



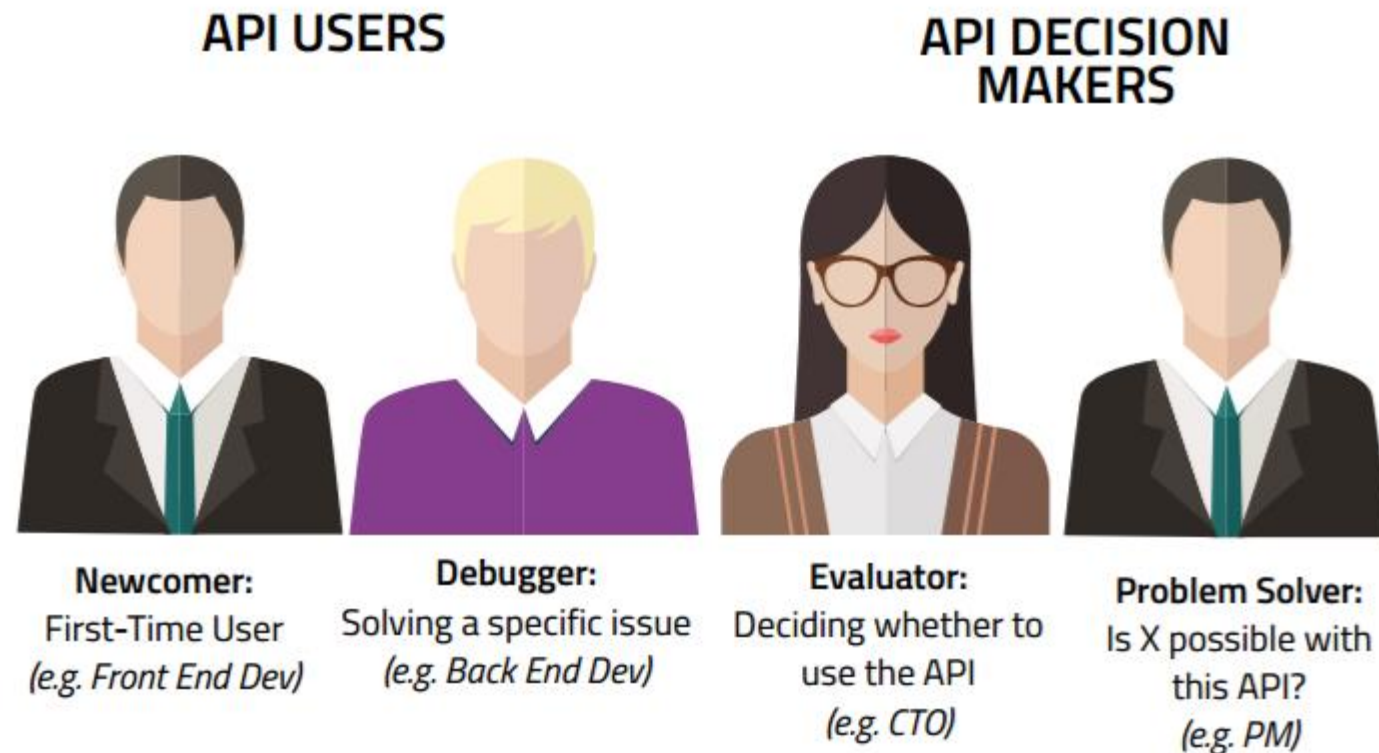
LẬP TRÌNH C# 6

BÀI 1: API DOCUMENT

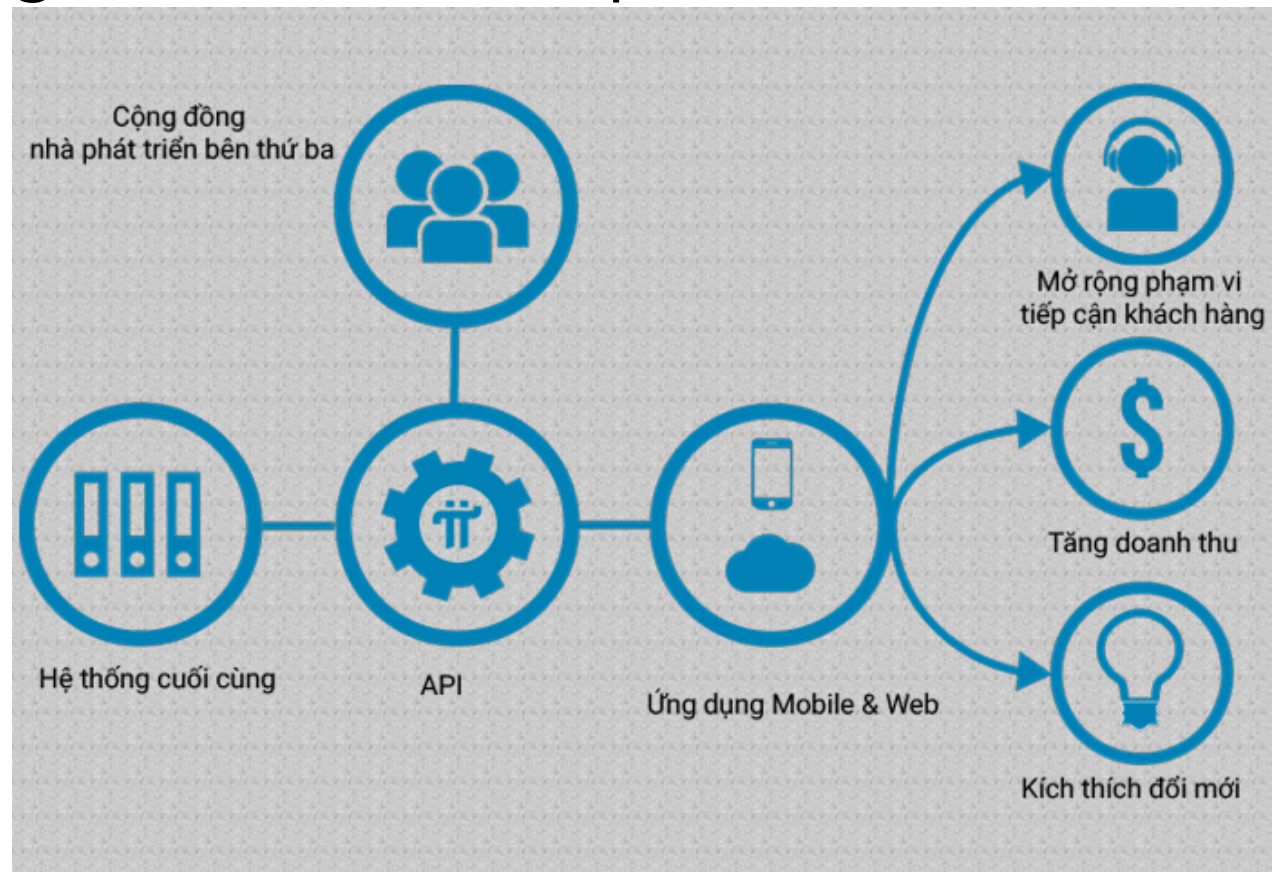
- ⦿ Tài liệu hướng dẫn sử dụng API
- ⦿ Công cụ Swagger
- ⦿ Swagger UI và ASP.NET Core Web API



