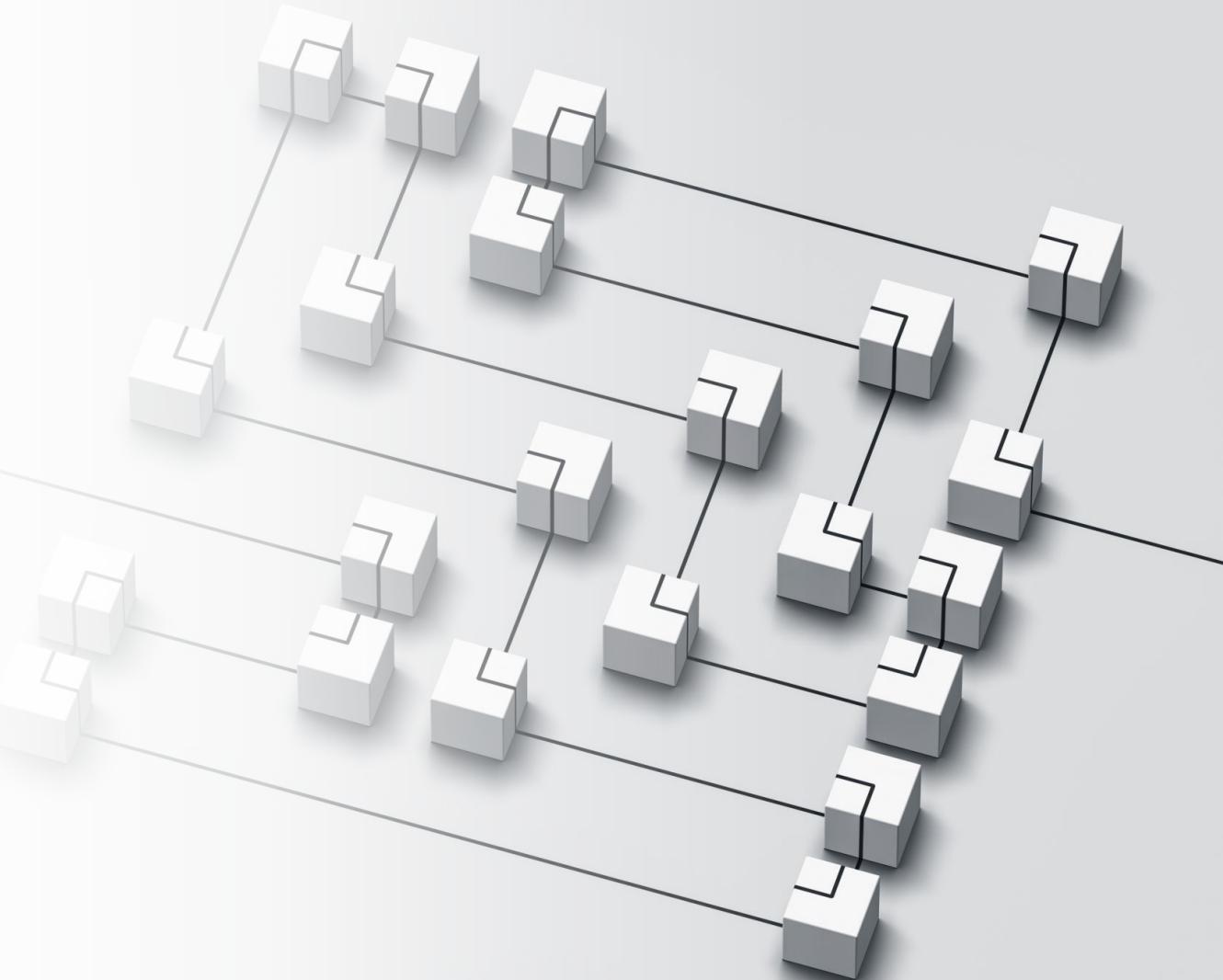


API Design

Software Engineering 2

What is API?

- API stands for Application Programming Interface
- An API is the interface that a software program presents to other program, to humans.



Why Do We Need APIs?

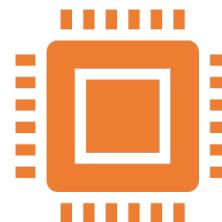


APIs have emerged out of a need to exchange information.

The data provider might be equipped to solve specific problems.

So, folks at other companies don't have to spend time solving those problems themselves.

For example, Google map, Log-in system, chatbot.



APIs enable businesses to develop unique products quickly.

Startups can differentiate their product offerings by taking advantage of existing technologies and tapping into other ecosystems.

API's history

In the '60s, developers began to build standard libraries and share these with other developers.

- These developers could use the standard functionality of these libraries without knowing their internal code.

70s

60s

During the '90s,

In the '70s and '80s, with the emergence of network-connected computers, came the first network APIs that exposed services developers could consume through Remote Procedure Calls (RPCs).

With the emergence of the internet, many companies wanted to standardize the way we build and expose APIs.

- Standards such as the Common Object Request Broker Architecture (CORBA), the Component Object Model (COM) and Distributed Component Object Model (DCOM) by Microsoft.
- Many others sought to become the de facto way to expose services over the web.

API's history (cont.)

Came more open and standard ways of accessing remote services over the web (web APIs).

- Simple Object Access Protocol (SOAP)
- Extensible Markup Language (XML)
- Representative State Transfer (REST)
- JavaScript Object Notation (JSON)

In the late '90s and early '00s

At present

Every tech company began exposing useful services through APIs—

- Amazon Affiliate API (2002)
- Flickr API (2004)
- Google Maps API (2005)
- Yahoo! Pipes API (2007)

There are now thousands of APIs exposing every service imaginable, from image manipulation to artificial intelligence.

Web API Example

Google Maps

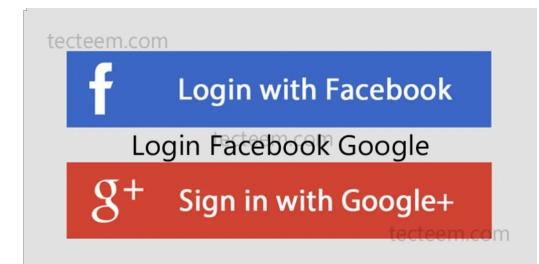
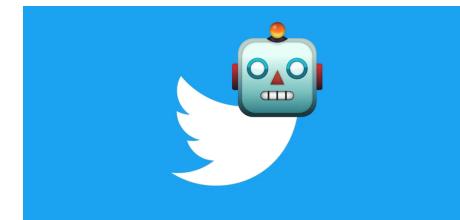
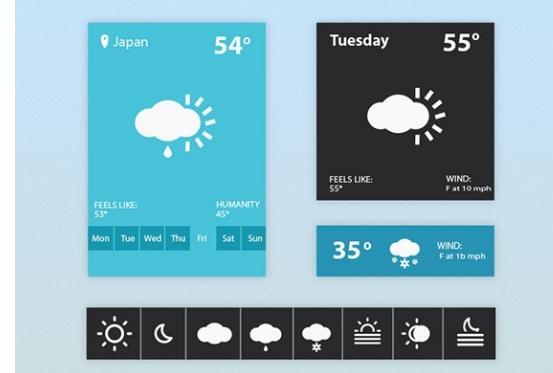
Pay with Paypal

