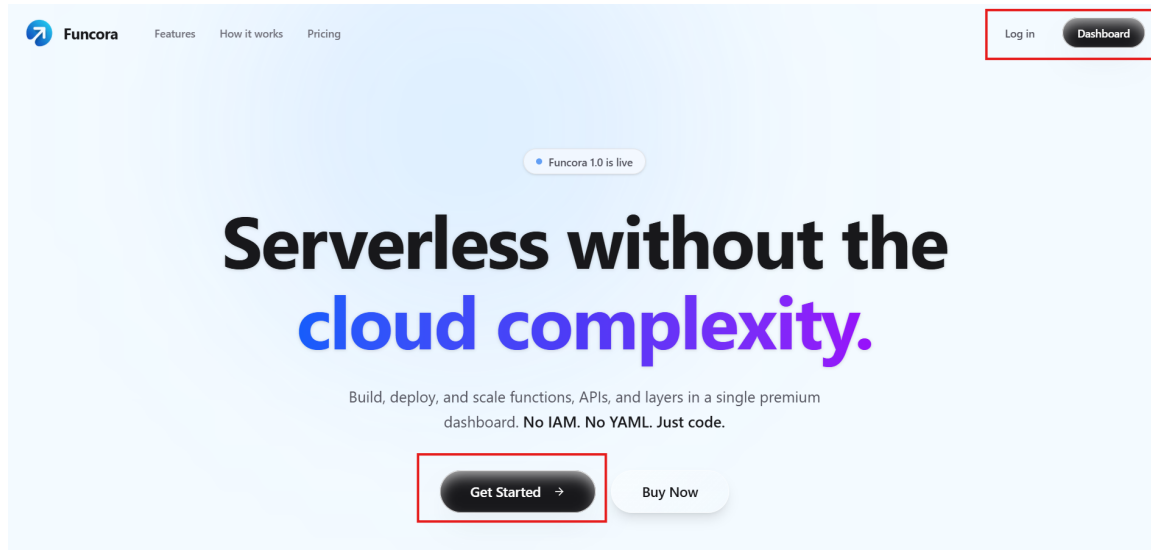


1. Introduction

Funcora is a robust serverless management platform that empowers developers to build, deploy, and scale applications with ease. This guide outlines the core workflows for managing projects, functions, APIs, and automated tasks within the ecosystem.

2. Access & Authentication

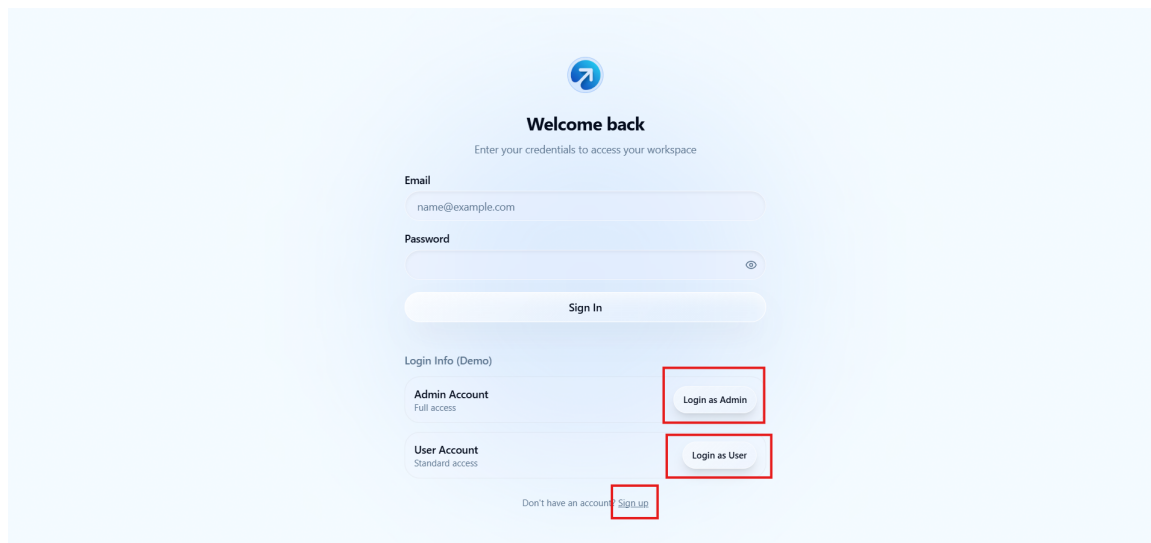
To begin, navigate to the application dashboard. The landing page provides options to view the live demo or log in directly.



2.1. Role-Based Access Control

The platform distinguishes between two primary roles:

- Admin: Full access, including the 'User Management' module to control permissions.
- User: Standard access to project and function management features, restricted from administrative settings.



User Management :

The screenshot shows the 'User Management' page in the Funcora Serverless Platform. The left sidebar contains a menu with 'User Management' highlighted. The main content area displays a table of users with columns for USER, ROLE, PERMISSIONS, JOINED, and ACTIONS. A search bar is located at the top right of the table area. The table lists three users: 'mock@example.com', 'user@Mounikai.com', and 'admin@Mounikai.com'. The 'admin' user has the role 'Admin' and 'Full Access' permissions. A 'Manage' button is present for each user. The page footer indicates 'Page 1 of 1' and includes navigation links for 'Previous' and 'Next'.

