ในระบบ
filename	string	ชื่อไฟล์ที่ระบบสร้างขึ้น
original_filename	string	ชื่อไฟล์ต้นฉบับที่อัปโหลด
file_type	string	MIME Type ของไฟล์
file_extension	string	นามสกุลไฟล์
file_size	number	ขนาดไฟล์ (bytes)
storage	string	ชื่อ Storage ที่จัดเก็บ
uuid	string	Unique Identifier ของไฟล์

## ตัวอย่างการใช้งาน (Code Examples)

### 1. cURL Command Line

```
bash

# Basic Upload (ไม่ต้องใส่ Authorization เพราะใช้ IP Whitelist)
curl -X POST "https://ihospital.kkh.go.th/api/v2/interface/lab/outlab/LAB2024001234" \
  -F "type=BLOOD" \
  -F "labcode=LAB001" \
  -F "labname=ห้องปฏิบัติการกลาง" \
  -F "file=@/path/to/result.pdf"

# With verbose output for debugging
curl -v -X POST "https://ihospital.kkh.go.th/api/v2/interface/lab/outlab/LAB2024001234" \
  -F "type=BLOOD" \
  -F "labcode=LAB001" \
  -F "labname=ห้องปฏิบัติการกลาง" \
  -F "file=@/path/to/result.pdf"

# Check your IP address (สำหรับแจ้งผู้ดูแลระบบ)
curl ifconfig.me
```

## 2. C# (.NET 6+)

cssharp

```
using System;
using System.IO;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Threading.Tasks;
using Newtonsoft.Json;

public class OutlabApiClient
{
    private readonly HttpClient _httpClient;
    private const string BASE_URL = "https://ihospital.kkh.go.th/api/v2/interface/lab/outlab";

    public OutlabApiClient()
    {
        _httpClient = new HttpClient();
        // ไม่ต้องใส่ Authorization Header เพราะใช้ IP Whitelist
    }

    public async Task<OutlabResponse> UploadLabResultAsync(
        string labNo,
        string type,
        string labCode,
        string labName,
        string filePath)
    {
        try
        {
            using var form = new MultipartFormDataContent();

            // Add form fields
            form.Add(new StringContent(type), "type");
            form.Add(new StringContent(labCode), "labcode");
            form.Add(new StringContent(labName), "labname");

            // Add file
            using var fileStream = File.OpenRead(filePath);
            using var fileContent = new StreamContent(fileStream);
            fileContent.Headers.ContentType = MediaTypeHeaderValue.Parse("application/pdf");
            form.Add(fileContent, "file", Path.GetFileName(filePath));

            // Send request
            var response = await _httpClient.PostAsync($"{BASE_URL}/{labNo}", form);
            var responseContent = await response.Content.ReadAsStringAsync();

            if (response.IsSuccessStatusCode)
            {
                Console.WriteLine($"Success: {responseContent}");
            }
        }
    }
}
```

```

        return JsonConvert.DeserializeObject<OutlabResponse>(responseContent);
    }
    else if (response.StatusCode == System.Net.HttpStatusCode.Unauthorized)
    {
        var errorResponse = JsonConvert.DeserializeObject<ErrorResponse>(responseContent);
        throw new UnauthorizedAccessException($"IP {errorResponse.Ip} not in whitelist");
    }
    else
    {
        var errorResponse = JsonConvert.DeserializeObject<ErrorResponse>(responseContent);
        throw new Exception(errorResponse.Message);
    }
}
catch (Exception ex)
{
    Console.WriteLine($"Exception: {ex.Message}");
    throw;
}
}
}

```

*// Response Models*

```

public class OutlabResponse
{
    [JsonProperty("create_date")]
    public string CreateDate { get; set; }

    [JsonProperty("module_name")]
    public string ModuleName { get; set; }

    [JsonProperty("ref")]
    public string Ref { get; set; }

    [JsonProperty("ref2")]
    public string Ref2 { get; set; }

    [JsonProperty("hn")]
    public string Hn { get; set; }

    [JsonProperty("vn")]
    public string Vn { get; set; }

    [JsonProperty("an")]
    public string An { get; set; }

    [JsonProperty("path")]
    public string Path { get; set; }
}