Weather Snippets

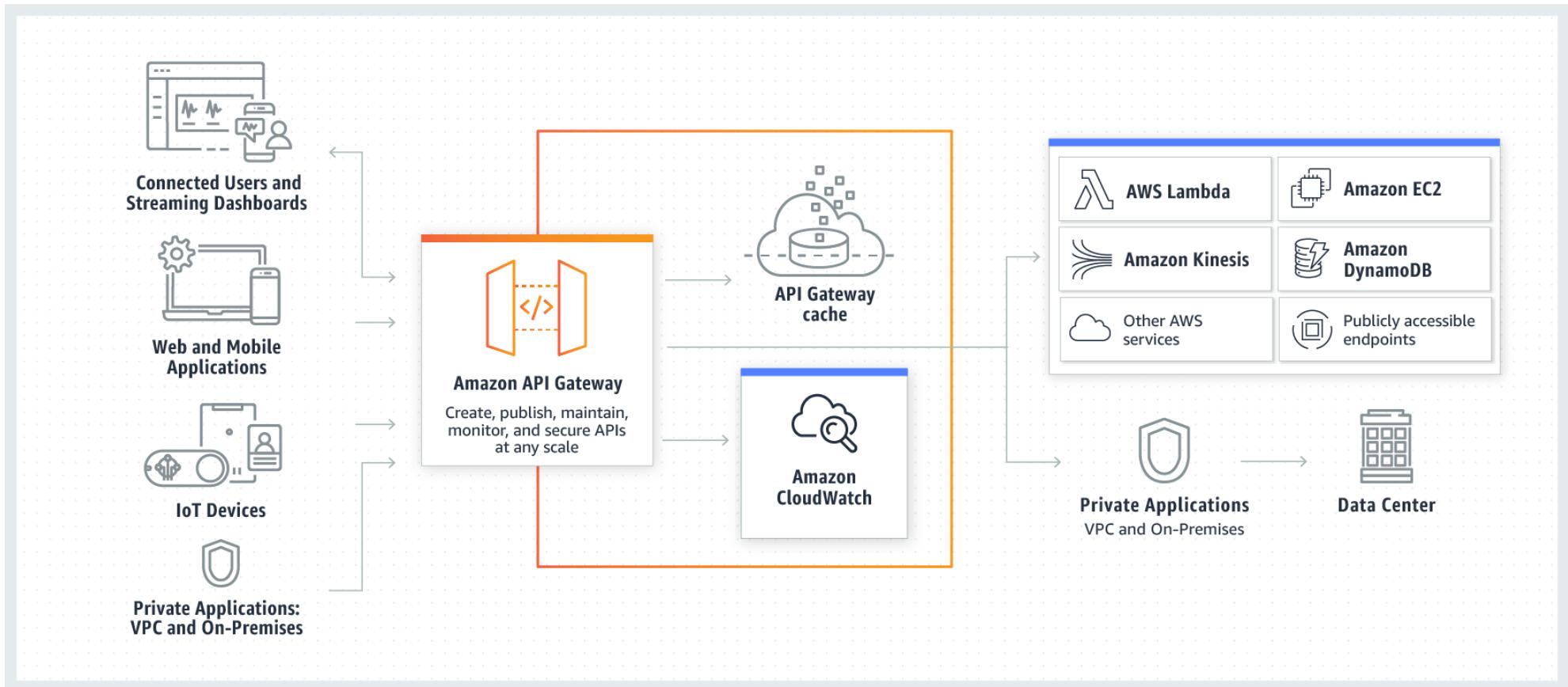
Twitter Bots

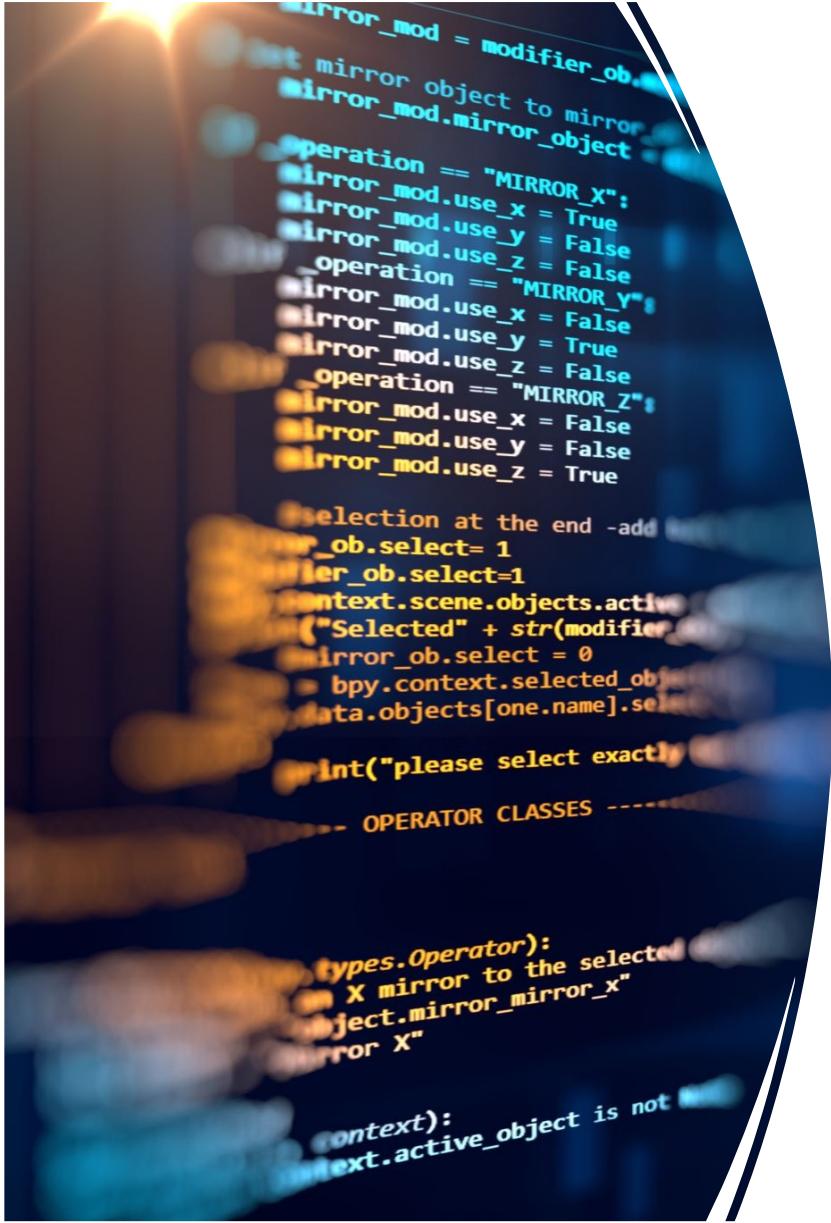
Log-In Using Facebook or Google

Travel Booking



Amazon API Gateway



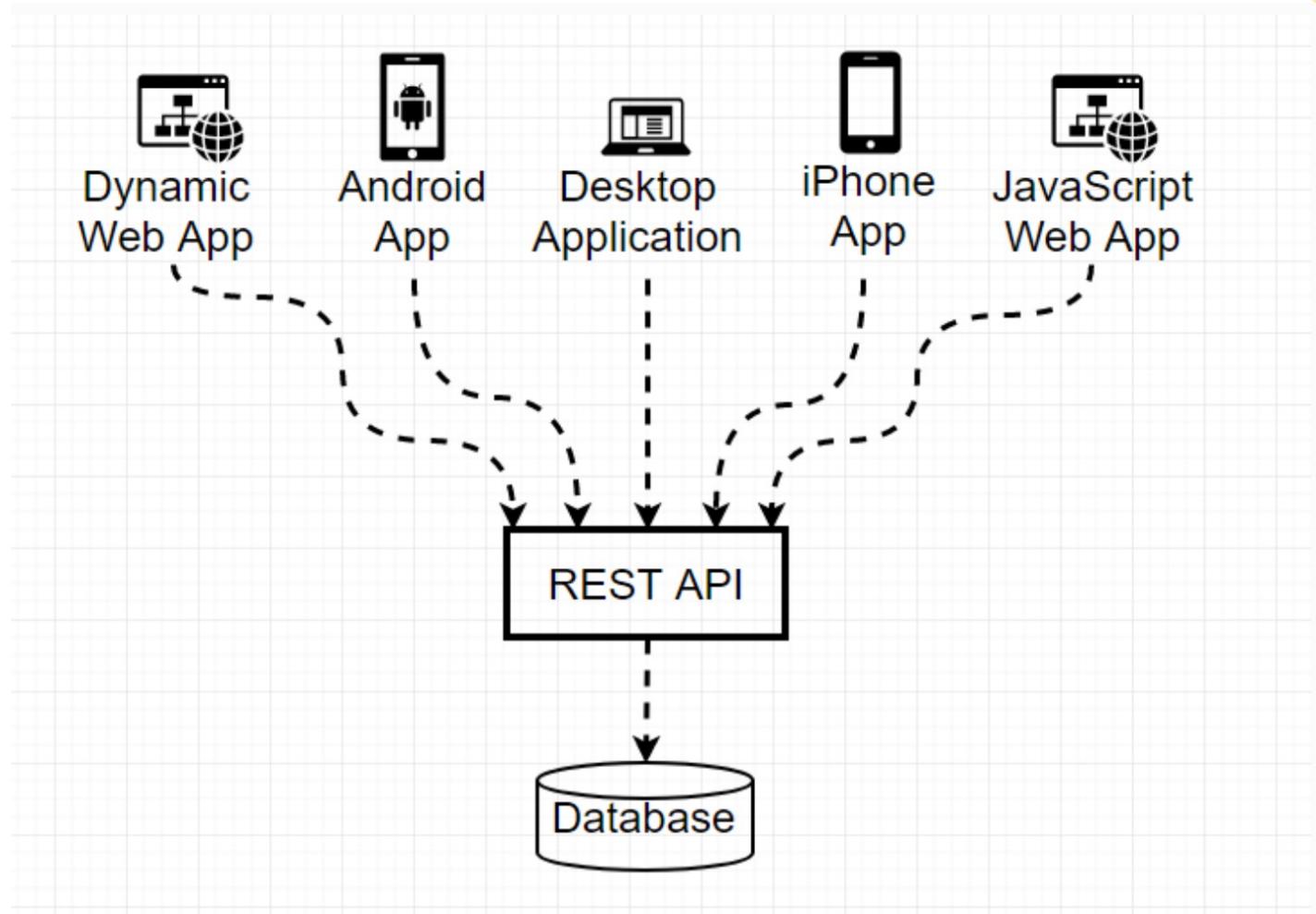


API Paradigms

- An API paradigm defines the interface exposing backend data of a service to other applications.
- Request–Response APIs
 - Representational State Transfer (REST)
 - Remote Procedure Call (RPC)
 - GraphQL
- Event-Driven APIs
 - WebHooks
 - WebSockets
 - HTTP Streaming