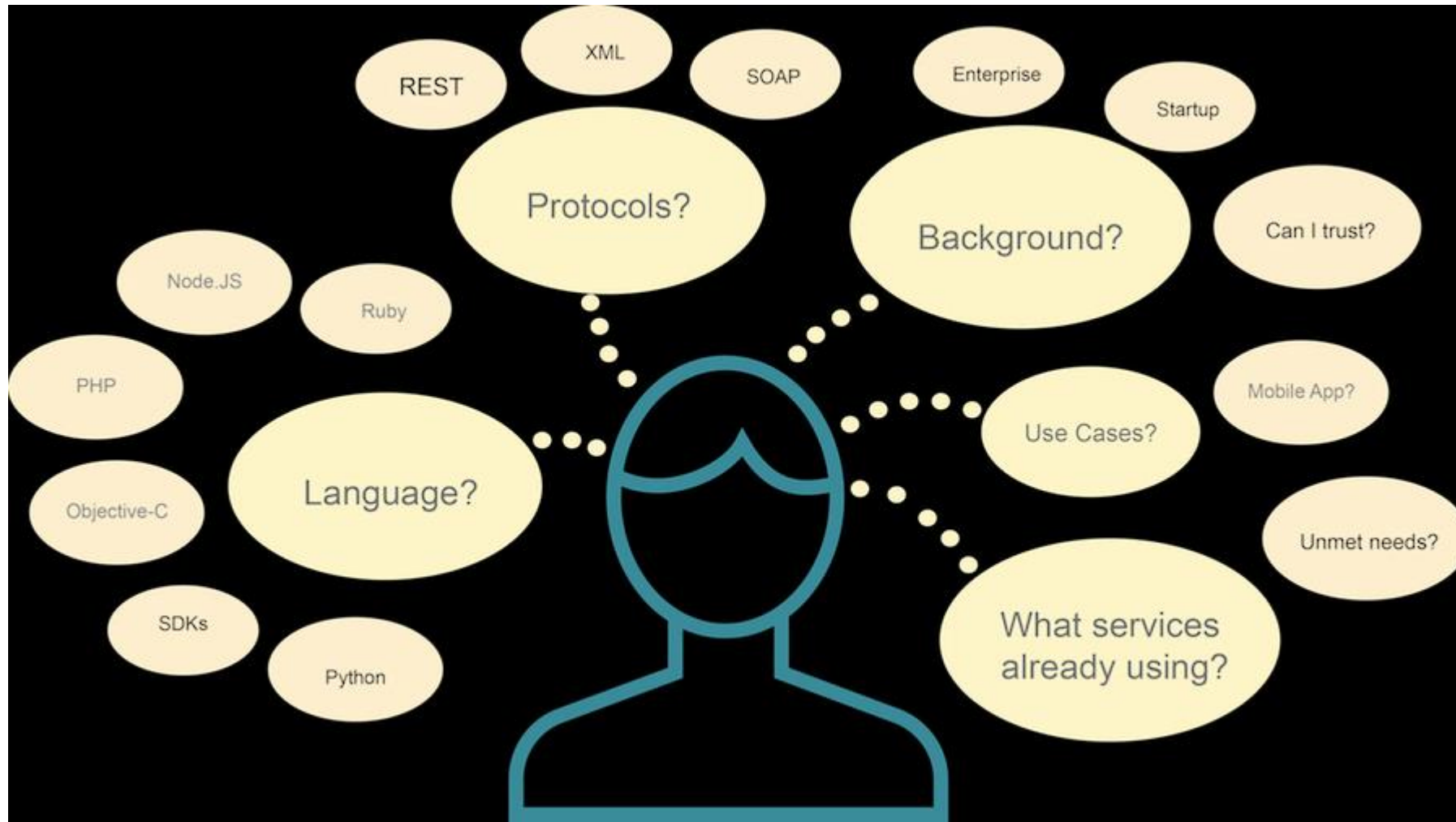
- ❑ Tài liệu hướng dẫn sử dụng API là một nội dung kỹ thuật, chứa tất cả các thông tin được yêu cầu để có thể làm việc với API