```

```
[JsonProperty("filename")]
public string Filename { get; set; }

[JsonProperty("original_filename")]
public string OriginalFilename { get; set; }

[JsonProperty("file_type")]
public string FileType { get; set; }

[JsonProperty("file_extension")]
public string FileExtension { get; set; }

[JsonProperty("file_size")]
public int FileSize { get; set; }

[JsonProperty("storage")]
public string Storage { get; set; }

[JsonProperty("uuid")]
public string Uuid { get; set; }
}

public class ErrorResponse
{
    [JsonProperty("statusCode")]
    public int StatusCode { get; set; }

    [JsonProperty("ip")]
    public string Ip { get; set; }

    [JsonProperty("message")]
    public string Message { get; set; }
}

// Usage Example
class Program
{
    static async Task Main(string[] args)
    {
        var client = new OutlabApiClient();

        try
        {
            var result = await client.UploadLabResultAsync(
                labNo: "LAB2024001234",
                type: "BLOOD",
```

```
        labCode: "LAB001",
        labName: "ห้องปฏิบัติการกลาง",
        filePath: @"C:\LabResults\result.pdf"
    );

    Console.WriteLine($"File uploaded successfully!");
    Console.WriteLine($"UUID: {result.Uuid}");
    Console.WriteLine($"HN: {result.Hn}, VN: {result.Vn}");
}
catch (UnauthorizedAccessException ex)
{
    Console.WriteLine($"Authorization Error: {ex.Message}");
    Console.WriteLine("Please contact admin to add your IP to whitelist");
}
catch (Exception ex)
{
    Console.WriteLine($"Error: {ex.Message}");
}
}
```

### 3. JavaScript (Vanilla)

javascript



```
// Function to upload lab result (ไม่ต้องใส่ Authorization)
async function uploadLabResult(labNo, formData) {
  const url = `https://ihospital.kkh.go.th/api/v2/interface/lab/outlab/${labNo}`;

  try {
    const response = await fetch(url, {
      method: 'POST',
      body: formData
    });

    const result = await response.json();

    if (response.ok) {
      console.log('Upload successful:', result);
      return result;
    } else {
      // Handle different error cases
      if (result.statusCode === 401) {
        console.error(`IP ${result.ip} not authorized. Contact admin to whitelist.`);
      } else if (result.statusCode === 400) {
        console.error('No file uploaded');
      } else if (result.statusCode === 500) {
        console.error('Lab number not found in database');
      }
      throw new Error(result.message || 'Upload failed');
    }
  } catch (error) {
    console.error('Error uploading file:', error);
    throw error;
  }
}
```

// HTML Form Example

```
const html = `
<form id="uploadForm">
  <input type="text" id="labNo" placeholder="Lab Number" required>
  <input type="text" id="type" placeholder="Type (BLOOD/URINE/etc)" required>
  <input type="text" id="labcode" placeholder="Lab Code" required>
  <input type="text" id="labname" placeholder="Lab Name" required>
  <input type="file" id="file" accept=".pdf,.jpg,.png" required>
  <button type="submit">Upload</button>
</form>
<div id="result"></div>

<script>
document.getElementById('uploadForm').addEventListener('submit', async (e) => {
  e.preventDefault();
```

```
const labNo = document.getElementById('labNo').value;
const formData = new FormData();
const resultDiv = document.getElementById('result');

formData.append('type', document.getElementById('type').value);
formData.append('labcode', document.getElementById('labcode').value);
formData.append('labname', document.getElementById('labname').value);
formData.append('file', document.getElementById('file').files[0]);

try {
  const result = await uploadLabResult(labNo, formData);
  resultDiv.innerHTML = '<div style="color: green;">Upload successful!<br>' +
    'UUID: ' + result.uuid + '<br>' +
    'HN: ' + result.hn + ', VN: ' + result.vn + '</div>';
} catch (error) {
  resultDiv.innerHTML = '<div style="color: red;">Upload failed: ' + error.message + '</div>';
}
});
</script>
`;
```