REST API

- **Representational State Transfer**
- When a client request -> the server transfers a representation of the state of the resource to the requester or endpoint.
- This information, or representation, is delivered in one of several formats via HTTP:
 - JSON (Javascript Object Notation)
 - HTML
 - XLT
 - Python
 - PHP
 - plain text



CRUD operations, HTTP verbs, and REST conventions

Operation	HTTP verb	URL: /users	URL: /users/U123
Create	POST	Create a new user	Not applicable
Read	GET	List all users	Retrieve user U123
Update	PUT or PATCH	Batch update users	Update user U123
Delete	DELETE	Delete all users	Delete user U123

HTTP Request examples

Example 2-1. HTTP request to retrieve a charge from the Stripe API

```
GET /v1/charges/ch_CWyutlXs9pZyfD  
HOST api.stripe.com  
Authorization: Bearer YNoJ1Yq64iCBhzfL9HN000fzVrsEjtVl
```

Example 2-2. HTTP request to create a charge from the Stripe API

```
POST /v1/charges/ch_CWyutlXs9pZyfD  
HOST api.stripe.com  
Content-Type: application/x-www-form-urlencoded  
Authorization: Bearer YNoJ1Yq64iCBhzfL9HN000fzVrsEjtVl  
  
amount=2000&currency=usd
```

Showing relationships

- A resource can be better represented as a sub-resource instead of a top-level resource in the URL.
- This makes the relationship clear for the developers using the API.
- For instance, the GitHub API uses sub-resources to represent relationships in various APIs:

`POST /repos/:owner/:repo/issues`

Create an issue.

`GET /repos/:owner/:repo/issues/:number`

Retrieve an issue.

`GET /repos/:owner/:repo/issues`

List all issues.

`PATCH /repos/:owner/:repo/issues/:number`

Edit an issue.

Welcome to the Rapid API Hub

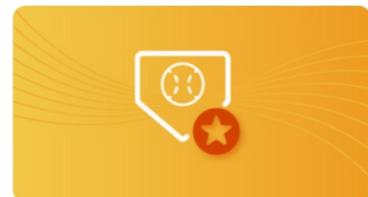
Discover and connect to thousands of APIs

Categories

- [Sports](#)
- [Finance](#)
- [Data](#)
- [Entertainment](#)
- [Travel](#)
- [Location](#)
- [Science](#)
- [Food](#)
- [Transportation](#)
- [Music](#)
- [Business](#)

Discover More APIs

Browse through our collections to learn about new use cases to implement in your app

[Top Baseball APIs](#)[Top Translation APIs](#)[Top Movie APIs](#)[Top Weather APIs](#)

Recommended APIs

[View All](#)

APIs curated by RapidAPI and recommended based on functionality offered, performance, and support!

Let's try an API.

RapidAPI

Search for APIs

Create Team Add Your API Docs Log In Sign Up

Free Public APIs for Developers

[View All](#)

If you're new to RapidAPI, this collection is a great place to start exploring APIs that are free to test, specifically updated for 2021.

 Google Translate
Dynamically translate between languages.

9.9 ⚡ 607 ms ✓ 100%



 Movie Database Alternative
RESTful web service to access information, pictures, and more from the movie database. Get Title, Plot, Cast, Crew, and more.

9.9 ⚡ 1,622 ms ✓ 100%

 API-FOOTBALL
+1 050 football leagues & cups. Livescore (15s), live & pre-match odds, events, line-ups, coaches, players, fixtures, results, stats, and more.
Verified ✓

10 ⚡ 339 ms ✓ 100%

Top AI Based APIs

[View All](#)

Artificial Intelligence (AI) based APIs are a way for developers to access the functionality of pre-trained AI models without the need for extensive knowledge of machine learning algorithms. These APIs can be integrated into a variety of applications and are typically categorized based on their functionality.

 Cortex
The Biggest AI Dataset

6.2 ⚡ 30,168 ms ✓ 100%

 Cloudlabs Text To Speech
Convert text to audio quickly, supports over 100 languages and 300+ speakers

9.6 ⚡ 923 ms ✓ 98%

 Text To Speech Neural/Google
Get audio from text easily, choose from our neural or google engine and many options for different voices

9.4 ⚡ 616 ms ✓ 99%

 Aeona
Api to the best AI chatbot around!
<https://huggingface.co/deep>

9.3 ⚡ 626 ms ✓ 100%

Tax APIs

[View All](#)

List of Tax and Tax calculator APIs



Google Translate

FREEMIUM

By Google Cloud | Updated 2 years ago | Text Analysis

Popularity

9.9 / 10

Latency

607ms

Service Level

100%

+ Health Check

N/A

Endpoints About Tutorials Discussions Pricing

Dynamically translate between languages. [Show more...](#)

V1 (Current) ▾

Search endpoints

GET languages

POST detect

POST translate

POST translate

Test Endpoint

Code Snippets

Results

(Node.js Axios) ▾ Copy Code

```
const axios = require('axios');
```

```
const encodedParams = new URLSearchParams();
encodedParams.set('q', 'cat');
encodedParams.set('target', 'th');
encodedParams.set('source', 'en');
```

```
const options = {
  method: 'POST',
  url: 'https://google-translate1.p.rapidapi.com/language/translate/v2',
  headers: {
    'content-type': 'application/x-www-form-urlencoded',
    'Accept-Encoding': 'application/gzip',
    'X-RapidAPI-Key': '2759db347bmsh956292195b2d850p169f82jsn14fb90afb',
    'X-RapidAPI-Host': 'google-translate1.p.rapidapi.com'
  },
  data: encodedParams,
};

try {
```

try {

Home > Google Translate1

Language: English ▾



Google Translate FREEMIUM

By Google Cloud | Updated 2 years ago | Text Analysis

Popularity

9.9 / 10

Latency

607ms

Service Level

100%

Health Check

N/A

Endpoints About Tutorials Discussions Pricing

Dynamically translate between languages. [Show more...](#)

V1 (Current) ▾

Search endpoints

POST translate

Test Endpoint

Code Snippets

Results

GET languages

POST detect

POST translate

Required Parameters

q

STRING

cat

REQUIRED The input text to translate. Repeat this parameter to perform translation operations on multiple text inputs.

target

STRING

th

REQUIRED The language to use for translation of the input text, set to one of the language codes listed in the overview tab

Optional Parameters

format

STRING

OPTIONAL The format of the source text, in either HTML (default) or plain-text. A value of html indicates HTML and a value of

200 Success

Body Headers

- ▼ {} 1 key
- ▼ data: {} 1 key
- ▼ translations: [] 1 item
- ▼ 0: {} 1 key
- translatedText: "ແມ່ນ"

+ Expand All Copy

Home > Google Translate1

Language: English ▾

Let's code our own APIs.

1. Set up NodeJS – <https://nodejs.org/>

The screenshot shows the official Node.js website. At the top, there is a navigation bar with links for LEARN, ABOUT, DOWNLOAD, GUIDES, BLOG, DOCS, and CERTIFICATION. On the far right of the bar are icons for a moon (dark mode), a user profile, and a search icon. Below the navigation bar, a large heading reads "Node.js® is an open-source, cross-platform JavaScript runtime environment." Underneath this heading are two prominent green buttons. The left button is labeled "20.11.0 LTS" and "Recommended For Most Users". The right button is labeled "21.6.1 Current" and "Latest Features". Below each button are links to "Other Downloads", "Changelog", and "API Docs". A note below the buttons says, "For information about supported releases, see the [release schedule](#)." At the bottom of the page, there is a dark footer bar containing legal text: "Copyright OpenJS Foundation and Node.js contributors. All rights reserved. The OpenJS Foundation has registered trademarks and uses trademarks. For a list of trademarks of the OpenJS Foundation, please see our [Trademark Policy](#) and [Trademark List](#). Trademarks and logos not indicated on the [list of OpenJS Foundation trademarks](#) are trademarks™ or registered® trademarks of their respective holders. Use of them does not imply any affiliation with or endorsement by them."

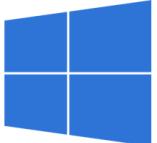
2. Set up Visual Studio Code - <https://code.visualstudio.com/Download>

Visual Studio Code Docs Updates Blog API Extensions FAQ Learn Search Docs [Download](#) [X](#)

[Version 1.85](#) is now available! Read about the new features and fixes from November.