USER	ROLE	PERMISSIONS	JOINED	ACTIONS
mock@example.com 00000000...	User	0 Enabled	1/24/2026	Manage
user@Mounikai.com 25bee5a4...	User	25 Enabled	1/19/2026	Manage
admin@Mounikai.com 2bc56504...	Admin	Full Access	1/19/2026	Manage

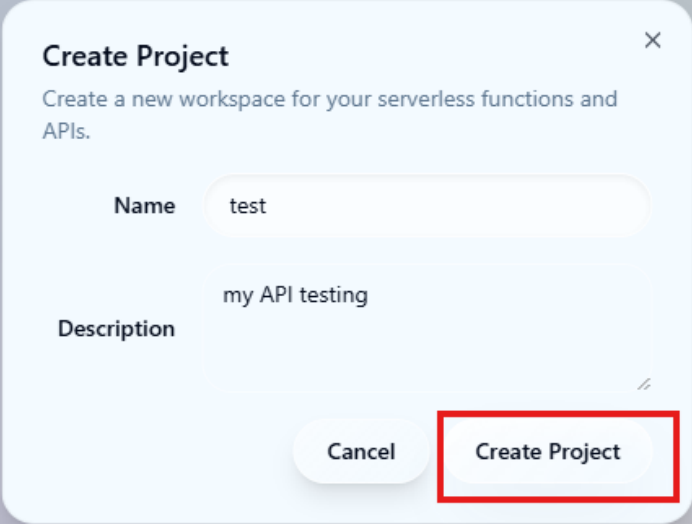
3. Project Management

Resources in Funcora are organized into Projects. You can select an existing project or initialize a new one via the 'New Project' interface.

The screenshot shows the 'Projects' page in the Funcora Serverless Platform. The left sidebar contains a menu with 'Projects' highlighted. The main content area displays a list of projects: 'AI Suite' and 'NodeChat'. Each project card shows the number of functions and APIs, the plan type, and the last update time. A 'New Project' button is located at the top right of the page. A 'Create New Project' button is also visible at the bottom right of the project list.

Project Name	Plan	Functions	APIs	Last Updated	Location
AI Suite	Standard Plan	2	1	Updated 13 days ago	main
NodeChat	Standard Plan	1	1	Updated 13 days ago	main

When creating a project, provide a unique Name and Description. Once created, you will be redirected to the Project Dashboard, which provides a high-level view of all associated Functions, Cron Jobs, and API Gateways.



The image shows a 'Create Project' dialog box with a light blue header and a white body. The title 'Create Project' is in bold, followed by the subtitle 'Create a new workspace for your serverless functions and APIs.' Below this, there are two input fields: 'Name' with the value 'test' and 'Description' with the value 'my API testing'. At the bottom right, there are two buttons: 'Cancel' and 'Create Project'. The 'Create Project' button is highlighted with a red rectangular border.

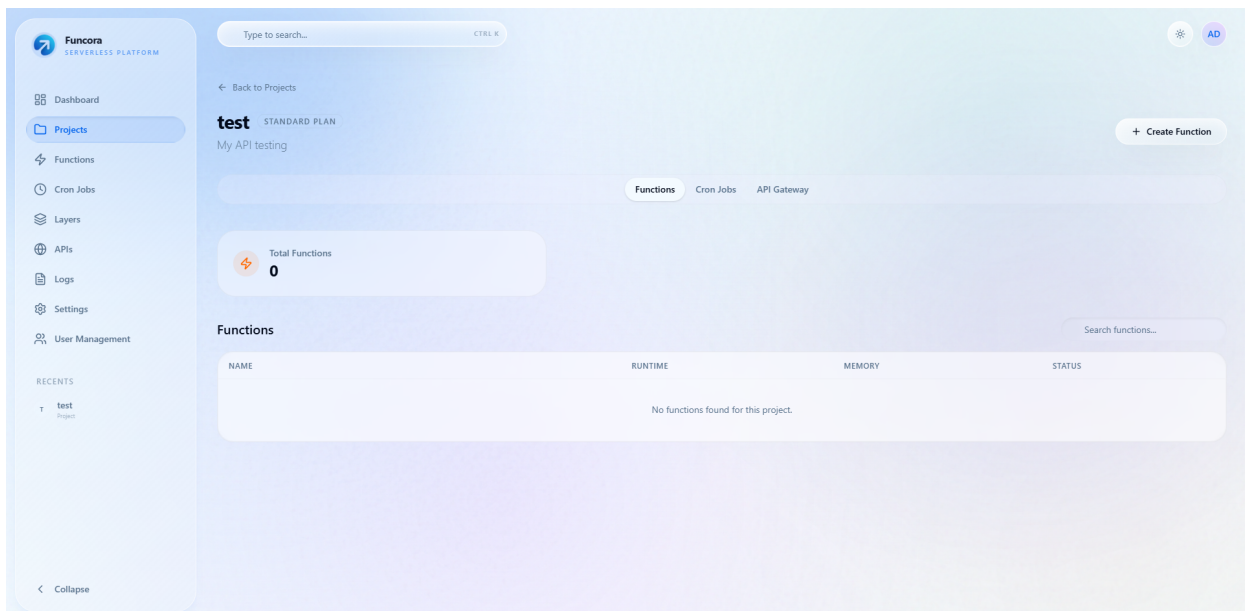
Create Project ×

Create a new workspace for your serverless functions and APIs.

Name test

Description my API testing

Cancel Create Project



The image shows the 'Funcora' Project Dashboard. The left sidebar contains a navigation menu with items: Dashboard, Projects (selected), Functions, Cron Jobs, Layers, APIs, Logs, Settings, and User Management. Below this is a 'RECENTS' section showing a project named 'test'. The main content area has a search bar at the top, a 'Back to Projects' link, and a project header for 'test' (STANDARD PLAN) with a 'My API testing' description and a '+ Create Function' button. Below the header is a tabbed interface with 'Functions' selected. A 'Total Functions' card shows a count of 0. A table with columns 'NAME', 'RUNTIME', 'MEMORY', and 'STATUS' is shown, with a message 'No functions found for this project.' at the bottom.