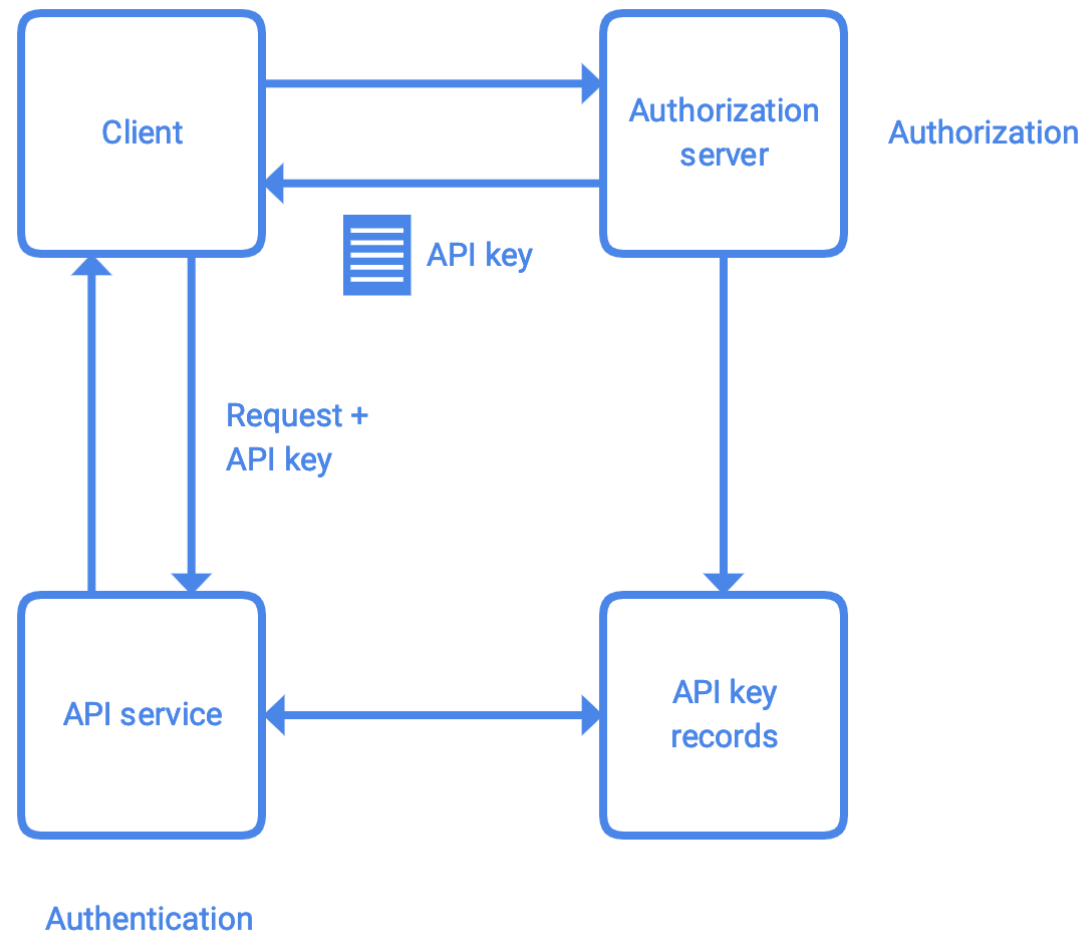
- ❑ Cải thiện trải nghiệm người dùng
- ❑ Giúp nhiều người biết đến API
- ❑ Tiết kiệm thời gian hỗ trợ và chi phí
- ❑ Dễ bảo trì



❑ Cân nhắc đối tượng sử dụng API



□ Authentication



□ API Resources: developer.wordpress.com/docs/api/1.1/get/me/

Resource Information

Method	GET
URL	https://public-api.wordpress.com/rest/v1.1/me
Requires authentication?	Yes

Query Parameters

Parameter	Type	Description
http_envelope	(bool)	<p>false: (default)</p> <p>true: Some environments (like in-browser JavaScript or Flash) block or divert responses with a non-200 HTTP status code. Setting this parameter will force the HTTP status code to always be 200. The JSON response is wrapped in an "envelope" containing the "real" HTTP status code and headers.</p>
pretty	(bool)	<p>false: (default)</p> <p>true: Output pretty JSON</p>
meta	(string)	Optional. Loads data from the endpoints found in the 'meta' part of the response. Comma-separated list. Example: meta=site,likes

[Resource Errors](#)

[Example](#)

Authentication

Users

List the users of a site.

Update details of a user of a site.

Get details of a user of a site by login.

Deletes or removes a user of a site.

Get a list of possible users to suggest for @mentions.

Get metadata about the current user.

Get list of current user's billing history and upcoming charges.

Get the current user's settings.

Update the current user's settings.

Get the current user's settings.

Update the current user's preferences.

Verify strength of a user's new password.

❑ Error Messages: mailchimp.com/developer/marketing/docs/errors/

MAILCHIMP MARKETING API

Overview
Guides
Documentation

Fundamentals
Methods and Parameters
E-Commerce
Errors

The basics
Error format
Common causes
Error glossary
Mobile SDK
API Reference

Error format

We expose API errors in two ways: standard HTTP response codes and human-readable messages in JSON format.

Here's an HTTP 405 error in both formats:

Error - HTTP ResponsePLAIN TEXT

```

HTTP/1.1 405 Method Not Allowed
Server: nginx
Content-Type: application/problem+json; charset=utf-8
Content-Length: 253
X-Request-Id: a1efb240-f8d8-40fe-a680-c3a5619a42e9
Link: <https://us2.api.mailchimp.com/schema/3.0/ProblemDetailDocument.json>; rel="describedBy"
Date: Thu, 17 Sep 2015 19:02:28 GMT
Connection: keep-alive
Set-Cookie: _AVESTA_ENVIRONMENT=prod; path=/