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

[↓ Windows](#)
Windows 10, 11

[↓ .deb](#)
Debian, Ubuntu

[↓ .rpm](#)
Red Hat, Fedora, SUSE

[↓ Mac](#)
macOS 10.15+

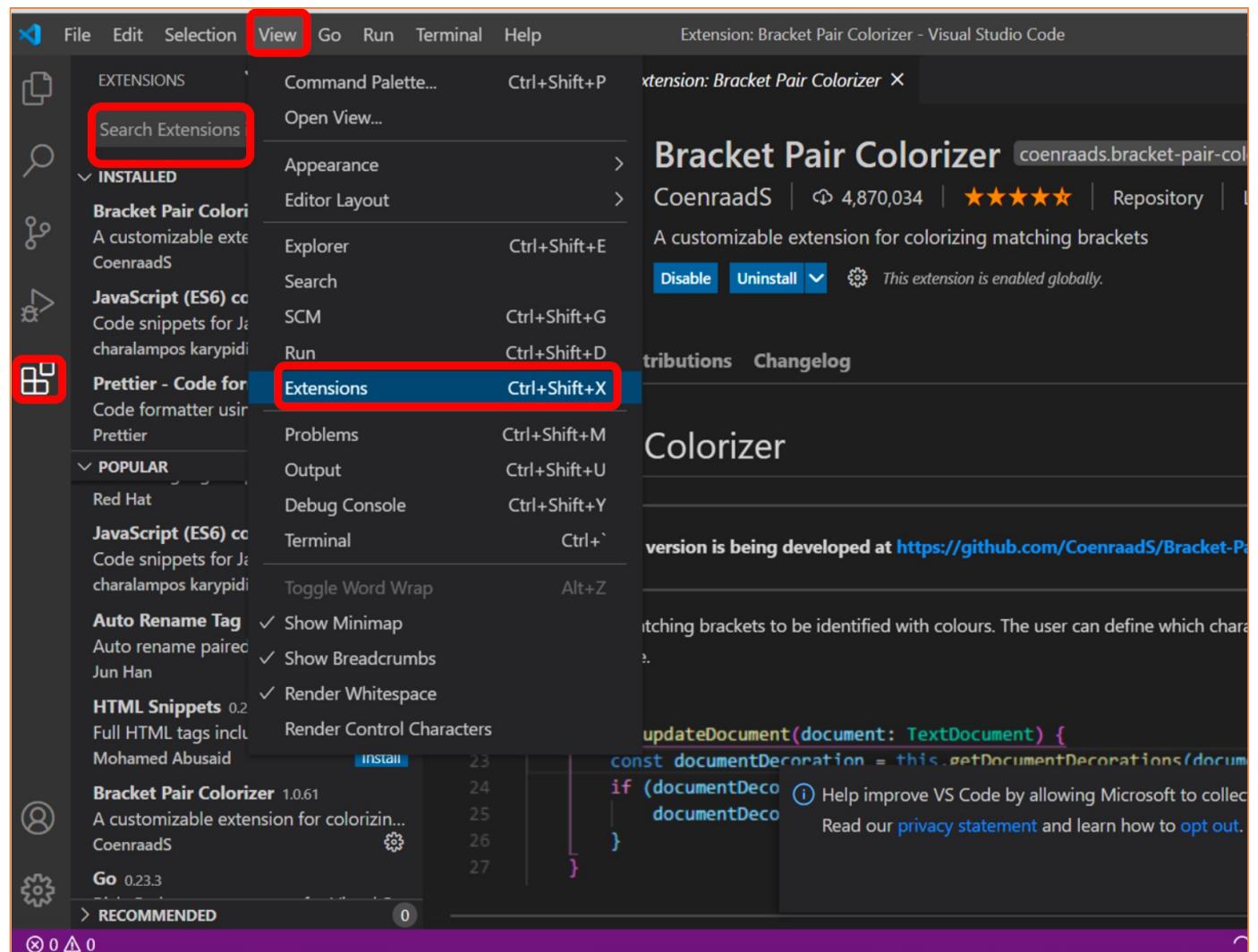
User Installer [x64](#) [Arm64](#)
System Installer [x64](#) [Arm64](#)
.zip [x64](#) [Arm64](#)
CLI [x64](#) [Arm64](#)

.deb [x64](#) [Arm32](#) [Arm64](#)
.rpm [x64](#) [Arm32](#) [Arm64](#)
.tar.gz [x64](#) [Arm32](#) [Arm64](#)
Snap [Snap Store](#)
CLI [x64](#) [Arm32](#) [Arm64](#)

.zip [Intel chip](#) [Apple silicon](#) [Universal](#)
CLI [Intel chip](#) [Apple silicon](#)

3. Install extension (recommended)

- 1. Bracket Pair Colorizer
- 2. DotENV
- 3. ExpressSnippet
- 4. JavaScript (ES6) code snippets
- 5. Prettier - Code formatter
- 6. REST Client



4. Let's start our project!

1. Create a folder : “MyCat”
2. Open VS Code
3. Open your folder “MyCat” in VS Code
 - Tab File > Open folder > Browser “MyCat” folder
4. Open Terminal
 - Tab: Terminal > New Terminal
5. Create new project
 - \$ npm init //entry point: server.js
 - \$ npm i express dotenv
 - \$ npm i -D nodemon

The screenshot shows the VS Code interface with the following details:

- EXPLORER**: Shows a tree view with a red box around the "package.json" entry under "OPEN EDITORS".
- OPEN EDITORS**: Shows multiple "package.json" files, with one specifically highlighted by a red box.
- MYCAT**: A folder containing "node_modules" and "package-lock.json".
- EDITOR**: The main code editor window titled "package.json – MyCat" contains the following JSON code:

```
1  { "name": "mycat",  
2   "version": "1.0.0",  
3   "description": "",  
4   "main": "server.js",  
5   "scripts": {  
6     "test": "echo \\\"Error: no test specified\\\" & exit 1",  
7     "dev" : "nodemon server.js"  
8   },  
9   "author": "",  
10  "license": "ISC",  
11  "dependencies": {  
12    "dotenv": "^16.0.0",  
13    "express": "^4.17.2"  
14  },  
15  "devDependencies": {  
16    "nodemon": "^2.0.15"  
17  }  
18 }  
19 }  
20 }
```
- TERMINAL**: Shows the command-line output of an npm install operation:

```
run `npm fund` for details  
found 0 vulnerabilities  
Pittipols-MacBook-Pro:MyCat Pittipol$ npm i -D nodemon  
added 117 packages, and audited 169 packages in 4s  
18 packages are looking for funding  
  run `npm fund` for details  
found 0 vulnerabilities
```
- STATUS BAR**: Shows "Ln 20, Col 1" and other file metadata.

เพิ่มบรรทัดนี้ใน
"package.json"

"dev" : "nodemon server.js"

The screenshot shows the VS Code interface with the following details:

- EXPLORER**: Shows the project structure with a folder named "MYCAT". Inside "MYCAT", there is a "config" folder which contains a file named "config.env". Both the "config" folder and the "config.env" file are highlighted with red boxes.
- OPEN EDITORS**: Shows two open files: "package.json" and "config.env". The "config.env" file is currently active and displayed in the editor area.
- config.env Content**: The file contains the following code:

```
config > config.env
1  PORT = 5050
2  NODE_ENV = development
```
- TERMINAL**: Shows the output of an npm install command:

```
run `npm fund` for details
found 0 vulnerabilities
Pittipols-MacBook-Pro:MyCat Pittipol$ npm i -D nodemon
added 117 packages, and audited 169 packages in 4s
18 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
```
- Bottom Status Bar**: Shows the current file is "config.env", line 1, column 12, with 4 spaces, using UTF-8 encoding, and includes links for Environment Variables, Prettier, and a bell icon.