## 4. Node.js

javascript

```
const fs = require('fs');
const FormData = require('form-data');
const axios = require('axios');

class OutlabApiClient {
  constructor() {
    this.baseUrl = 'https://ihospital.kkh.go.th/api/v2/interface/lab/outlab';
    // ไม่ต้องใช้ API Token เพราะใช้ IP Whitelist
  }

  async uploadLabResult(labNo, type, labCode, labName, filePath) {
    const form = new FormData();

    // Add form fields
    form.append('type', type);
    form.append('labcode', labCode);
    form.append('labname', labName);

    // Add file
    form.append('file', fs.createReadStream(filePath));

    try {
      const response = await axios.post(
        `${this.baseUrl}/${labNo}`,
        form,
        {
          headers: form.getHeaders()
        }
      );

      console.log('Upload successful!');
      console.log('UUID:', response.data.uuid);
      console.log('HN:', response.data.hn, 'VN:', response.data.vn);
      return response.data;
    } catch (error) {
      if (error.response) {
        const { statusCode, message, ip } = error.response.data;

        if (statusCode === 401) {
          console.error(`IP ${ip} not in whitelist. Contact admin.`);
        } else if (statusCode === 400) {
          console.error('No file uploaded');
        } else if (statusCode === 500) {
          console.error('Lab number not found:', message);
        }
      }
    }
  }
}
```

```

        throw new Error(message);
    } else {
        console.error('Network error:', error.message);
        throw error;
    }
}
}

// Helper function to check server IP
async checkMyIP() {
    try {
        const response = await axios.get('https://api.ipify.org?format=json');
        console.log('Your server IP:', response.data.ip);
        console.log('Send this IP to admin for whitelist');
        return response.data.ip;
    } catch (error) {
        console.error('Cannot get IP:', error.message);
    }
}
}

```

*// Usage Example*

```

async function main() {
    const client = new OutlabApiClient();

    // Check server IP (for whitelist request)
    await client.checkMyIP();

    try {
        const result = await client.uploadLabResult(
            'LAB2024001234', // Lab number ที่มีอยู่ในระบบ
            'BLOOD',
            'LAB001',
            'ห้องปฏิบัติการกลาง',
            './lab_result.pdf'
        );

        console.log('Result:', result);

    } catch (error) {
        console.error('Error:', error.message);
    }
}

```

*// Alternative using native fetch (Node.js 18+)*

```

const { readFile } = require('fs/promises');

```

```
async function uploadWithFetch(labNo, type, labCode, labName, filePath) {
  const formData = new FormData();

  const fileBuffer = await readFile(filePath);
  const file = new Blob([fileBuffer], { type: 'application/pdf' });

  formData.append('type', type);
  formData.append('labcode', labCode);
  formData.append('labname', labName);
  formData.append('file', file, 'result.pdf');

  try {
    const response = await fetch(
      `https://ihospital.kkh.go.th/api/v2/interface/lab/outlab/${labNo}`,
      {
        method: 'POST',
        body: formData
      }
    );

    const result = await response.json();

    if (!response.ok) {
      throw new Error(result.message);
    }

    console.log('Success:', result);
    return result;

  } catch (error) {
    console.error('Error:', error.message);
    throw error;
  }
}

// Run the example
if (require.main === module) {
  main();
}
```

## Error Handling Best Practices

### 1. Retry Logic

javascript

```
async function uploadWithRetry(labNo, formData, apiToken, maxRetries = 3) {
  let lastError;

  for (let i = 0; i < maxRetries; i++) {
    try {
      return await uploadLabResult(labNo, formData, apiToken);
    } catch (error) {
      lastError = error;
      console.log(`Attempt ${i + 1} failed, retrying...`);

      // Wait before retry (exponential backoff)
      await new Promise(resolve => setTimeout(resolve, 1000 * Math.pow(2, i)));
    }
  }

  throw lastError;
}
```