```

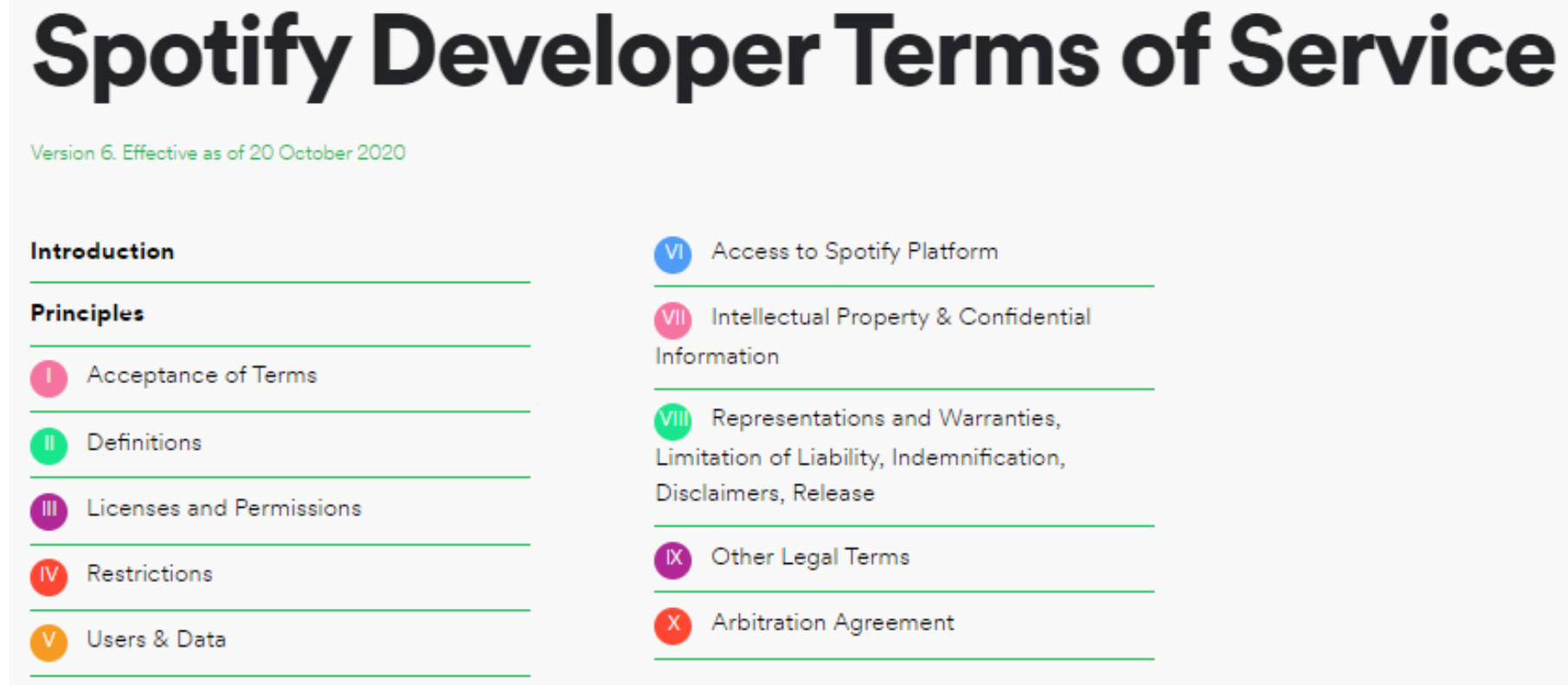
Error - JSONJSON

```

{ "type": "https://developer.mailchimp.com/documentation/mailchimp/guides/error-glossary/#405",
  "title": "Method Not Allowed",
  "status": 405,
  "detail": "The requested method and resource are not compatible. See the Allow header for this resource's a",
  "instance": ""

```


□ Terms of use



Spotify Developer Terms of Service	
Version 6. Effective as of 20 October 2020	
Introduction	VI Access to Spotify Platform
Principles	VII Intellectual Property & Confidential Information
I Acceptance of Terms	VIII Representations and Warranties, Limitation of Liability, Indemnification, Disclaimers, Release
II Definitions	IX Other Legal Terms
III Licenses and Permissions	X Arbitration Agreement
IV Restrictions	
V Users & Data	

<https://developer.spotify.com/terms/#iii>

❑ Changelog: <https://developer.github.com/changes/>

Antiope preview graduation

October 1, 2020 benemdon

The `antiope` preview is now an official part of the API. These preview headers are no longer required. The Checks API endpoints no longer require the `antiope` preview header.

```
application/vnd.github.antiope-preview+json
```

For more information, see the “Checks API.”

Thanks again to everyone that tried out these API features during the preview period.

Machine-man and sailor-v previews graduate

August 20, 2020 jakewilkins gallexi

Some API previews have graduated and are now an official part of the API. These preview headers are no longer required.

`machine-man` preview graduates

The GitHub App endpoints and endpoints that returned the `performed_via_github_app` property no longer require the `machine-man` preview.

Recent Posts

Antiope preview graduation
October 1, 2020

Machine-man and sailor-v previews graduate
August 20, 2020

GitHub Actions API - Introducing workflow
usage endpoints
May 15, 2020

Introducing the skipped check run and check
suite conclusion
May 8, 2020

Suspending GitHub App installations
May 1, 2020

Expiring user-to-server access tokens for
GitHub Apps
April 30, 2020

Replacing the GitHub Apps “Creating an
installation access token” endpoint
April 15, 2020

- ☐ Hạn chế thuật ngữ kỹ thuật chuyên sâu
- ☐ Tài liệu đầy đủ nhưng không quá phức tạp, khó hiểu
- ☐ Nguồn tài nguyên bổ sung, nguồn tham khảo
- ☐ Source code mẫu, tài liệu hướng dẫn nhanh