1. สร้าง folder "config"
2. สร้างไฟล์ "config/config.env"
3. ใส่ code

PORT = 5050 // ตามกำหนด
NODE_ENV = development

A screenshot of the Visual Studio Code (VS Code) interface. The title bar shows "server.js — MyCat". The left sidebar (Explorer) lists files: package.json, config.env, server.js (selected), config, config.env, node_modules, package-lock.json, package.json, and server.js (another entry). The main editor area displays the code for "server.js". The code defines an Express app with routes for GET, POST, PUT, and DELETE requests to '/api/v1/cats/:id'. It uses dotenv for environment variables and logs the server's port. A red box highlights the editor area. The status bar at the bottom shows "Ln 26, Col 56" and other settings.

```
1 const express = require('express');
2 const dotenv = require('dotenv');
3
4 //Load env vars
5 dotenv.config({path:'./config/config.env'});
6
7 const app=express();
8
9 app.get('/api/v1/cats', (req,res) => {
10   res.status(200).json({success:true, msg:'Show all cats'});
11 });
12
13 app.get('/api/v1/cats/:id', (req,res) => {
14   res.status(200).json({success:true, msg:'Show cat ${req.params.id}'});
15 });
16
17 app.post('/api/v1/cats', (req,res) => {
18   res.status(200).json({success:true, msg:'Create new cat'});
19 });
20
21 app.put('/api/v1/cats/:id', (req,res) => {
22   res.status(200).json({success:true, msg:'Update cat ${req.params.id}'});
23 });
24
25 app.delete('/api/v1/cats/:id', (req,res) => {
26   res.status(200).json({success:true, msg:'Delete cat ${req.params.id}'});
27 });
28
29 const PORT=process.env.PORT || 5050;
30 app.listen(PORT, console.log('Server running in ', process.env.NODE_ENV, 'mode on port ', PORT));
```

```
const express = require('express');
const dotenv = require('dotenv');

//Load env vars
dotenv.config({path:'./config/config.env'});

const app=express();

app.get('/api/v1/cats', (req,res) => {
res.status(200).json({success:true, msg:'Show all cats'});
});

app.get('/api/v1/cats/:id', (req,res) => {
res.status(200).json({success:true, msg:'Show cat ${req.params.id}'});
});

app.post('/api/v1/cats', (req,res) => {
res.status(200).json({success:true, msg:'Create new cat'});
});

app.put('/api/v1/cats/:id', (req,res) => {
res.status(200).json({success:true, msg:'Update cat ${req.params.id}'});
});

app.delete('/api/v1/cats/:id', (req,res) => {
res.status(200).json({success:true, msg:'Delete cat ${req.params.id}'});
});

const PORT=process.env.PORT || 5050;
app.listen(PORT, console.log('Server running in ', process.env.NODE_ENV, 'mode on port ', PORT));
```

- 
1. สร้างไฟล์ `server.js`
 2. Copy -> Paste
 3. Save

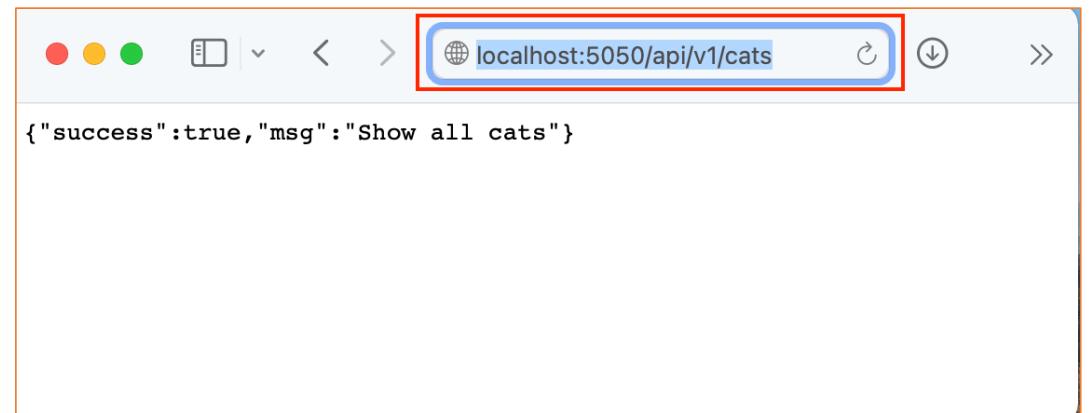
5.Run server / Test

- ที่ Terminal สรัน \$ npm run dev

```
Pittipols-MacBook-Pro:MyCat Pittipo$ npm run dev
> mycat@1.0.0 dev
> nodemon server.js

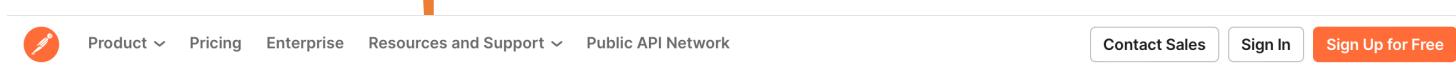
[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js'
Server running in development mode on port 5050
```

- เปิด browser แล้วรัน
<http://localhost:5050/api/v1/cats>



การทดสอบ API ด้วย Postman

- <https://www.postman.com/downloads/>



The Postman app

Download the app to get started with the Postman API Platform.

Mac Intel Chip

Mac Apple Chip

By downloading and using Postman, I agree to the [Privacy Policy](#) and [Terms](#).

[Release Notes](#)

Not your OS? Download for Windows ([x64](#)) or Linux ([x64, arm64](#))

A screenshot of the Postman application interface. The interface has a header with "Home", "Workspaces", "API Network", and "Explore" buttons, along with a search bar. The main workspace shows a "Notion's Public Workspace" with a "Databases" collection. A specific API endpoint, "GET Retrieve a database", is selected. The details pane shows the method as "GET", the URL as "https://api.notion.com/v1/databases/:id", and various configuration tabs like Params, Auth, Headers, Body, Pre-req., Tests, and Setting. Below the URL, there is a table for "Query params" with columns for KEY, VALUE, and DESC. The table has one row with "Key" and "Value".



Home Workspaces API Network Reports Explore

Search Postman

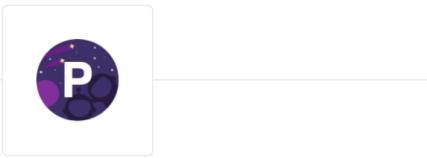


Upgrade



Hey there, night owl!

Pick up where you left off, catch up with your team's work.

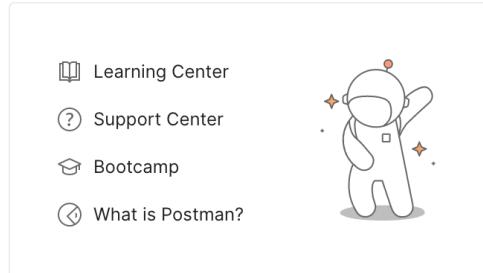


Pittipol

pittipol.postman.co



- Workspaces >
- Private API Network >
- Integrations >



Learning Center

Support Center

Bootcamp

What is Postman?

Recently visited workspaces

My Workspace



Get started with Postman

Start with something new

Create a new request, collection, or API in a workspace

[Create New →](#)

Import an existing file

Import any API schema file from your local drive or Github

[Import file →](#)

Explore our public network

Browse featured APIs, collections, and workspaces published by the Postman community.

[Explore →](#)

Work smarter with Postman

Learn how Postman can help you at every stage of the API development.

[Learn →](#)

Announcements •

Show

Activity Feed

January 26, 2022

Pittipol Kantavat created Team Workspace workspace
12:54 PM

Postman

File Edit View Help

Home Workspaces Reports Explore Search Postman Invite Upgrade

My Workspace New Import + ⚙️ No Environment X

Collections DevCamper API

APIs Environments Mock Servers Monitors History

Create New

Building Blocks

HTTP Request Create a basic HTTP request

WebSocket Request BETA Test and debug your WebSocket connections

Collection Save your requests in a collection for reuse and sharing

Environment Save values you frequently use in an environment

Workspace Create a workspace to build independently or in collaboration

Advanced

API Documentation Create and publish beautiful documentation for your APIs

Mock Server Create a mock server for your in-development APIs

Monitor Schedule automated tests and check performance of your APIs

API Manage all aspects of API design, development, and testing

Not sure where to start? [Explore](#) featured APIs, collections, and workspaces published by the Postman community.

[Learn more on Postman Docs](#)

Find and Replace Console Bootcamp Runner Trash ?

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

No Environment

My Workspace

New Import

Create new Collection

+

Enter request URL

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Response

Enter the URL and click Send to get a response

Find and Replace Console

Bootcamp Runner Trash

The screenshot shows the Postman application interface. On the left, there's a sidebar with icons for Collections, APIs, Environments, Mock Servers, Monitors, and History. The 'Collections' section has a red box around the 'Create new Collection' button. The main workspace shows an 'Untitled Request' with a 'GET' method and an empty URL field. Below the method is a table for 'Query Params'. At the bottom, there's a placeholder text 'Enter the URL and click Send to get a response' with a cartoon character holding a rocket. The bottom navigation bar includes 'Find and Replace', 'Console', 'Bootcamp', 'Runner', 'Trash', and a help icon.

Postman

Home Workspaces API Network Reports Explorers Search Postman Invite Settings Bell Upgrade

My Workspace New Import

Collections APIs Environments Mock Servers Monitors Flows History

+

Cats / Test get all cats

Save Send

GET localhost:5050/api/v1/cats

Params Auth Headers (6) Body Pre-req. Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body

Pretty Raw Preview Visualize JSON

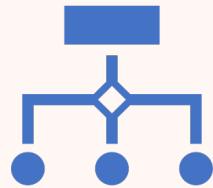
1 {"success": true,
2 "msg": "Show all cats"
3
4 }

200 OK 18 ms 273 B Save Response

Find and Replace Console Capture requests and cookies Bootcamp Runner Trash

The screenshot shows the Postman application interface. On the left, there's a sidebar with categories like Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. A red box highlights the 'Cats' collection. Inside 'Cats', several tests are listed: 'Test get all cats' (highlighted with a red box), 'Test get a cat', 'Test add a new cat', 'Test update a cat', and 'Test delete a cat'. The main workspace shows a test named 'Test get all cats' with a red box around its details. The method is 'GET' and the URL is 'localhost:5050/api/v1/cats'. The 'Send' button is also highlighted with a red box. Below the URL, tabs for Params, Auth, Headers (6), Body, Pre-req., Tests, and Settings are visible. The 'Headers' tab shows six entries. The 'Body' section has tabs for Pretty, Raw, Preview, Visualize, and JSON, with 'Pretty' selected. The response body is displayed in a JSON-like format: '1 {"success": true, "msg": "Show all cats"}'. The status bar at the bottom indicates a 200 OK response with 18 ms and 273 B. Navigation icons for Find and Replace, Console, Capture requests and cookies, Bootcamp, Runner, Trash, and Help are at the bottom.