Funcora
SERVERLESS PLATFORM

Type to search... CTRL K

⚙️ AD

← Back to Projects

test STANDARD PLAN + Create Function

My API testing

Functions Cron Jobs API Gateway

Total Functions
0

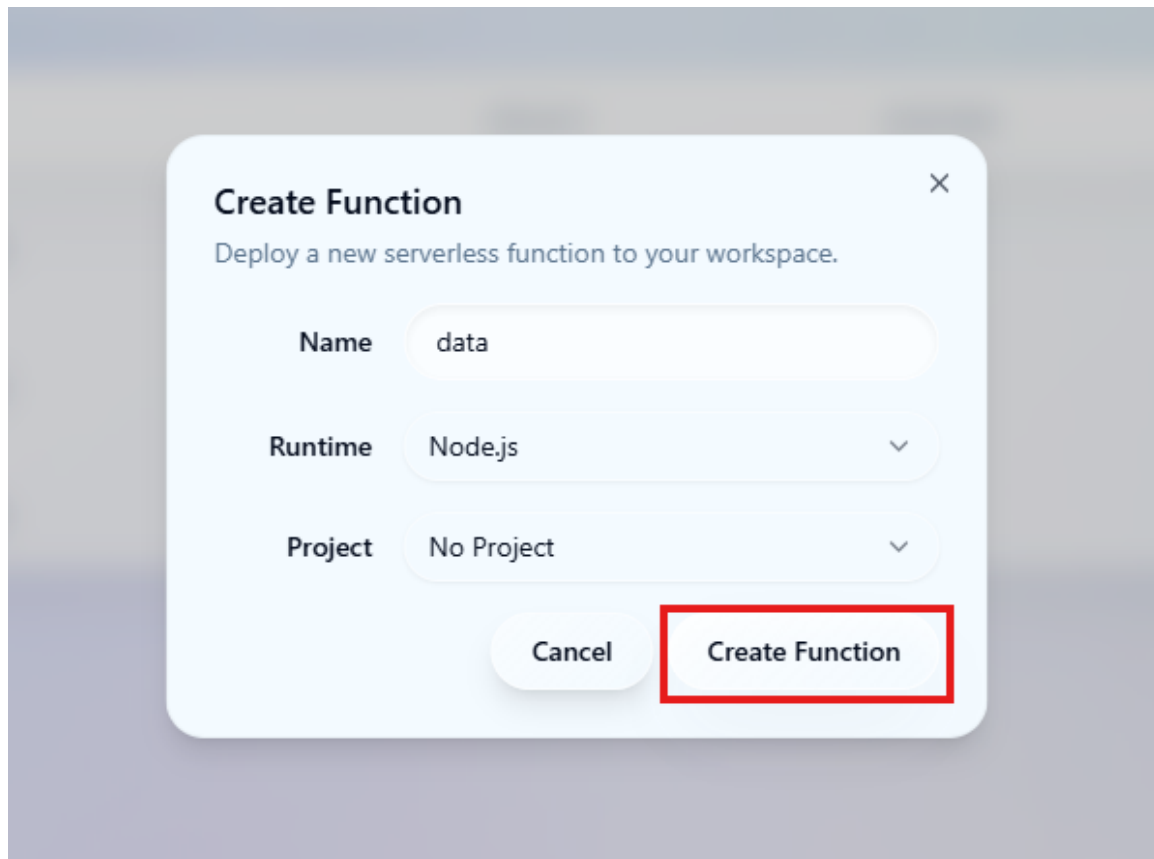
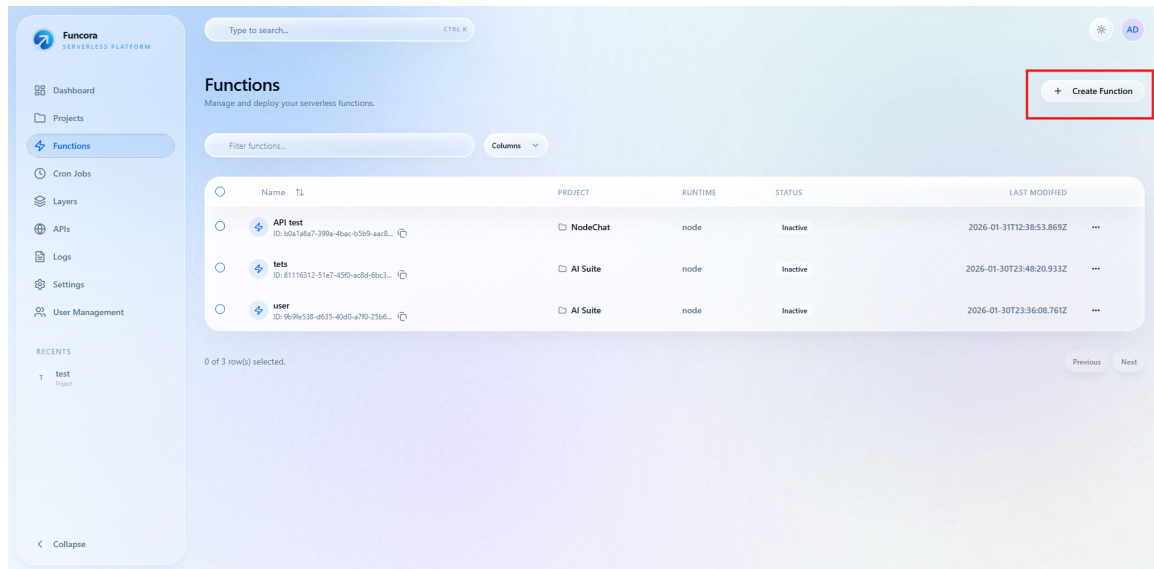
Functions Search functions...