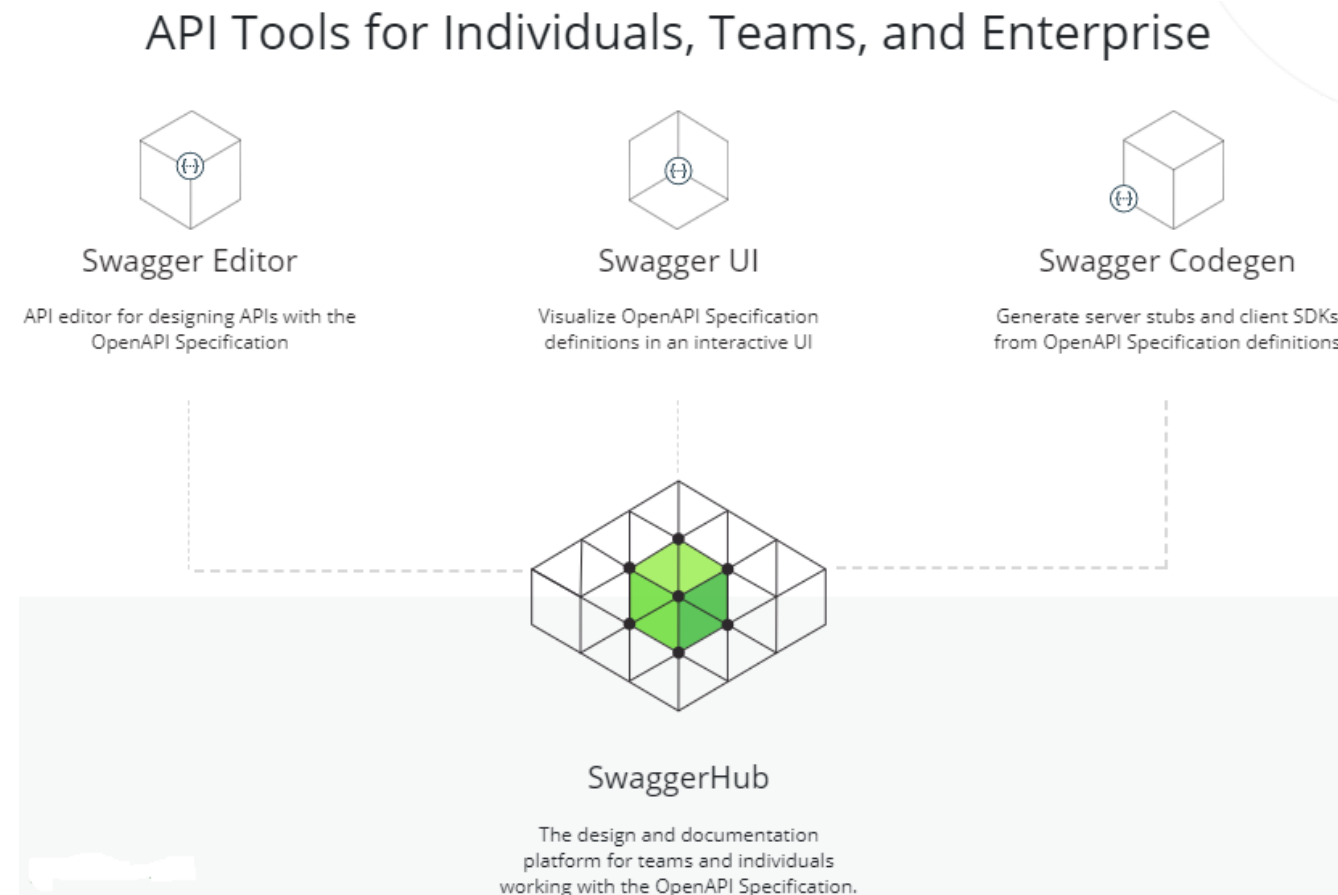


LẬP TRÌNH C# 6

BÀI 1: API DOCUMENT

- ❑ OpenAPI Specification là một định dạng mô tả API dành cho REST APIs. Một file OpenAPI cho phép bạn mô tả toàn bộ API bao gồm:
 - ❖ Cho phép những endpoints (/users) và cách thức hoạt động của mỗi endpoint (GET, POST, PUT, DELETE).
 - ❖ Các tham số đầu vào & đầu ra của từng API.
 - ❖ Phương thức xác thực.
 - ❖ Thông tin liên hệ, chứng chỉ (HTTP/ HTTPS), điều khoản sử dụng và những thông tin khác.
- ❑ API specifications có thể được viết bằng YAML hoặc JSON. Định dạng này dễ đọc, dễ hiểu cho cả người dùng lẫn ngôn ngữ máy tính.

- ❑ Swagger là một bộ công cụ mã nguồn mở để xây dựng OpenAPI specifications giúp chúng ta có thể thiết kế, xây dựng tài liệu và sử dụng REST APIs.



- ❑ Swagger UI là công cụ giúp tạo trang html css mô tả về các APIs được cấu hình bởi file .yaml. Ngoài ra, công cụ này còn cho phép mockup đến api đó để kiểm tra
- ❑ Viết document cho Swagger có hai cách tiếp cận chính
 - ❖ Top-down approach: thiết kế các API trước khi code
 - ❖ Bottom-up approach: từ các API có sẵn thiết kế file config để mô tả

❑ Demo với Swagger UI <http://petstore.swagger.io/>

Swagger Petstore ^{1.0.5}

[Base URL: petstore.swagger.io/v2]
<https://petstore.swagger.io/v2/swagger.json>

This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> or on [#swagger](http://irc.freenode.net). For this sample, you can use the api key `special-key` to test the authorization filters.

[Terms of service](#)

[Contact the developer](#)

[Apache 2.0](#)

[Find out more about Swagger](#)

Schemes

HTTPS

Authorize

pet Everything about your Pets

Find out more: <http://swagger.io>

store Access to Petstore orders

POST `/store/order` Place an order for a pet

GET `/store/order/{orderId}` Find purchase order by ID

DELETE `/store/order/{orderId}` Delete purchase order by ID

GET `/store/inventory` Returns pet inventories by status

```
swagger: "2.0"
info:
  description: "This is a sample server Petstore server. You can find out
    more about Swagger at [http://swagger.io](http://swagger.io) or on
    [irc.freenode.net, #swagger](http://swagger.io/irc/). For this sample
    , you can use the api key `special-key` to test the authorization
    filters."
  version: "1.0.0"
  title: "Swagger Petstore"
  termsOfService: "http://swagger.io/terms/"
  contact:
    email: "apiteam@swagger.io"
  license:
    name: "Apache 2.0"
    url: "http://www.apache.org/licenses/LICENSE-2.0.html"
host: "petstore.swagger.io"
basePath: "/v2"
tags:
- name: "pet"
- name: "store"
- name: "user"
schemes:
- "https"
- "http"
```

❑ Base URL

```
1  host: api.example.com
2  basePath: /v1
3  schemes:
4    - https
```

❑ Path

```
1  paths:
2    /users:
3      get:
4        summary: Returns a list of users.
5        description: Optional extended description in Markdown
6        produces:
7          - application/json
8        responses:
9          200:
10           description: OK
```

☐ Parameters

☐ Authentication

```
1  securityDefinitions:  
2    BasicAuth:  
3      type: basic  
4  security:  
5    - BasicAuth: []
```

```
1  paths:  
2    /users/{userId}:  
3      get:  
4        summary: Returns a user by ID.  
5        parameters:  
6          - in: path  
7            name: userId  
8            required: true  
9            type: integer  
10           minimum: 1  
11           description: Parameter description in Markdown.  
12        responses:  
13          200:  
14            description: OK
```

