

Postman Tutorial

Let's code our own APIs.

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

The screenshot shows the official Node.js website. The main heading is "Run JavaScript Everywhere". Below it, a paragraph explains what Node.js is: "Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts." There are two prominent buttons: "Get Node.js®" and "Get security support for EOL Node.js versions". To the right, a code editor window displays a script named "server.mjs" which creates a simple HTTP server. The code is as follows:

```
1 // server.mjs
2 import { createServer } from 'node:http';
3
4 const server = createServer((req, res) => {
5   res.writeHead(200, { 'Content-Type': 'text/plain' });
6   res.end('Hello World!\n');
7 });
8
9 // starts a simple http server locally on port 3000
10 server.listen(3000, '127.0.0.1', () => {
11   console.log('Listening on 127.0.0.1:3000');
12 });
13
14 // run with `node server.mjs`
```

Below the code, there is a "Copy to clipboard" button. At the bottom of the page, there are links for "Learning materials" and footer navigation including "v22.18.0 Latest LTS", "v24.6.0 Latest Release", "Trademark Policy", "Privacy Policy", "Code of Conduct", "Security Policy", and social media links for GitHub, Discord, Mozilla, Twitter, and LinkedIn.

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

Visual Studio Code Docs Updates Blog API Extensions MCP FAQ Dev Days ⌂ ⌂ Download Try [MCP servers](#) to extend agent mode in VS Code! ×

Download Visual Studio Code

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

Windows [Windows 10, 11](#)

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

.deb [Debian, Ubuntu](#)

.deb [x64](#) [Arm32](#) [Arm64](#)

.rpm [Red Hat, Fedora, SUSE](#)

.rpm [x64](#) [Arm32](#) [Arm64](#)

.tar.gz

.tar.gz [x64](#) [Arm32](#) [Arm64](#)

Snap [Snap Store](#)

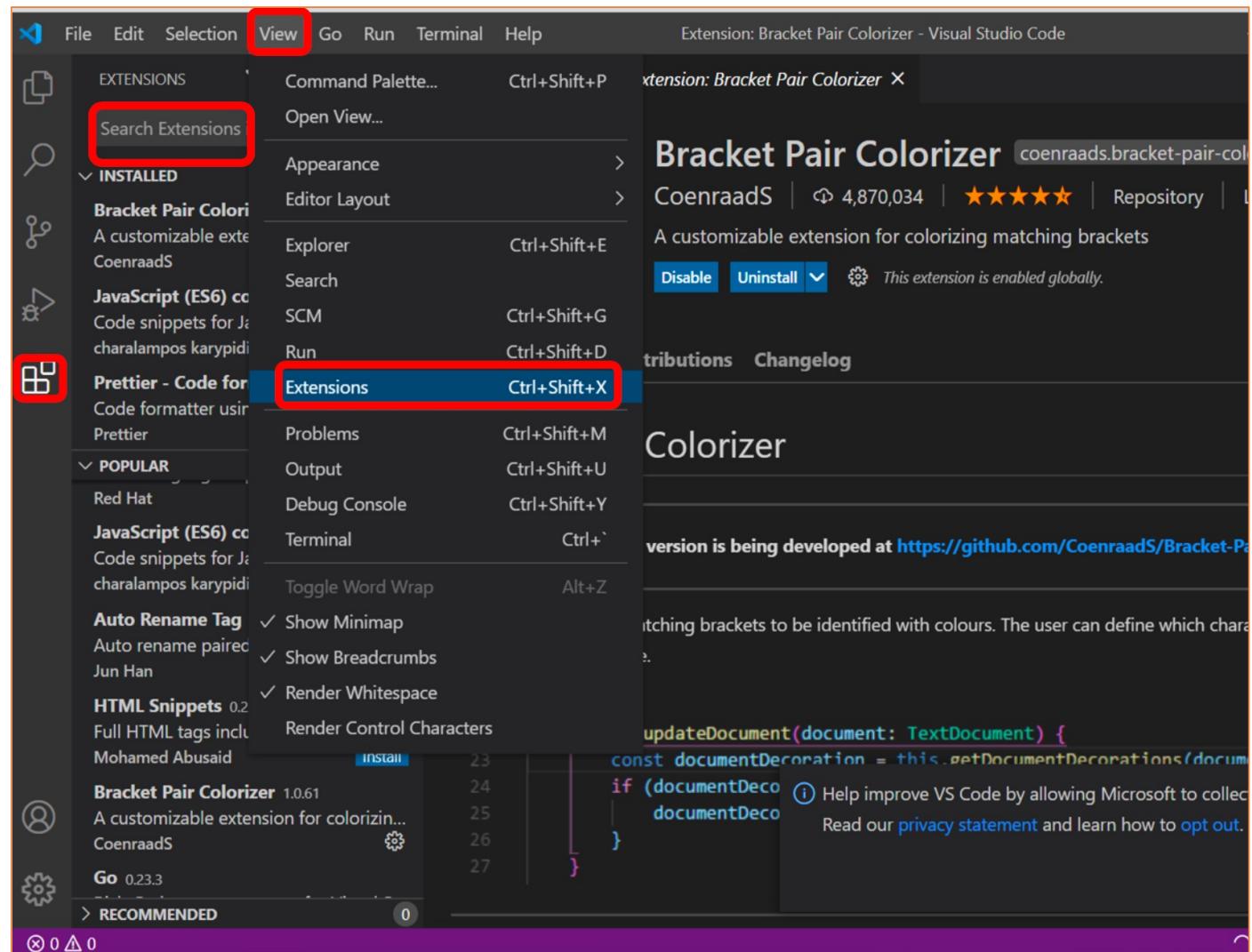
CLI [x64](#) [Arm32](#) [Arm64](#)