## 2. File Validation

javascript

```
function validateFile(file) {
  const maxSize = 10 * 1024 * 1024; // 10MB
  const allowedTypes = ['application/pdf', 'image/jpeg', 'image/png'];

  if (file.size > maxSize) {
    throw new Error('File size exceeds 10MB limit');
  }

  if (!allowedTypes.includes(file.type)) {
    throw new Error('Invalid file type. Allowed: PDF, JPG, PNG');
  }

  return true;
}
```

## Security Considerations

### 1. IP Whitelist Management

- แจ้ง IP Address ของ Server ที่จะเชื่อมต่อให้ผู้ดูแลระบบ
- หาก IP เปลี่ยน ต้องแจ้งผู้ดูแลระบบทันที
- ใช้ Static IP เพื่อความเสถียรของระบบ

- สามารถตรวจสอบ IP ปัจจุบันได้ด้วยคำสั่ง `curl ifconfig.me`

## 2. File Upload Security

- ตรวจสอบประเภทไฟล์ก่อนอัปโหลด
- จำกัดขนาดไฟล์ตามที่ระบบกำหนด
- Scan ไฟล์ด้วย Antivirus ถ้าเป็นไปได้

## 3. HTTPS Only

- ใช้ HTTPS เสมอ ไม่ใช้ HTTP
- ตรวจสอบ SSL Certificate

## 4. Input Validation

- ตรวจสอบ Lab Number ว่ามีอยู่ในระบบก่อนอัปโหลด
- Sanitize ข้อมูลก่อนส่ง

---

## Testing Checklist

### Unit Tests

- ☐ ทดสอบกับไฟล์ขนาดต่างๆ
- ☐ ทดสอบกับประเภทไฟล์ต่างๆ
- ☐ ทดสอบกับข้อมูลที่ไม่ครบถ้วน
- ☐ ทดสอบกับ Lab Number ที่ไม่ถูกต้อง

### Integration Tests

- ☐ ทดสอบการอัปโหลดไฟล์จริง
- ☐ ทดสอบ Error Handling
- ☐ ทดสอบ Retry Logic
- ☐ ทดสอบ Concurrent Uploads

### Performance Tests

- ☐ ทดสอบกับไฟล์ขนาดใหญ่ (5-10MB)
- ☐ ทดสอบการอัปโหลดพร้อมกันหลายไฟล์
- ☐ ทดสอบ Response Time

---

## FAQ และ Troubleshooting

### Q1: ได้รับ Error 401 "You are not authorized to access this route"

A: IP Address ของ Server ไม่อยู่ใน whitelist ให้ตรวจสอบ IP และแจ้งผู้ดูแลระบบเพื่อเพิ่มใน whitelist

### Q2: วิธีตรวจสอบ IP Address ของ Server

A: ใช้คำสั่ง `curl ifconfig.me` หรือ `curl https://api.ipify.org`

**Q3: ได้รับ Error 400 "No file uploaded"**

A: ตรวจสอบว่าได้แนบไฟล์ในฟิลด์ "file" และใช้ method POST แบบ multipart/form-data

**Q4: ได้รับ Error 500 "Lab No not found in the database"**

A: หมายเลข Lab Number ไม่มีในระบบ ตรวจสอบหมายเลขให้ถูกต้อง

**Q5: Upload สำเร็จ แต่ต้องการดูไฟล์ที่อัปโหลด**

A: ใช้ UUID หรือ filename ที่ได้รับจาก response เพื่อ query หาไฟล์ในระบบ

---

## Contact & Support

### Technical Support

- Email: [api-support@kkh.go.th](mailto:api-support@kkh.go.th)
- Phone: 043-123456 ต่อ 1234
- Line: @kkh\_api\_support

### Documentation Updates

- Version: 1.0.0
- Last Updated: January 2024
- API Version: v2

### Change Log

- v1.0.0 (Jan 2024): Initial release
- Support for PDF, JPG, PNG file types
- Maximum file size 10MB