❑ Input and Output Models

```
1  paths:
2    /users/{userId}:
3      get:
4        summary: Returns a user by ID.
5        parameters:
6          - in: path
7            name: userId
8            required: true
9            type: integer
10       responses:
11         200:
12           description: OK
13           schema:
14             $ref: '#/definitions/User'
15     /users:
16       post:
17         summary: Creates a new user.
18         parameters:
19           - in: body
20             name: user
21             schema:
22               $ref: '#/definitions/User'
23       responses:
24         200:
25           description: OK
```



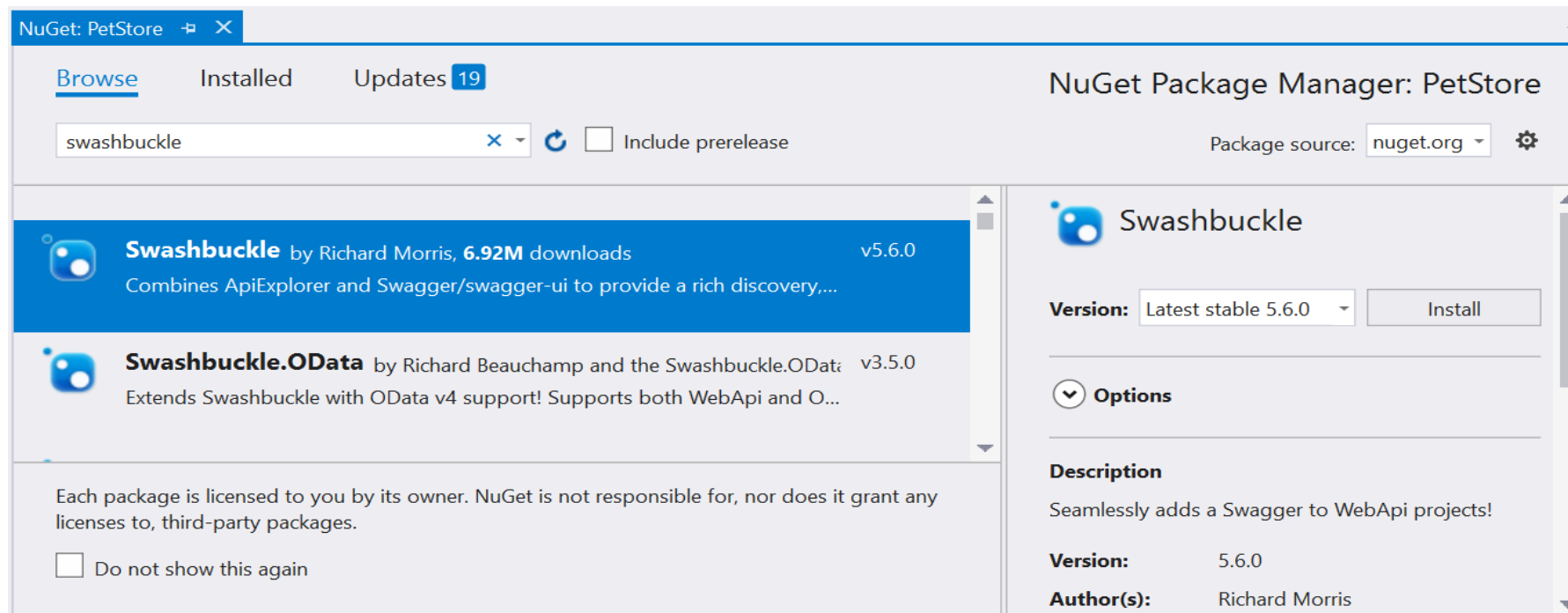


LẬP TRÌNH C# 6

BÀI 1: API DOCUMENT

- ❑ Sử dụng Swashbuckle package
 - ❖ Swashbuckle.AspNetCore.Swagger
 - ❖ Swashbuckle.AspNetCore.SwaggerGen
 - ❖ Swashbuckle.AspNetCore.SwaggerUI

```
Install-Package Swashbuckle.AspNetCore -version 5.0.0-rc4
```



□ Cấu hình Swagger Middleware

```
public void ConfigureServices(IServiceCollection services)
{
    // Register the Swagger generator, defining 1 or more Swagger documents
    services.AddSwaggerGen(c =>
    {
        c.SwaggerDoc("v1", new OpenApiInfo { Title = "My API", Version = "v1" });
    });
}
```

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    // Enable middleware to serve generated Swagger as a JSON endpoint.
    app.UseSwagger();

    // Enable middleware to serve swagger-ui (HTML, JS, CSS, etc.),
    // specifying the Swagger JSON endpoint.
    app.UseSwaggerUI(c =>
    {
        c.SwaggerEndpoint("/swagger/v1/swagger.json", "My API V1");
    });
}
```

❑ Thiết lập Api cung cấp tài liệu cho Swagger

```
public class Employee
{
    3 references
    public int Id { get; set; }
    2 references
    public string FirstName { get; set; }
    2 references
    public string LastName { get; set; }
    [Required]
    2 references
    public string EmailId { get; set; }
}

