API Session – Chapter 13

- Swagger, Curl and Documentation -

Swagger is a suite of tools and specifications for designing, building, documenting, and consuming RESTful APIs. It provides a standardized way to describe the structure and behavior of an API using a format called the OpenAPI Specification (OAS). Swagger helps both developers and consumers understand and interact with APIs more efficiently. Here’s an overview of its key components:

# **1. OpenAPI Specification (OAS):**

* **Definition**: A standard format for describing RESTful APIs. Originally known as Swagger Specification, it has been renamed to OpenAPI Specification and is maintained by the OpenAPI Initiative.
* **Format**: Typically written in YAML or JSON, it provides details about endpoints, request/response formats, parameters, authentication methods, and more.

# **2. Swagger UI:**

* **Purpose**: A web-based user interface that automatically generates interactive documentation from the OpenAPI Specification.
* **Features**: Allows users to visualize and interact with API endpoints, send requests, and view responses directly from the browser.

# **3. Swagger Editor:**

* **Purpose**: An online or desktop tool for creating and editing OpenAPI Specification documents.
* **Features**: Provides a user-friendly interface with real-time validation, auto-completion, and syntax highlighting to simplify API design.

# **4. Swagger Codegen:**

* **Purpose**: A tool that generates client libraries, server stubs, API documentation, and other artifacts from an OpenAPI Specification.
* **Features**: Supports various programming languages and frameworks, helping to accelerate development and ensure consistency.

# **5. Swagger Hub:**

* **Purpose**: A collaborative platform for designing, building, and documenting APIs.
* **Features**: Offers integrated tools for API design, collaboration, versioning, and hosting API documentation.

# **Benefits of Using Swagger:**

* **Standardization**: Provides a consistent way to describe API functionality, which helps ensure clarity and reduces ambiguity.
* **Automation**: Facilitates automated generation of documentation, client SDKs, and server stubs, speeding up development and integration processes.
* **Interactive Documentation**: Allows users to explore and test API endpoints in a user-friendly environment.
* **Collaboration**: Enhances collaboration between API developers and consumers by providing a clear and interactive API design.

# **How It Works:**

1. **Design**: Define your API using the OpenAPI Specification.
2. **Document**: Generate interactive API documentation using Swagger UI.
3. **Generate**: Use Swagger Codegen to create client libraries and server stubs.
4. **Collaborate**: Use Swagger Hub to manage and collaborate on API design and documentation.

Swagger’s tools and specifications have become integral to modern API development and documentation, helping teams create robust, well-documented APIs with greater efficiency and consistency.

# Sample

<https://fakerestapi.azurewebsites.net/index.html>

# CURL

**cURL** (pronounced "curl") is a command-line tool used to transfer data using various network protocols, most notably HTTP. It's a versatile tool that can perform a wide range of tasks related to network communication, such as:

* **Downloading files:** You can use cURL to download files from the internet, specifying the URL of the file you want to download.
* **Uploading files:** You can upload files to servers using cURL.
* **Sending HTTP requests:** cURL can be used to send various types of HTTP requests, such as GET, POST, PUT, DELETE, etc.
* **Executing custom headers:** You can add custom headers to your requests, which can be useful for authentication, content negotiation, and other purposes.
* **Performing authentication:** cURL supports various authentication mechanisms, including basic authentication, digest authentication, and OAuth.
* **Interacting with APIs:** You can use cURL to interact with RESTful APIs, making requests and parsing responses.

**Basic usage:**

Bash

curl [URL]

For example, to download a file from a specific URL:

**Bash**

curl https://example.com/file.zip -o downloaded\_file.zip

**Key features and capabilities:**

* **Protocol support:** cURL supports a wide range of protocols, including HTTP, HTTPS, FTP, FTPS, SCP, SFTP, TELNET, and more.
* **Customizable options:** cURL offers a wide range of options and flags to customize its behavior, allowing you to perform complex network operations.
* **Scripting:** cURL can be used in scripts and automation tools to automate network tasks.
* **Debugging:** cURL provides various options for debugging network requests, such as displaying headers, response codes, and detailed information about errors.

In summary, cURL is a powerful and flexible command-line tool that can be used for a variety of network-related tasks. It's a valuable tool for developers, system administrators, and anyone who needs to interact with network resources from the command line.

# Create Request From CURL

Importing a cURL command into Postman is a straightforward process. Here’s how you can do it step-by-step:

1. **Open Postman**: Launch the Postman application on your computer.
2. **Open the Import Dialog**:

* Click on the “Import” button, usually found in the top left corner of the Postman window.

1. **Choose the cURL Option**:

* In the Import dialog, you will see several options for importing. Select the “Raw Text” tab.

1. **Paste the cURL Command**:

* Copy your cURL command from your source.
* Paste it into the text area provided in the “Raw Text” tab.

1. **Click on Import**:

* After pasting your cURL command, click the “Continue” or “Import” button. Postman will process the command and convert it into a request.

1. **Review and Save**:

* Postman will create a new request based on the cURL command you provided. Review the request details to ensure everything looks correct.
* You can then save this request to a collection by clicking on the “Save” button.

1. **Run and Test**:

* You can now run the request by clicking the “Send” button and see the response from the server.

# Postman Documentation

Creating documentation for an API collection in Postman is a straightforward process that helps you share and maintain clear, comprehensive details about your APIs. Here’s how to create and publish documentation for an API collection in Postman:

## **1. Create a Collection**

**Open Postman**.

**Create a New Collection**:

* Click on the New button in the top-left corner.
* Select Collection.
* Enter a name and description for your collection.
* Click Create to save the collection.

**Add Requests to the Collection**:

* Add requests by creating new ones and saving them to your collection.
* Organize requests into folders within the collection if needed.

## **2. Add Descriptions and Details**

1. **Add Descriptions to Requests**:

* Open a request within your collection.
* Add a description in the Description tab below the request name.
* Include details about the request, parameters, response, and any other relevant information.

1. **Add Descriptions to Collection and Folders**:

* Click on the collection name in the sidebar to open the collection details.
* Add a description for the collection in the Description tab.
* Click on any folder within the collection and add descriptions to organize requests better.

## **3. Generate Documentation**

1. **Open the Collection**:

* Click on the collection name in the sidebar.

1. **Access Documentation Settings**:

* Click on the View in Web button or the View Documentation button (depending on your Postman version) in the collection details.
* Alternatively, click on the three dots (...) next to the collection name and select Publish Docs if you want to create and share public documentation.

1. **Customize Documentation**:

* **Title**: Enter a title for your documentation.
* **Description**: Add an overview and detailed explanations about the collection and its purpose.
* **Version**: (Optional) Specify the version of your API or collection.

1. **Publish Documentation**:

* Once you’re satisfied with the documentation, click on the Publish button to make it available.
* Choose the visibility settings (Public, Team, or Private) depending on who should have access.

## **4. Share Documentation**

1. **Get a Shareable Link**:

* After publishing, you can obtain a shareable link to your documentation from Postman’s documentation settings.
* Share this link with your team or external stakeholders.

1. **Embed Documentation**:

* Postman also allows you to embed the documentation into other platforms or websites using provided embed codes.

## **5. Update and Maintain Documentation**

1. **Edit Documentation**:

* You can edit and update documentation at any time by accessing the documentation settings of your collection.
* Update descriptions, request details, or other information as needed.

1. **Version Control**:

* If you have multiple versions of an API, consider creating separate collections for each version and document them accordingly.