Design Best Practices



1. Designing for Real-Life Use Cases



2. Designing for a Great Developer Experience

Designing for Real-Life Use Cases

- When designing an API, it's best to make decisions that are grounded in specific, real-life use cases.
- Think about the developers who are using your API.
 - What tasks should they be able to complete with your API?
 - What types of apps should developers be able to build?
- Quite often APIs are designed based on the internal architecture of the application, leaking details of the implementation.
 - leading to confusion for third-party developers and a bad developer experience.

Designing for a Great Developer Experience (1)

- Make It Fast and Easy to Get Started
 - Developers may be using your API to avoid having to build out a secondary product suite to support their main product.
 - Don't make them regret that decision with an API that's difficult to use.
- Work Toward Consistency
 - The API should be reflected in your endpoint names, input parameters, and output responses.
 - Developers should be able to guess parts of your API even without reading the documentation.
 - Consistency helps your existing developers in adapting new features by reducing forks in their code.

Designing for a Great Developer Experience (2)

- Make Troubleshooting Easy
 - Meaningful errors are easy to understand, unambiguous, and actionable.
 - They help developers to understand the problem and to address it.

Error category	HTTP status	HTTP header	Error code (machine-readable)	Error code (human-readable)
System-level error	500	--	--	--
Business logic error	429	Retry - After	rate_limit_exceeded	"You have been rate-limited. See Retry-After and try again."
API Request formatting error	400	--	missing_required_parameter	"Your request was missing a {user} parameter."
Auth error	401	--	invalid_request	"Your ClientId is invalid."

Designing for a Great Developer Experience (3)

- Make Your API Extensible
 - There will always be a need for change and growth as your product evolves.
 - You should create an opportunity for feedback with your top partners
 - “beta” or “early adopter” program
 - In some cases, you might want to version your API.
 - For some companies and products that businesses rely on, maintaining backward-compatible versions is needed.

REST API design guidelines

1. Accept and respond with JSON
2. Use nouns instead of verbs in endpoint paths
3. Name collections with plural nouns
4. Nesting resources for hierarchical objects
5. Handle errors gracefully and return standard error codes
6. Allow filtering, sorting, and pagination
7. Maintain Good Security Practices
8. Cache data to improve performance
9. Versioning our APIs

OpenAPI - <https://www.openapis.org>

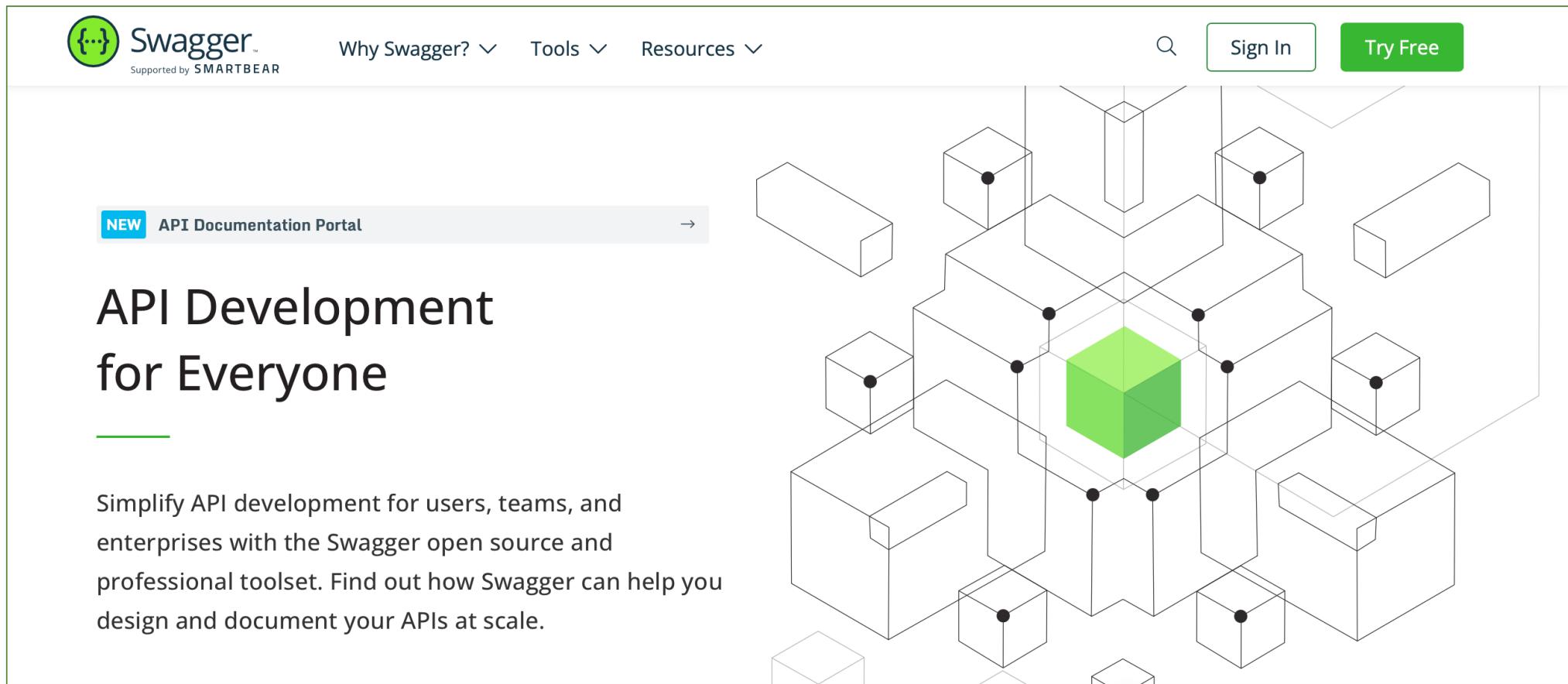
The screenshot shows the homepage of the OpenAPI Initiative. At the top, there is a navigation bar with links for About, Specification, Participate, Governance, Membership, Blog, FAQ, What is OpenAPI?, Events, and social media icons for X, LinkedIn, GitHub, and a search icon. Below the navigation is a dark blue header section with white text that reads "The world's most widely used API description standard". There are two green call-to-action buttons: "View Latest Specs" and "How to Get Involved". The main content area has a white background. On the left, there is a section titled "What is OpenAPI?" with the text: "The OpenAPI Specification provides a formal standard for describing HTTP APIs. This allows people to understand how an API works, generate client code, create". To the right of this text is a flowchart diagram illustrating the API development process:

```
graph TD; Requirements[Requirements  
What should my API do?] --> Design[Design  
What does my API look like?]; Design --> Configure[Configure  
Configure my API infrastructure]; Configure --> Publish[Publish  
Create experience for developers]; Configure --> Develop[Develop  
Implement API based on design]
```

What is OpenAPI?

- **OpenAPI Specification** (formerly Swagger Specification) is an API description format for REST APIs.
- An **OpenAPI** file allows you to describe your entire API, including:
 - Available endpoints (/users) and operations on each endpoint (GET /users, POST /users)
 - Operation parameters Input and output for each operation
 - Authentication methods
 - Contact information, license, terms of use and other information.
- API specifications can be written in YAML or JSON.
 - The format is easy to learn and readable to both humans and machines.

Swagger - <https://swagger.io>



The screenshot shows the official Swagger website. At the top left is the logo "Swagger" with a green circle icon containing three dots, followed by "Supported by SMARTBEAR". To the right are navigation links: "Why Swagger? ▾", "Tools ▾", "Resources ▾", a search bar with a magnifying glass icon, and two buttons: "Sign In" and "Try Free". A banner at the top says "NEW API Documentation Portal". Below the banner, the main headline reads "API Development for Everyone". A descriptive paragraph follows: "Simplify API development for users, teams, and enterprises with the Swagger open source and professional toolset. Find out how Swagger can help you design and document your APIs at scale." To the right of the text is a large, stylized 3D diagram consisting of many white cubes connected by lines, forming a complex network structure. One central cube is highlighted in green.

NEW API Documentation Portal →

API Development for Everyone

Simplify API development for users, teams, and enterprises with the Swagger open source and professional toolset. Find out how Swagger can help you design and document your APIs at scale.