NAME	RUNTIME	MEMORY	STATUS
No functions found for this project.			

← Collapse

4. Serverless Functions

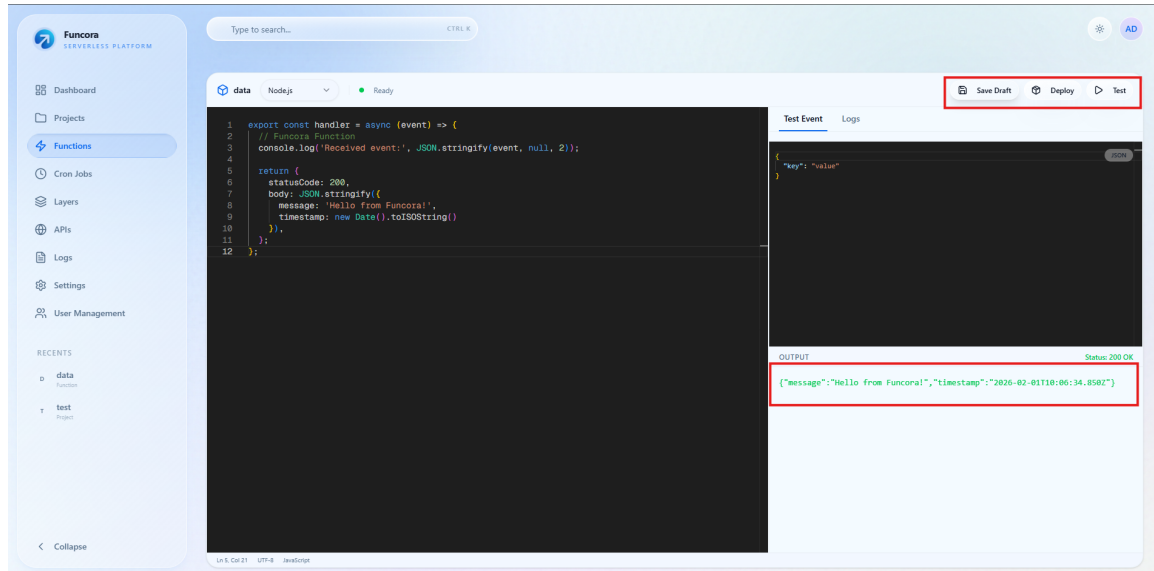
Functions are the core units of logic. To create a function, navigate to the Function route and select 'Create Function'. Specify the function name and runtime environment (currently supporting Node.js, with Python and Java support on the roadmap).



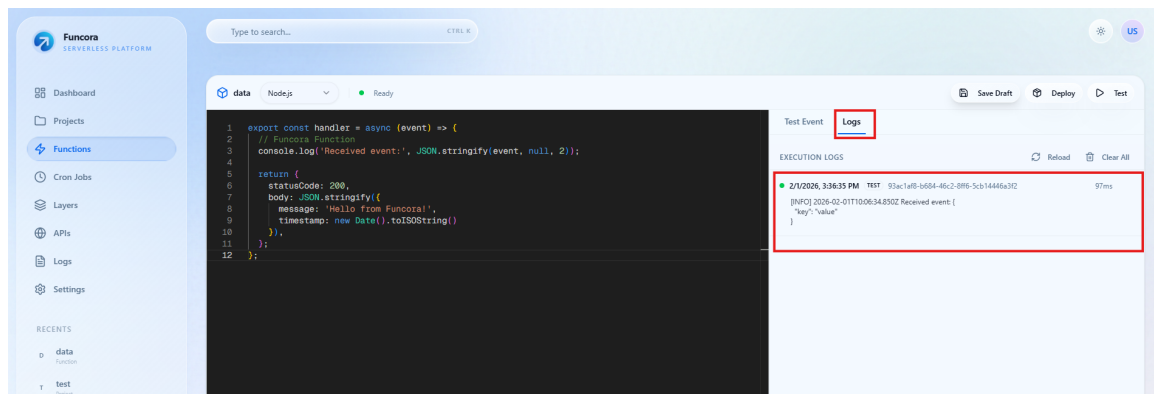
4.1. The Function Editor

The integrated editor allows for real-time code development. It features three key actions:

1. **Save Draft:** Save changes locally to the editor without deploying to the live API.
2. **Test:** Execute the code in the sandbox environment to verify logic (viewable in the Output log).
3. **Deploy:** Push the tested code to production, making it live.