Mac [macOS 11.0+](#)

.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 file `package.json` open in the editor. The file content is as follows:

```
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 }
```

A red box highlights the entire `scripts` object, and another red box highlights the `"dev": "nodemon server.js"` line. A yellow arrow points from the text "เพิ่มบรรทัดนี้ใน" to the `"dev": "nodemon server.js"` line. The terminal below shows the command `npm i -D nodemon` being run.

เพิ่มบรรทัดนี้ใน
“package.json”

"dev": "nodemon server.js"

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

- EXPLORER** sidebar: Shows files like package.json, config.env, node_modules, package-lock.json, package.json, and server.js. The folder "config" and its "config.env" file are highlighted with a red box.
- OPEN EDITORS** tab bar: Shows package.json and config.env. config.env is the active editor, indicated by a red border.
- config.env** editor content:

```
config > config.env
1 PORT = 5050
2 NODE_ENV = development
```

A yellow arrow points from the third step in the instructions to this code block.
- TERMINAL**: Shows the output of npm install -D nodemon, including audit results and package addition details.
- STATUS BAR**: Shows Ln 1, Col 12, Spaces: 4, UTF-8, LF, Environment Variables, Prettier, and a bell icon.

- สร้าง folder "config"
- สร้างไฟล์ "config/config.env"
- ใส่ code

PORT = 5050 // ตามสัง打球
NODE_ENV = development

server.js — MyCat

EXPLORER ...

OPEN EDITORS

- ({}) package.json
- ({}) config.env config
- JS server.js

MYCAT

- config
- config.env
- > node_modules
- ({}) package-lock.json
- ({}) package.json
- JS server.js

```
JS server.js > app.delete('/api/v1/cats/:id') callback > msg
```

```
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));
```