What is Swagger?

- **Swagger** is a set of open-source tools built around the OpenAPI Specification
- **Swagger** can help you design, build, document and consume REST APIs.
- The major Swagger tools include:
 - **Swagger Editor** – browser-based editor where you can write OpenAPI specs.
 - **Swagger UI** – renders OpenAPI specs as interactive API documentation.
 - **Swagger Codegen** – generates server stubs and client libraries from an OpenAPI spec.
- The easiest way to understand the difference is:
 - **OpenAPI** = Specification
 - **Swagger** = Tools for implementing the specification

Why Use OpenAPI and Swagger?

- An **OpenAPI specification** and **Swagger tools** can drive your API development further in various ways:
 - Design-first users: use **Swagger Codegen** to generate a server stub for your API.
 - Use **Swagger UI** to generate interactive API documentation that lets your users try out the API calls directly in the browser.
 - Use the spec to connect API-related tools to your API.
 - And more! Check out the open-source and commercial tools that integrate with Swagger.



About Specification Participate Governance Membership Blog FAQ

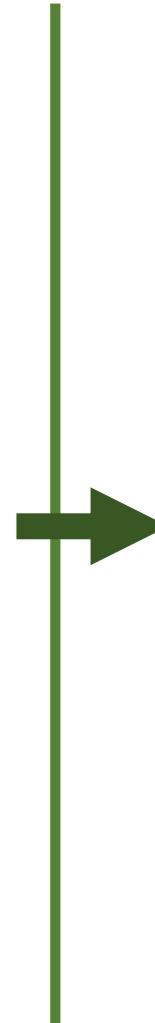


Current Members



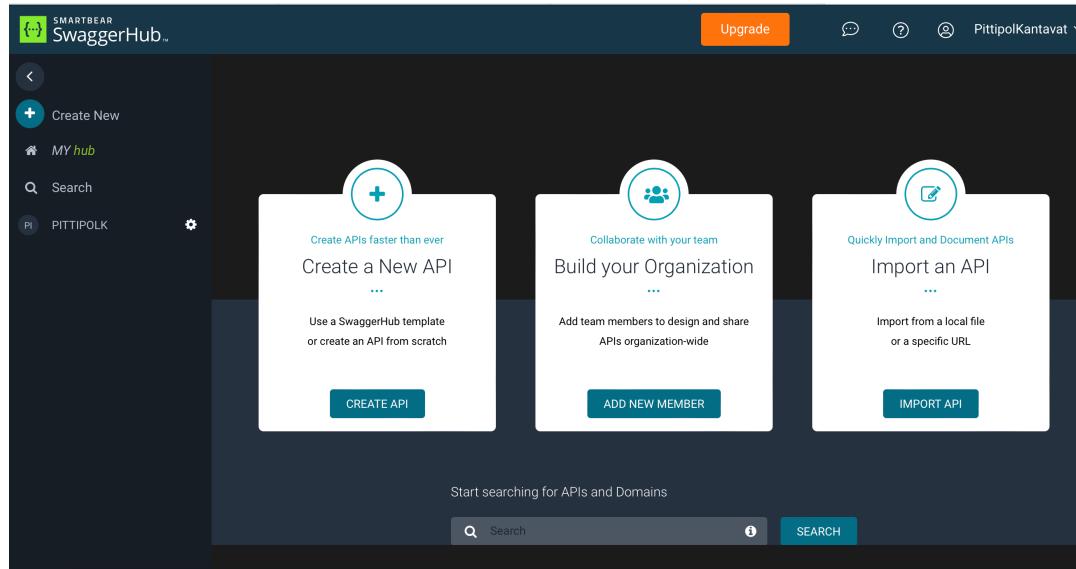
<https://swagger.io/>

The screenshot shows the main landing page of [Swagger](https://swagger.io/). At the top, there's a navigation bar with links for "Why Swagger?", "Tools", and "Resources". Below the navigation is a large heading "API Development for Everyone" with a subtext: "Simplify API development for users, teams, and enterprises with the Swagger open source and professional toolset. Find out how Swagger can help you design and document your APIs at scale." To the right of the text is a 3D wireframe illustration of a cube structure with a central green cube highlighted. At the bottom left is a button labeled "Explore Swagger Tools".



The screenshot shows the "Log In" screen for [SwaggerHub](#). It features a "Log In with GitHub" button at the top, followed by a "LOG IN WITH SSO" button. Below these are fields for "User Name or Email" and "Password". There are links for "Forgot Login Info?" and "LOG IN". At the bottom, it says "New to SwaggerHub? Sign Up" and includes terms of use and privacy policy links.

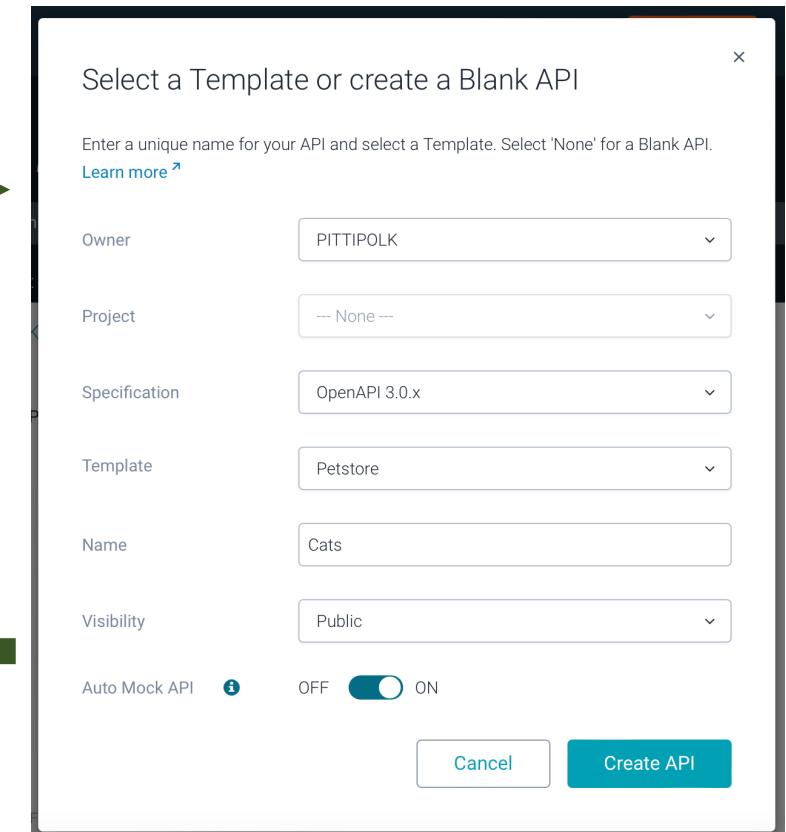
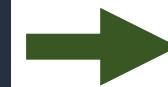
The screenshot shows the "Authorize SwaggerHub SaaS" screen from GitHub. It displays a message from "SwaggerHub SaaS by SmartBear" asking for permission to access the user's Pittipol account. It shows the user's personal data: "Email addresses (read-only)". There are "Cancel" and "Authorize SmartBear" buttons. Below the buttons, it says "Authorizing will redirect to <https://app.swaggerhub.com>". At the bottom, there are status indicators: "Not owned or operated by GitHub", "Created 2 years ago", and "More than 1K GitHub users". A link "Learn more about OAuth" is also present.



The screenshot shows the main interface of the SwaggerHub application. At the top, there's a navigation bar with the SmartBear logo, the text "SwaggerHub", an "Upgrade" button, and user account information ("PittipolKantavat"). Below the navigation is a dark header with three main call-to-action cards:

- Create a New API**: Use a SwaggerHub template or create an API from scratch. Includes a "CREATE API" button.
- Build your Organization**: Add team members to design and share APIs organization-wide. Includes an "ADD NEW MEMBER" button.
- Import an API**: Import from a local file or a specific URL. Includes an "IMPORT API" button.

Below these cards is a search bar with placeholder text "Start searching for APIs and Domains".

