

Zappier — Step-by-Step Usage Walkthrough

Version: 0.1.0 · **Last updated:** 2026-07-27

A full guided tour: from starting the server to pricing changes, customer creation, and live API calls. Every step shows the real screen you should see. Companion documents: [USER-MANUAL.md](#) · [DEVELOPER.md](#).

Step 1 — Start the server

```
cd /Users/marchon/zappier
npm install    # first time only
npm run dev
```

Wait for the two “listening” lines:

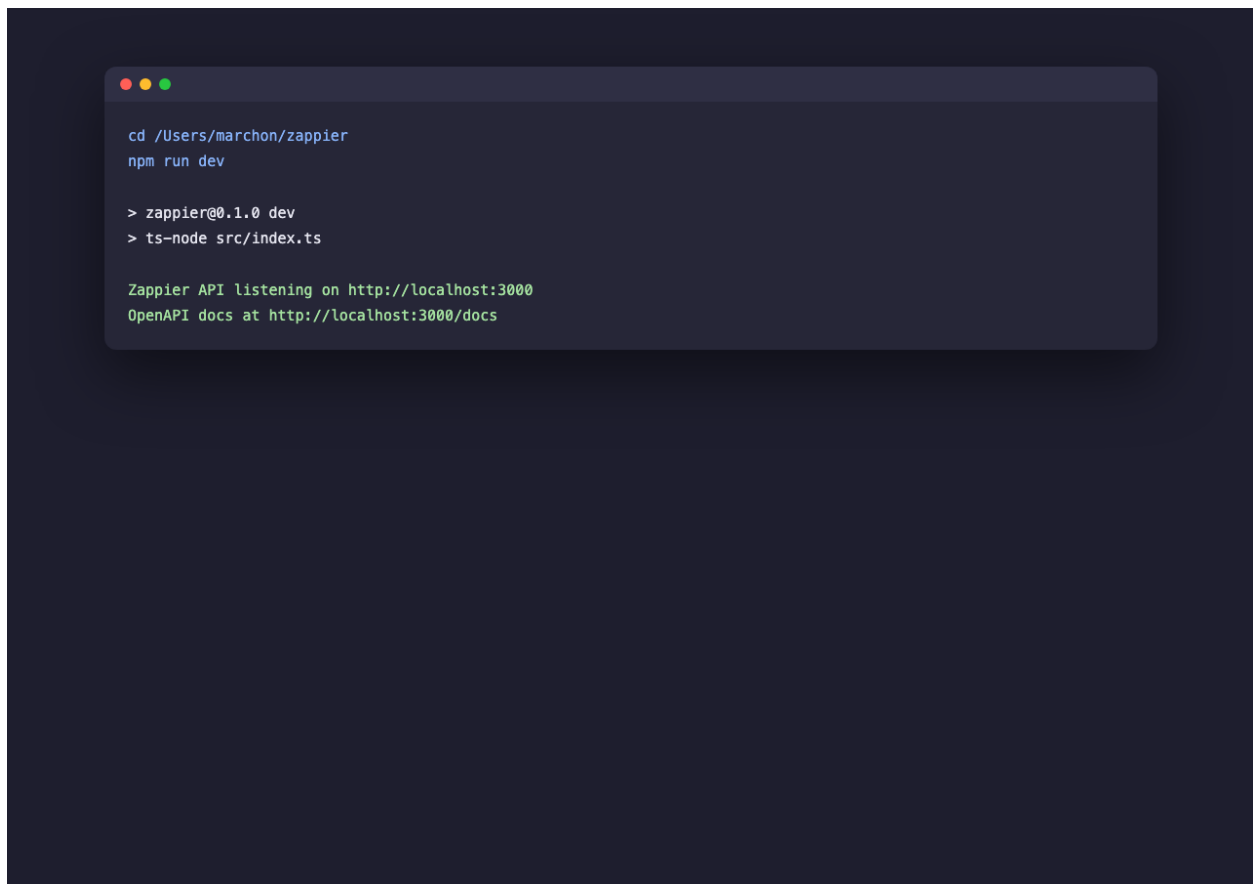


Figure 1: Start the server

Step 2 — Open the admin UI and sign in

Go to <http://localhost:3000/admin>. You’ll see the sign-in screen:

Two accounts are available:

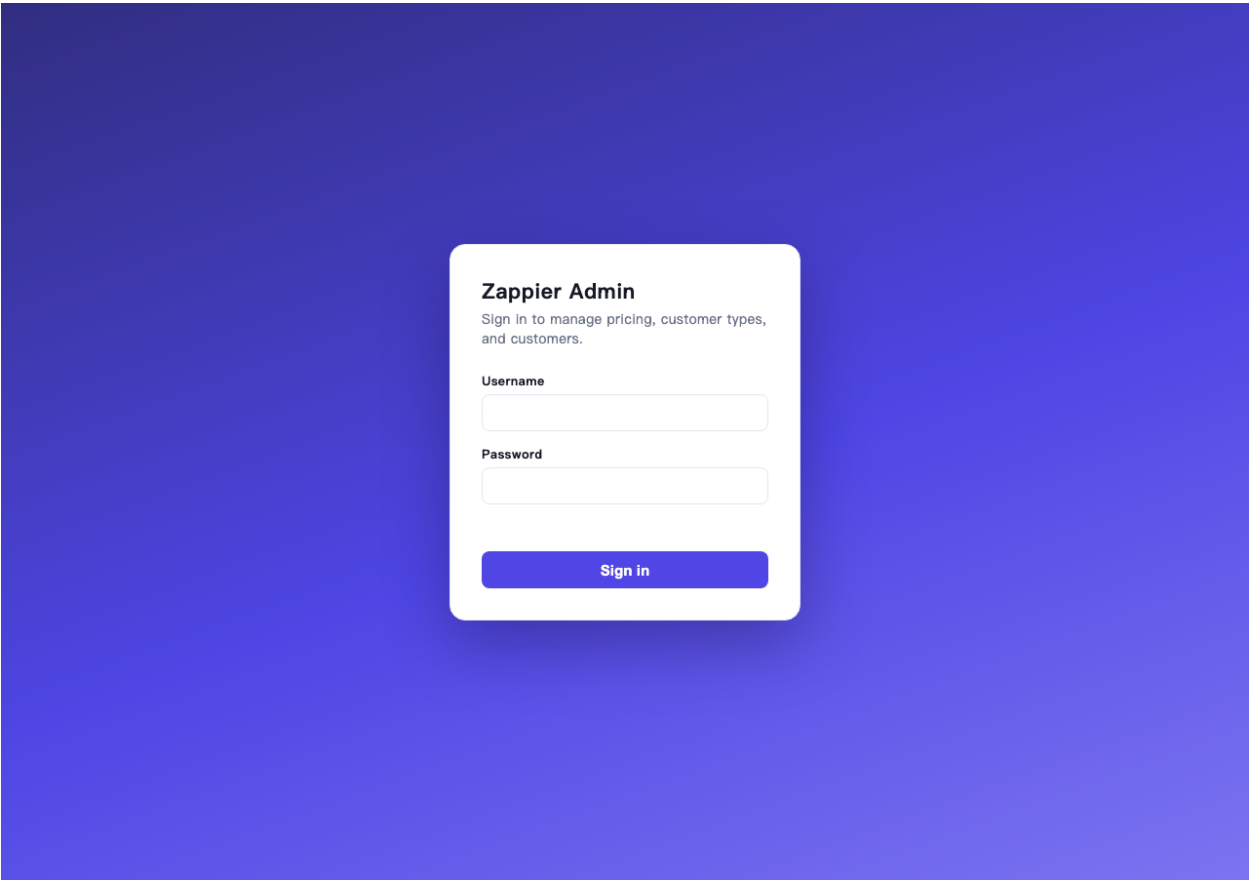


Figure 2: Sign in

Username	Password	Purpose
admin	the ADMIN_KEY env value (default admin-dev-key)	Primary operator
demo	\$\$\$Admin###	Demo / stakeholder access

Override either credential with the ADMIN_USER, ADMIN_KEY, DEMO_ADMIN_USER, and DEMO_ADMIN_PASSWORD environment variables.

Step 3 — The rate card

After sign-in you land on the **Rate card** — one row per API endpoint with its price in cents:

Rate card
Per-endpoint list prices, in cents. Changes apply to the next API call — no restart.

ENDPOINT (OPERATIONID)	KIND	SET KIND	PRICES (CENTS)	
status	free	free		Save Delete
storage-list	free	free		Save Delete
transform	fixed	fixed	fixedCents 4	Save Delete
storage	variable	variable	baseCents 10 perKbCents 1 perMbCents 50	Save Delete

Add endpoint

operationId fixed Add

The operationId must match an operation in openapi.yaml.