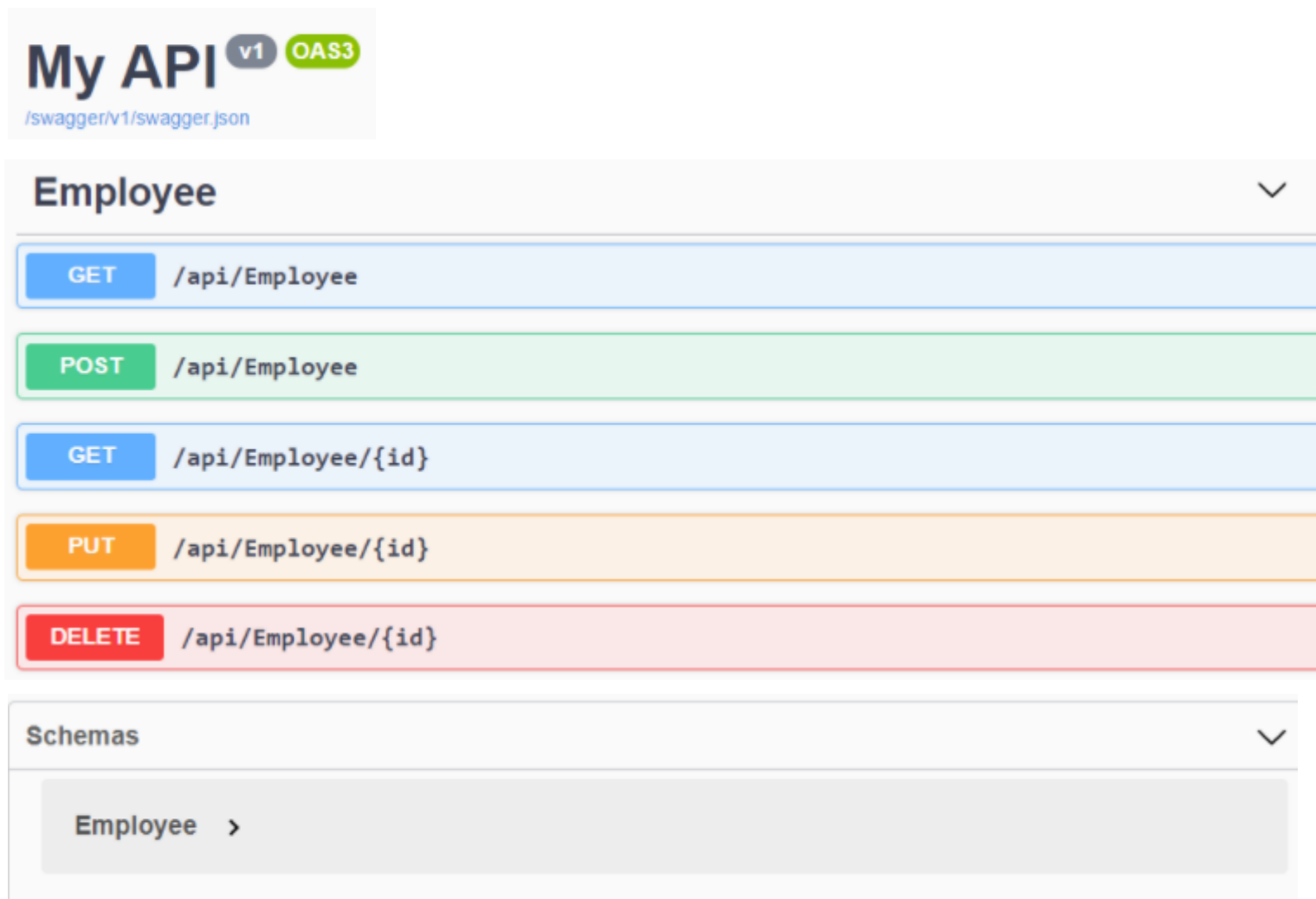
private List<Employee> GetEmployees()
{
    return new List<Employee>()
    {
        new Employee()
        {
            Id = 1,
            FirstName= "Phan Viet",
            LastName = "The",
            EmailId ="thepv@gmail.com"
        },
    }
}
```

```
[Route("api/[controller]")]
[ApiController]
0 references
public class EmployeeController : Controller
{
    /// <summary>
    /// Gets the list of all Employees.
    /// </summary>
    /// <returns>The list of Employees.</returns>
    // GET: api/Employee
    [HttpGet]
    0 references
    public IEnumerable<Employee> Get()
    {
        return GetEmployees();
    }
    // GET: api/Employee/5
    [HttpGet("{id}", Name = "Get")]
    0 references
    public Employee Get(int id)
    {
        return GetEmployees().Find(e => e.Id == id);
    }
}
```

□ <https://localhost:<port>/swagger/v1/swagger.json>

```
{
  "openapi": "3.0.1",
  "info": {
    "title": "My API",
    "version": "v1"
  },
  "paths": {
    "/api/Employee": {
      "get": {
        "tags": [
          "Employee"
        ],
        "responses": {
          "200": {
            "description": "Success",
            "content": {
              "text/plain": {
                "schema": {
                  "type": "array",
                  "items": {
                    "$ref": "#/components/schemas/Emp
"
            }
          }
        },
        "application/json": {
          "schema": {
            "type": "array",
            "items": {
              "$ref": "#/components/schemas/Emp
"
            }
          }
        }
      }
    },
    "/api/Employee/{id}": {
      "get": {
        "tags": [
          "Employee"
        ],
        "operationId": "Get",
        "parameters": [
          {
            "name": "id",
            "in": "path",
            "required": true,
            "schema": {
              "type": "integer",
              "format": "int32"
            }
          }
        ],
        "responses": {
          "200": {
            "description": "Success",
            "content": {
              "text/plain": {
                "schema": {
                  "$ref": "#/components/schemas/Employee"
                }
              },
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/Employee"
                }
              }
            }
          }
        }
      }
    }
  }
}
```

□ Swagger UI



❑ Thực hiện Get

The screenshot shows the Swagger UI interface for a GET endpoint. The top bar displays the HTTP method 'GET' and the path '/api/Employee'. Below this, the 'Parameters' section is visible, indicating 'No parameters' and featuring a 'Cancel' button. The 'Execute' button is highlighted in blue. The 'Responses' section shows a '200' status code. The 'Request URL' is 'https://localhost:5001/api/Employee'. The 'Server response' section displays a JSON array of two employee objects. A 'Download' button is located at the bottom right of the response body.

GET /api/Employee

Parameters Cancel

No parameters

Execute Clear

Responses

Curl

```
curl -X GET "https://localhost:5001/api/Employee" -H "accept: text/plain"
```

Request URL

```
https://localhost:5001/api/Employee
```

Server response

Code	Details
200	<p>Response body</p> <pre>[{ "id": 1, "firstName": "Phan Viet", "lastName": "The", "emailId": "thepv@gmail.com" }, { "id": 2, "firstName": "Osin", "lastName": "Admin", "emailId": "OsinAdmin@gmail.com" }]</pre> Download

□ Bổ sung thông tin mô tả Api