OUTLINE

Ln 26, Col 56 Spaces: 4 UTF-8 LF () JavaScript ✕ Prettier 8 🔍

```
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

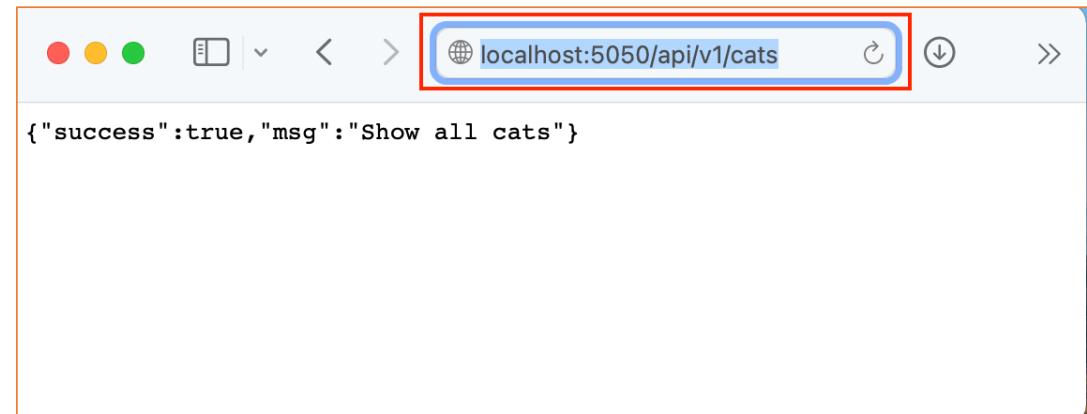
5.Run server / Test

- ที่ Terminal สั่งรัน \$ 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 screenshot shows the Postman website with a search bar at the top. Below it, there's a large heading "Download Postman" and a subtext: "Download the app to get started using the Postman API Platform today. Or, if you prefer a browser experience, you can try the web version of Postman." On the left, there's a section titled "The Postman app" with download links for "Mac Apple Chip" and "Mac Intel Chip". At the bottom, there's a note about agreeing to the Privacy Policy and Terms, a link to Release Notes, and a note for non-Mac users. On the right, there's a screenshot of the Postman desktop application interface showing a workspace named "Notion's Public Workspace" with a database collection and a specific API request for "Notion API / Databases / Retrieve a database".

Product ▾ Resources ▾ Enterprise Pricing Contact Sales Sign In Sign Up for Free

Download Postman

Download the app to get started using the Postman API Platform today. Or, if you prefer a browser experience, you can try the web version of Postman.

The Postman app

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

Mac Apple Chip Mac Intel 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](#))

Home Workspaces API Network Overview GET Retrieve a database

Notion's Public Workspace New Import

Collection Environment History

+ Overview

Notion API

Users Databases

GET Retrieve a database

200 Success- Retriev...
POST Query a database
POST Create a database
PATCH Update a database

Params Auth Headers(10) Body Script

Query params

KEY	VALUE
Key	Value

Path variables

KEY	VALUE
id	{DATABASE_ID}



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



+ Invite

- 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

Collections APIs Environments Mock Servers Monitors History

Create New

Building Blocks

HTTP Request Create a basic HTTP request

Environment Save values you frequently use in an environment

WebSocket Request BETA Test and debug your WebSocket connections

Collection Save your requests in a collection for reuse and sharing

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

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

Upgrade

My Workspace

New Import

Create new Collection

Untitled Request

GET Untitled Request X + ⚙️

Save ⚙️ 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 ?14

Postman

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

My Workspace New Import GET Test get all cats + ⚙️ No Environment

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 200 OK 18 ms 273 B Save Response

Pretty Raw Preview Visualize JSON ⚙️

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

Find and Replace Console Capture requests and cookies Bootcamp Runner Trash 15 ⓘ

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. Below these are buttons for 'New', 'Import', and a search bar. The main workspace is titled 'Cats / Test get all cats'. It shows a test collection named 'Cats' containing several tests: 'GET Test get all cats', 'GET Test get a cat', 'POST Test add a new cat', 'PUT Test update a cat', and 'DEL Test delete a cat'. The 'GET Test get all cats' test is highlighted with a red box. The request details show a 'GET' method and the URL 'localhost:5050/api/v1/cats'. The 'Send' button is also highlighted with a red box. Below the request details, the 'Body' section displays a JSON response with a success message. The response body is highlighted with a large orange oval.