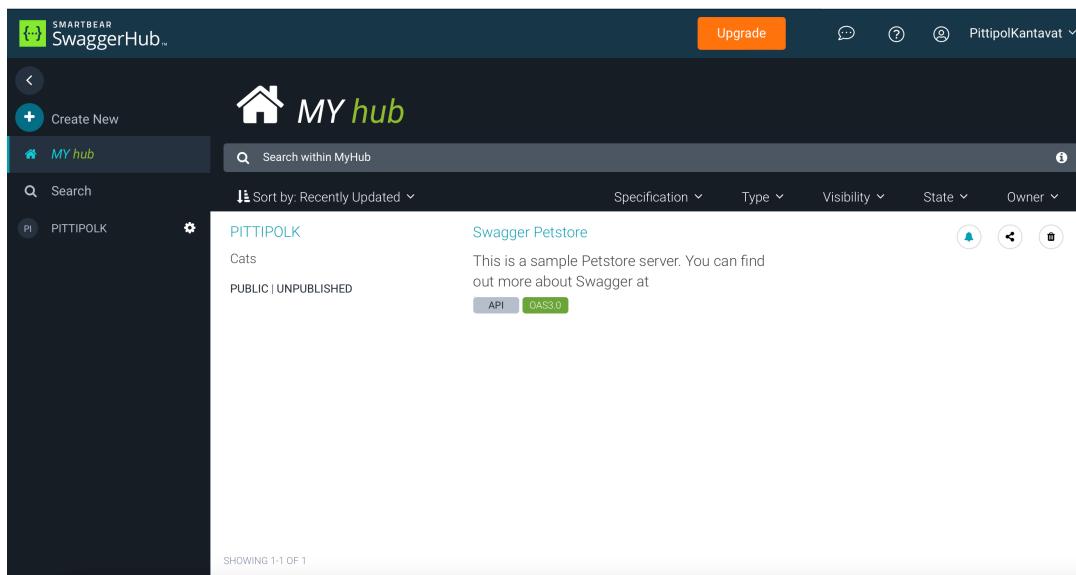


Select a Template or create a Blank API

Enter a unique name for your API and select a Template. Select 'None' for a Blank API.
[Learn more](#)

Owner	PITTIPOLK
Project	--- None ---
Specification	OpenAPI 3.0.x
Template	Petstore
Name	Cats
Visibility	Public
Auto Mock API	OFF <input checked="" type="checkbox"/> ON

[Cancel](#) [Create API](#)

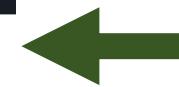


The screenshot shows the "MY hub" section of the application. The header includes the "MY hub" icon, a search bar, and user account information ("PittipolKantavat"). Below the header is a search bar with placeholder text "Search within MyHub".

The main content area displays a list of APIs under the owner "PITTIPOLK". There is one entry:

- Cats**: A sample Petstore server. You can find out more about Swagger at [this link](#).
Specification: API
Type: OAS3.0

At the bottom left, it says "SHOWING 1-1 OF 1".



The “Petstore” template

The screenshot shows the SwaggerHub interface for the "Petstore" template. On the left, the API structure is displayed with endpoints for `Cats` and version `1.0.0`. The main area shows the `Info` section of the OpenAPI 3.0 specification, which includes details like the title "Swagger Petstore", terms of service, contact information, and license information. The right side shows a preview of the API documentation, titled "Swagger Petstore", with sections for Terms of service, Contact the developer, Apache 2.0, and Find out more about Swagger. A "Servers" section at the bottom lists the URL `https://virtserver.swaggerhub.com/PITTIPOLK/Cats/1.0.0`. At the bottom right, there is an "Authorize" button with a lock icon.

```
openapi: 3.0.0
info:
  description: |
    This is a sample 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/).
  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'
servers:
  - description: SwaggerHub API Auto Mocking
    url: https://virtserver.swaggerhub.com/PITTIPOLK/Cats/1.0.0
    - url: 'https://petstore.swagger.io/v2'
```

Last Saved: 9:18:49 pm - Jan 31, 2024 ✓ VALID

Swagger Petstore
1.0.0 OAS 3.0

This is a sample Petstore server. You can find out more about Swagger at <http://swagger.io> or on [irc.freenode.net, #swagger](#).

Terms of service
Contact the developer
Apache 2.0
Find out more about Swagger

Servers
<https://virtserver.swaggerhub.com/PITTIPOLK/Cats/1....>

Authorize

The screenshot shows the SwaggerHub interface for a project named "Cats" at version 1.0.0. The main area displays the "Info" tab with a title of "Petstore for my cats" and a description of "This is a sample Petstore server. You can find out more about Swagger at <http://swagger.io> or on [irc.freenode.net, #swagger](#)". On the right, the "Codegen Options" sidebar is open, showing various server stub options like "nodejs-server" (which is selected and highlighted with a red box), "php-silex", "php-symfony", etc. A large green arrow points from the "nodejs-server" option in the sidebar down to a generated project window titled "nodejs-server-server-generated". This window displays a file tree with files like "api", "controllers", "service", "utils", "index.js", "package.json", and "README.md".

SMARTBEAR
SwaggerHub™

Cats 1.0.0

Info

Tags

Search

pet

- POST /pet
- PUT /pet
- GET /pet/findByStatus
- GET /pet/findByTags
- GET /pet/{petId}
- POST /pet/{petId}
- DELETE /pet/{petId}
- POST /pet/{petId}/uploadImage

store

user

Models

SAVE SYNC

Pittipol

Export

Petstore for my cats

1.0.0

[Base URL: virtserver]

This is a sample Petstore server. You can find out more about Swagger at <http://swagger.io> or on [irc.freenode.net, #swagger](#).

Terms of service

Contact the developer

Apache 2.0

Find out more about Swagger

Schemes

HTTPS

Codegen Options

Client SDK

Server Stub

Documentation

Download API

msf4j

nancyfx

nodejs-server

php-silex

php-symfony

pistache-server

python-flask

rails5

restbed

rust-server

scalatra

sinatra

Authorize

nodejs-server-server-generated

nodejs-server-server-generated.zip

api

controllers

service

utils

index.js

package.json

README.md

<https://www.npmjs.com/package/swagger-jsdoc>

Nominating Presidential Muppets

Pro Teams Pricing Documentation

npm Search packages Sign Up Sign In

swagger-jsdoc DT

6.2.8 • Public • Published a year ago

[Readme](#) [Code](#) Beta [6 Dependencies](#) [461 Dependents](#) [93 Versions](#)

swagger-jsdoc

This library reads your **JSDoc**-annotated source code and generates an **OpenAPI (Swagger)** specification.

downloads 1.8M/month CI failing

Getting started

Imagine having API files like these:

```
/**  
 * @openapi
```

Install

```
> npm i swagger-jsdoc
```

Repository

github.com/Surnet/swagger-jsdoc

Homepage

github.com/Surnet/swagger-jsdoc

Weekly Downloads

426,350



```
server.js — MyCat

JS server.js ×

JS server.js > ...

1  const express = require('express');
2  const dotenv = require('dotenv');
3
4  //Load env vars
5  dotenv.config({path: './config/config.env'});
6
7  const app=express();
8
9  const swaggerJsDoc = require('swagger-jsdoc');
10 const swaggerUI = require('swagger-ui-express');
11
12 const swaggerOptions={
13     swaggerDefinition:{
14         info: {
15             title: 'Library API',
16             version: '1.0.0'
17         }
18     },
19     apis:['server.js'],
20 };
21
22 const swaggerDocs=swaggerJsDoc(swaggerOptions);
23 app.use('/api-docs',swaggerUI.serve, swaggerUI.setup(swaggerDocs));
24
25 /**
26 * @swagger
27 * /cats:
28 *   get:
29 *     description: Get all cats!
30 *     responses:
31 *       200:
32 *         description: Success
33 */
34
35 app.get('/api/v1/cats', (req,res) => {
36     res.status(200).json({success:true, msg:'Show all cats'});
37 });


```

1. เพิ่ม code ที่ server.js
2. เปิด browser และไปที่ url:
<http://localhost:5050/api-docs/>



The screenshot shows the Swagger UI interface for the 'Library API'. The title bar says 'Swagger' and 'Supported by SMARTBEAR'. Below it, the API title is 'Library API 1.0.0'. A green 'Authorize' button is visible. The main content area is titled 'default'. It shows a 'GET /cats' endpoint with the description 'Get all cats!'. Under 'Parameters', it says 'No parameters'. Under 'Responses', it says 'Response content type application/json'. There is a 'Try it out' button next to the parameters section.

```
const swaggerJsDoc = require('swagger-jsdoc');
const swaggerUI = require('swagger-ui-express');

const swaggerOptions={
  swaggerDefinition:{
    info: {
      title: 'Library API',
      version: '1.0.0'
    }
  },
  apis:['server.js'],
};

const swaggerDocs=swaggerJsDoc(swaggerOptions);
app.use('/api-docs',swaggerUI.serve,swaggerUI.setup(swaggerDocs));

/**
 * @swagger
 * /cats:
 *   get:
 *     description: Get all cats!
 *     responses:
 *       200:
 *         description: Success
 */

```

1. เปิดไฟล์ server.js
2. Copy -> Paste
3. Save

Reference

Book

- Brenda Jin, Saurabh Sahni, and Amir Shevat. 2018.
Designing Web APIs: Building APIs That Developers Love.
O'Reilly Media, Inc.

Web Blog

- <https://developer.mozilla.org/en-US/docs/Web/HTTP>Status>
- <https://swagger.io/resources/articles/best-practices-in-api-design/>
- <https://swagger.io/docs/specification/about/>
- <https://www.openapis.org>
- <https://stackoverflow.blog/2020/03/02/best-practices-for-rest-api-design/#h-accept-and-respond-with-json>