```
//Extending the documentation
services.AddSwaggerGen(c =>
{
    c.SwaggerDoc("v1", new OpenApiInfo
    {
        Title = "Employee API",
        Version = "v1",
        Description = "An API to perform Employee operations",
        TermsOfService = new Uri("https://example.com/terms"),
        Contact = new OpenApiContact
        {
            Name = "Thepv",
            Email = "thepv@gmail.com",
            Url = new Uri("https://caodang.fpt.edu.vn/"),
        },
        License = new OpenApiLicense
        {
            Name = "Employee API LICX",
            Url = new Uri("https://example.com/license"),
        }
    });
});
services.AddControllers();
```

Employee API v1 OAS3

</swagger/v1/swagger.json>

An API to perform Employee operations

[Terms of service](#)

[Thepv - Website](#)

[Send email to Thepv](#)

[Employee API LICX](#)

□ Bổ sung XML Comments

The screenshot shows the Visual Studio Build properties window for the project 'GettingStartedWithSwaggerUI'. The 'Build' tab is selected in the left sidebar. The 'Errors and warnings' section is expanded, showing 'Warning level' set to 4 and 'Suppress warnings' set to '1701;1702;1591'. The 'Treat warnings as errors' section is also expanded, with 'Specific warnings' selected and 'NU1605' entered. The 'Output' section is expanded, showing 'Output path' as an empty text box with a 'Browse...' button. Below this, 'XML documentation file' is checked and set to 'GettingStartedWithSwaggerUI\GettingStartedWithSwaggerUI.xml'. The 'Generate serialization assembly' is set to 'Auto'. A red box highlights the 'XML documentation file' checkbox and its value, with a red circle containing the number '1' next to it. Another red box highlights the 'Suppress warnings' text box, with a red circle containing the number '2' next to it. An 'Advanced...' button is located at the bottom right.

GettingStartedWithSwaggerUI -> X NuGet: GettingStartedWithSwaggerUI Startup.cs NuGet - Solution EmployeeCon

Application
Build
Build Events
Package
Debug
Signing
Code Analysis
TypeScript Build
Resources

Configuration: Active (Debug) Platform: Active (Any CPU)

Errors and warnings

Warning level: 4

2 Suppress warnings: 1701;1702;1591

Treat warnings as errors

☐ None
☐ All
☒ Specific warnings: NU1605

Output

Output path: Browse...

1 ☒ XML documentation file: GettingStartedWithSwaggerUI\GettingStartedWithSwaggerUI.xml

☐ Register for COM interop

Generate serialization assembly: Auto

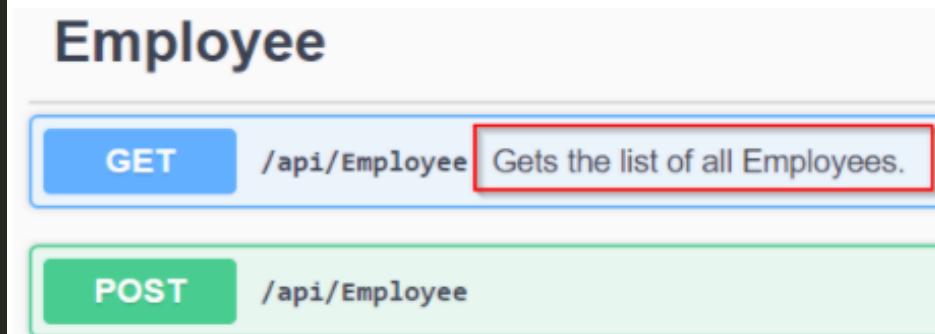
Advanced...

□ Bổ sung XML Comments

```
services.AddSwaggerGen(c =>
{
    c.SwaggerDoc("v1", new OpenApiInfo
    {
        Title = "Employee API",
        Version = "v1",
        Description = "An API to perform Employee operations",
        TermsOfService = new Uri("https://example.com/terms"),
        Contact = new OpenApiContact
        {
            Name = "Thepv",
            Email = "thepv@gmail.com",
            Url = new Uri("https://caodang.fpt.edu.vn/"),
        },
        License = new OpenApiLicense
        {
            Name = "Employee API LICX",
            Url = new Uri("https://example.com/license"),
        }
    });
    // Set the comments path for the Swagger JSON and UI.
    var xmlFile = $"{Assembly.GetExecutingAssembly().GetName().Name}.xml";
    var xmlPath = Path.Combine(AppContext.BaseDirectory, xmlFile);
    c.IncludeXmlComments(xmlPath);
});
```

□ Bổ sung XML Comments

```
/// <summary>
/// Gets the list of all Employees.
/// </summary>
/// <returns>The list of Employees.</returns>
// GET: api/Employee
[HttpGet]
public IEnumerable<Employee> Get()
{
    return GetEmployees();
}
```



□ Thêm mẫu ví dụ sử dụng Api

```

/// </summary>
/// <remarks>
/// Sample request:
///
///     POST api/Employee
///     {
///         "firstName": "Fpoly",
///         "lastName": "thepv",
///         "emailId": "thepv@fpt.com"
///     }
/// </remarks>
/// <param name="employee"></param>
/// <returns>A newly created employee</returns>
/// <response code="201">Returns the newly created item</response>
/// <response code="400">If the item is null</response>
[HttpPost]
[ProducesResponseType(201)]
[ProducesResponseType(400)]
[Produces("application/json")]
0 references
public Employee Post([FromBody] Employee employee)
{
    // Logic to create new Employee
    return new Employee();
}

```

Employee

GET /api/Employee Gets the list of all Employees.

POST /api/Employee Creates an Employee.

Sample request:

```

POST api/Employee
{
  "firstName": "Fpoly",
  "lastName": "thepv",
  "emailId": "thepv@fpt.com"
}

```

❑ Sử dụng Data Annotations

```
public class Employee
{
    public int Id { get; set; }

    public string FirstName { get; set; }

    public string LastName { get; set; }

    [Required]
    public string EmailId { get; set; }
}
```

Schemas

```
Employee ▾ {
  id                integer($int32)
  firstName         string
                  nullable: true
  lastName          string
                  nullable: true
  emailId*          string
                  nullable: true
}
```

Demo 1.3



Tổng kết bài học

- ◎ Tài liệu hướng dẫn sử dụng API
- ◎ Công cụ Swagger
- ◎ Swagger UI và ASP.NET Core Web API





KẾT THÚC