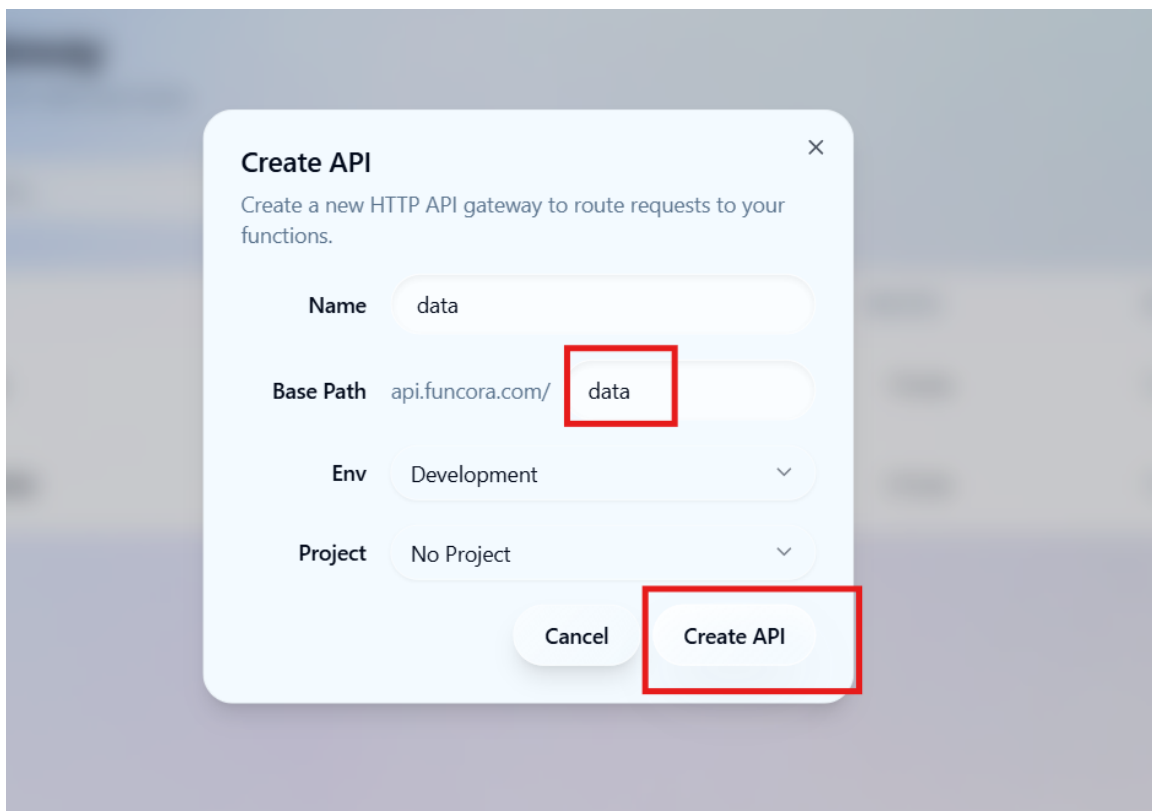
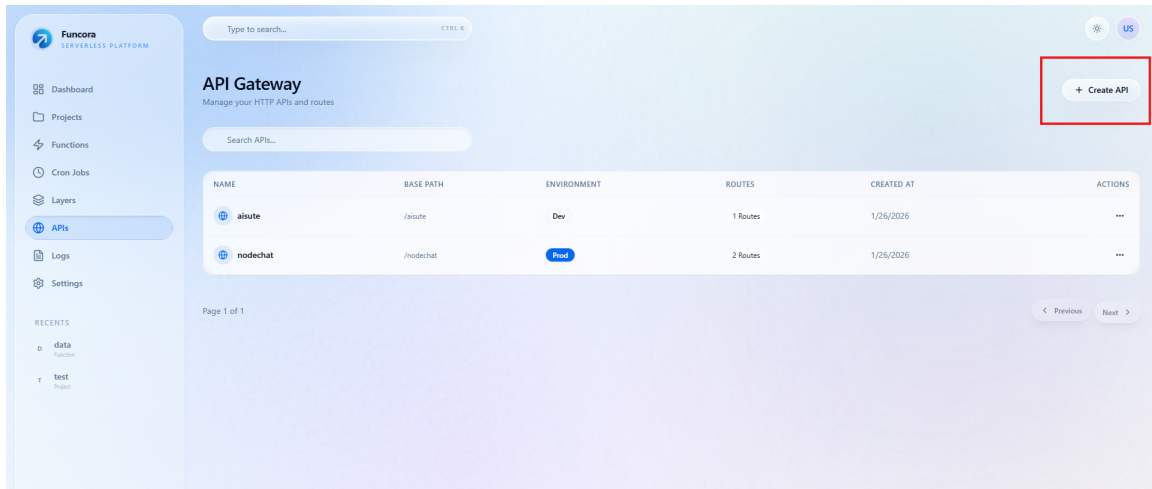


Monitor execution status and debug issues using the comprehensive Logs tab.



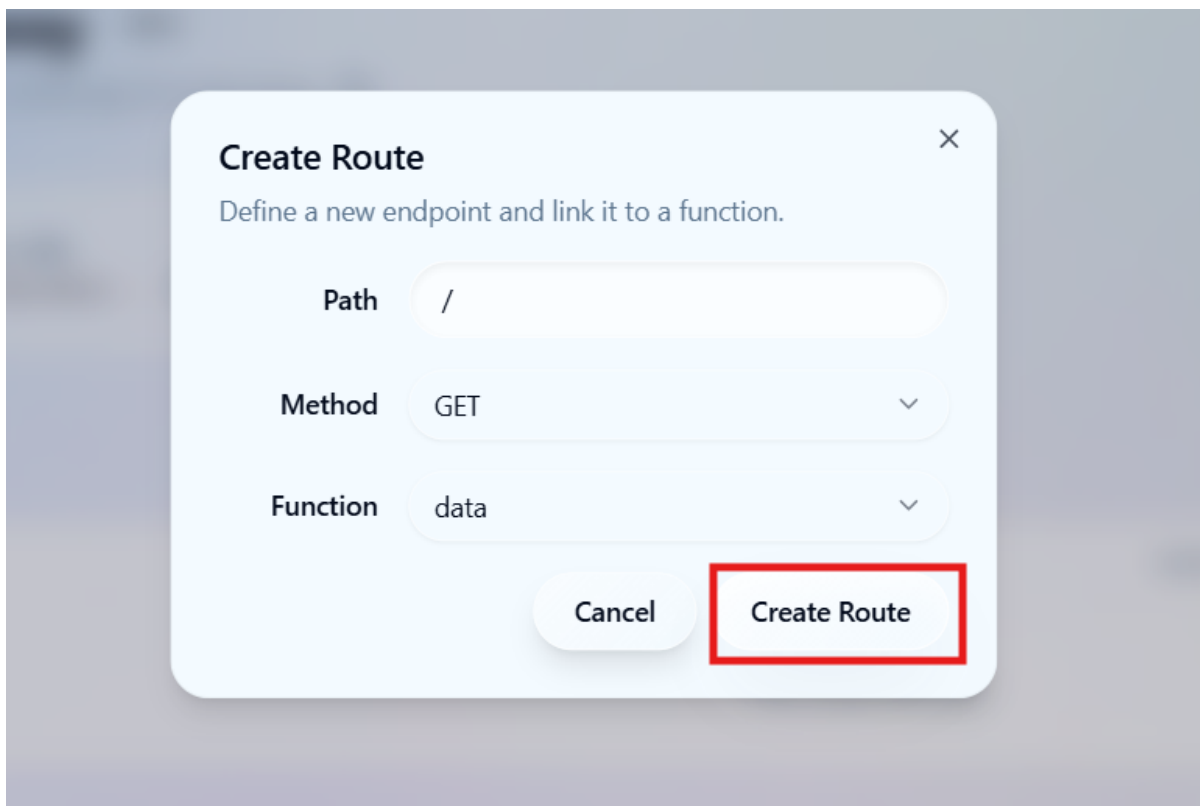
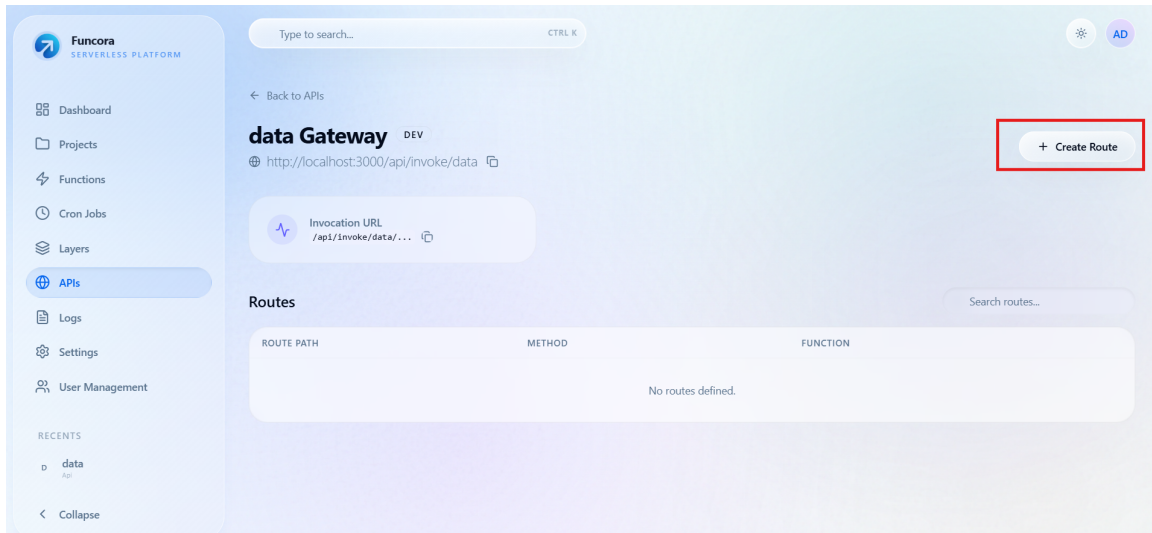
5. API Gateway Integration

Expose your functions to the world via the API Gateway. Start by creating a new API configuration, defining the Name, Base Path (endpoint root), and Environment (e.g., Prod, Test).

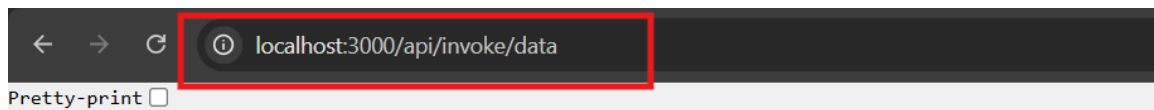
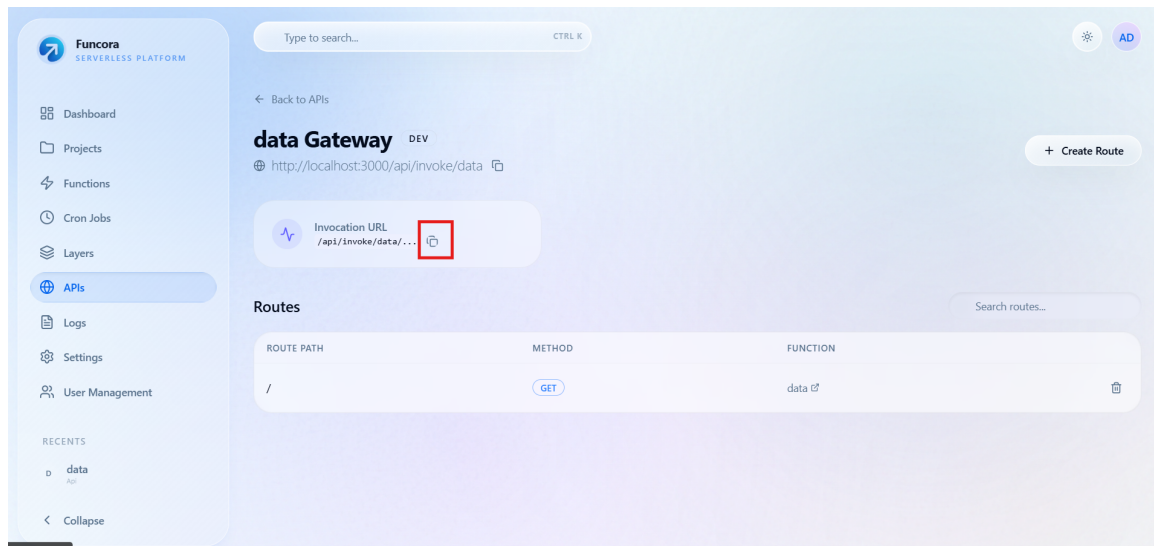


5.1. Route Configuration

Once the API is established, define Routes to map HTTP requests to specific functions. Configure the Path (endpoint suffix), HTTP Method (GET, POST, PUT, DELETE), and the target Function.



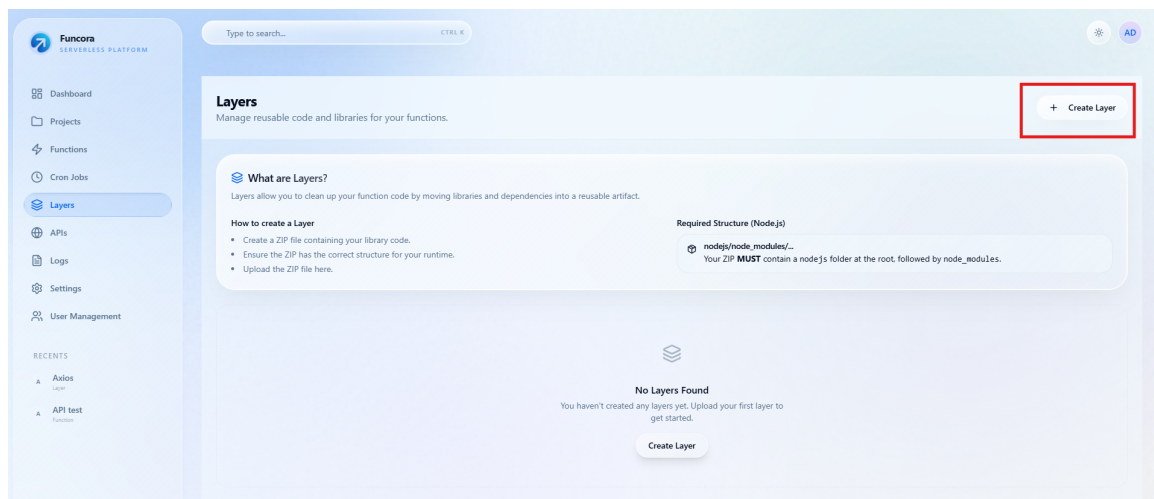
After configuration, copy the API endpoint and validate it using tools like Postman.

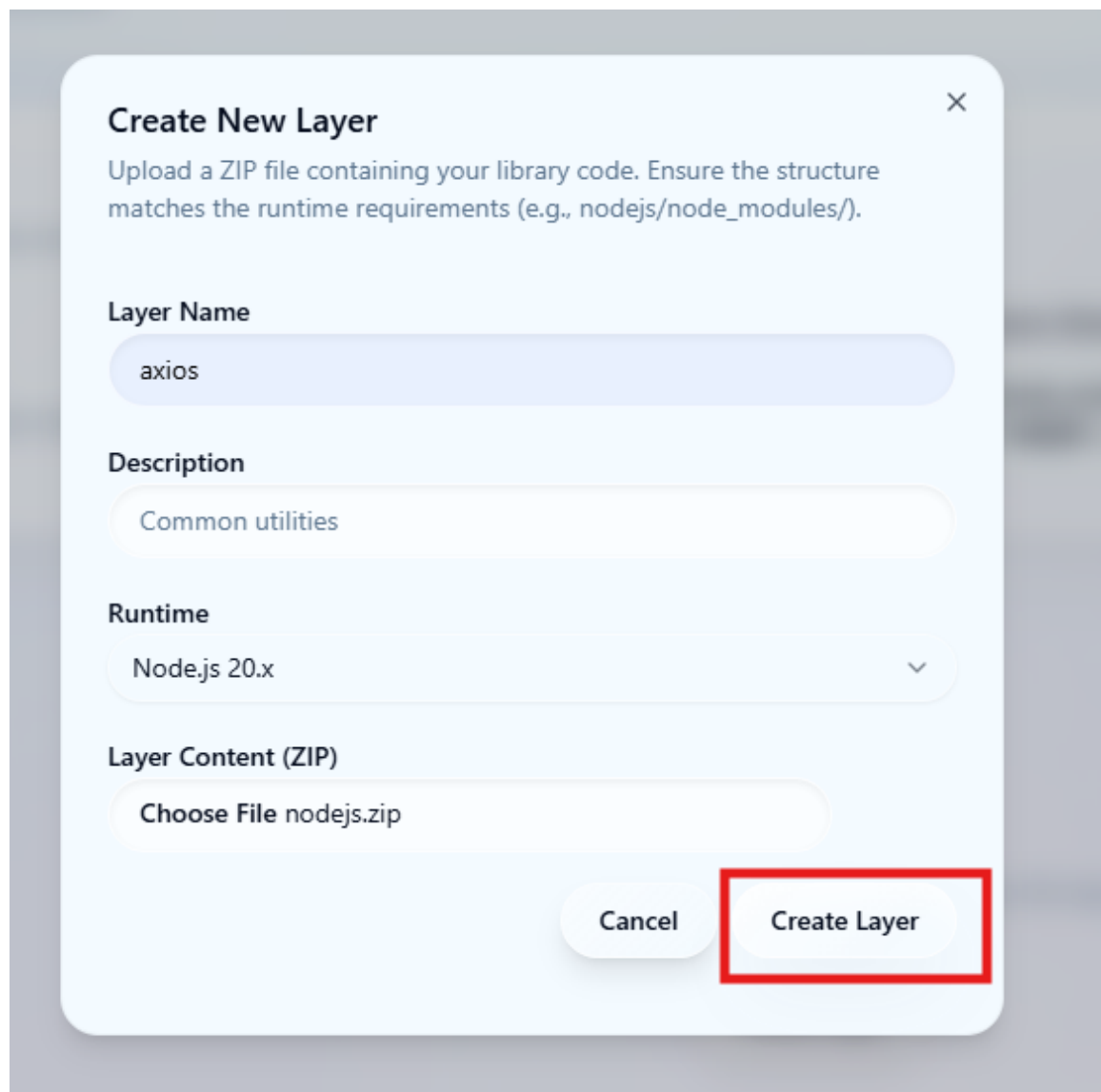


```
{"message\\":\\"Hello from Funcora!\\",\\"timestamp\\":\\"2026-02-01T11:31:23.631Z\\"}"
```

6. Dependency Management (Layers)

For external libraries (e.g., NPM packages), Funcora uses 'Layers'. This allows you to package dependencies once and reuse them across multiple functions.





Create New Layer ×

Upload a ZIP file containing your library code. Ensure the structure matches the runtime requirements (e.g., nodejs/node_modules/).

Layer Name
axios

Description
Common utilities

Runtime
Node.js 20.x ▼

Layer Content (ZIP)
Choose File nodejs.zip

Cancel Create Layer

To create a layer, upload a .zip file containing your dependencies. Structure your zip file as follows (for Node.js):

```
mkdir my-node-layer
cd my-node-layer
mkdir nodejs
cd nodejs
npm init -y
npm install <package-name>
zip -r my-node-layer.zip nodejs
```

Once uploaded, attach the layer to any function via the 'Add Layer' interface.

Functions

Cron Jobs

Layers

APIs

Logs

Settings

User Management

RECENTS

API test

data

axios

3

export const handler = async (event) => {

4

console.log('Received event: ', JSON.stringify(event, null, 2));

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

2025

{

1

"key": "value"

2

}

OUTPUT

Run a test to see output

LAYERS (0/5)

+ Add Layer

No layers attached. Add a layer to extend functionality.

Add Layers

Select or upload layers to attach to your function.

Search layers...

Create

axios

v1 nodejs20.x 2/1/2024

Cancel

Save Layers

Projects

Functions

Cron Jobs

Layers

APIs

Logs

Settings

User Management

RECENTS

API test

data

axios

1

import axios from 'axios';

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

Test Event

Logs

2025

{

1

"key": "value"

2

}

OUTPUT

Run a test to see output

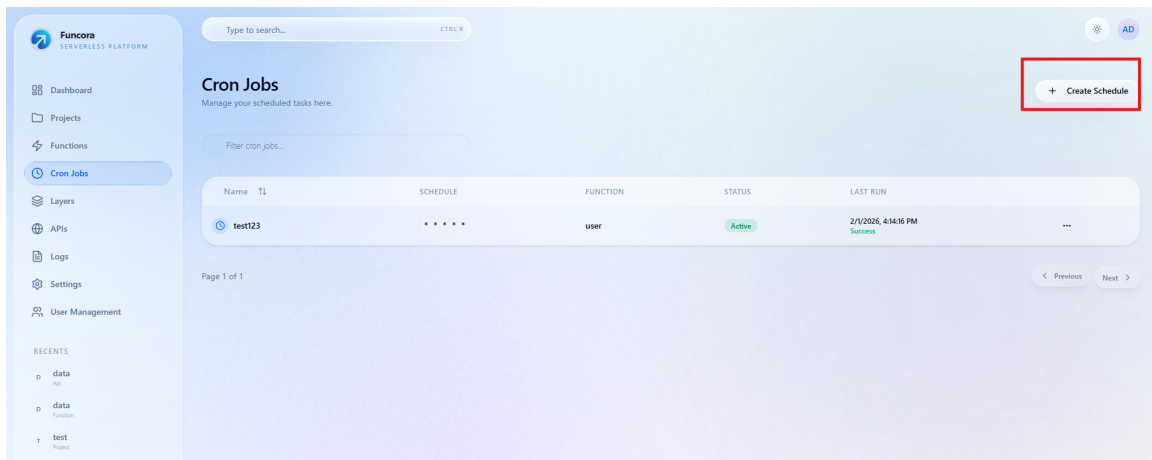
LAYERS (1/5)

+ Add Layer

1 axios 1 nodejs20.x

7. Scheduled Tasks (Cron Jobs)

Automate function execution using Cron Jobs. This is ideal for periodic tasks like database cleanup or reporting.



Create a Schedule by defining:

- Name: Identifier for the job.
- Schedule: The timing interval (Cron expression).
- Function: The target function to execute.
- Payload: Optional JSON data to pass to the function.