Figure 3: Rate card

- status, storage-list — free
- transform — fixed price per call
- storage — variable: base + per-KB metadata + per-MB attachments

Step 4 — Change a price

Edit any cent field — here **transform** is changed from **4¢** to **6¢** — and click **Save**. The change is live on the very next API call; no restart, no deploy.

Rate card

Per-endpoint list prices, in cents. Changes apply to the next API call — no restart.

ENDPOINT (OPERATIONID)	KIND	SET KIND	PRICES (CENTS)		
status	free	free		Save	Delete
storage-list	free	free		Save	Delete
transform	fixed	fixed	fixedCents 6	Save	Delete
storage	variable	variable	baseCents 10 perMbCents 50 perKbCents 1	Save	Delete

Add endpoint

operationId fixed Add

The operationId must match an operation in openapi.yaml.

Sign out

Figure 4: Edit a price

After saving, the table re-reads from the server and shows the new value:

The screenshot shows the Zappier interface for managing rate cards. On the left sidebar, there's a 'Zappier' logo and a 'Rate card' section with sub-items 'Customer types' and 'Customers'. The main area is titled 'Rate card' with a subtitle 'Per-endpoint list prices, in cents. Changes apply to the next API call — no restart.' Below this is a table with columns: ENDPOINT (OPERATIONID), KIND, SET KIND, and PRICES (CENTS). The table has four rows: 'status' (free), 'storage-list' (free), 'transform' (fixed), and 'storage' (variable). Each row has 'Save' and 'Delete' buttons. Below the table is an 'Add endpoint' form with an 'operationId' input, a 'fixed' dropdown, and an 'Add' button. A note below the form says 'The operationId must match an operation in openapi.yaml.' At the bottom left, there's a 'Sign out' button.

ENDPOINT (OPERATIONID)	KIND	SET KIND	PRICES (CENTS)	
status	free	free		<button>Save</button> <button>Delete</button>
storage-list	free	free		<button>Save</button> <button>Delete</button>
transform	fixed	fixed	fixedCents 6	<button>Save</button> <button>Delete</button>
storage	variable	variable	baseCents 10 perKbCents 1 perMbCents 50	<button>Save</button> <button>Delete</button>

Add endpoint

operationId fixed

The operationId must match an operation in openapi.yaml.

[Sign out](#)

Figure 5: Price saved

Step 5 — Customer types

Click **Customer types** in the sidebar. Each type is a pricing tier: a **multiplier** applied to every list price and a **monthly credit** of free included usage (cents).

Step 6 — Add a customer type

Fill the **Add customer type** form — here `edu / Education / ×0.6` — and click **Add**. The new type appears immediately and can be assigned to customers.

New types start with 0 monthly credit — edit the row and **Save** to grant one.

Step 7 — Customers

Click **Customers** in the sidebar. This is where accounts live: their type, and an optional **multiplier override** for per-customer deals (blank = inherit from type).

Customer types

Multiplier scales every list price (0.5 = 50%). Monthly credit is free included usage, in cents.

ID	NAME	MULTIPLIER	MONTHLY CREDIT (CENTS)		
free	Free	1	100	Save	Delete
pro	Pro	0.5	1000	Save	Delete
business	Business	0.25	10000	Save	Delete

Add customer type

multiplier

New types start with 0 monthly credit — edit after adding.

Sign out

Figure 6: Customer types

Customer types

Multiplier scales every list price (0.5 = 50%). Monthly credit is free included usage, in cents.

ID	NAME	MULTIPLIER	MONTHLY CREDIT (CENTS)		
free	Free	1	100	<button>Save</button>	<button>Delete</button>
pro	Pro	0.5	1000	<button>Save</button>	<button>Delete</button>
business	Business	0.25	10000	<button>Save</button>	<button>Delete</button>
edu	Education	0.6	0	<button>Save</button>	<button>Delete</button>

Add customer type

multiplier Add

New types start with 0 monthly credit — edit after adding.

Sign out

Figure 7: Add customer type

- Z

Zapier
- Rate card
- Customer types
- Customers

Customers

Assign types and per-customer deals. A multiplier override replaces the type multiplier for that customer.

ID	NAME	TYPE	MULTIPLIER OVERRIDE	
cust_1	Ada (free)	free	—	Save
cust_2	Grace (pro)	pro	—	Save
cust_3	Linus (business)	business	—	Save

Add customer

free

▼

Create

The new customer's API key is shown once in the notification — copy it immediately.

Sign out

Figure 8: Customers

Step 8 — Create a customer and copy the API key

Enter a name, pick a type, click **Create**. The API key appears **once** in the notification at the bottom-right — copy it and send it to the customer; it is never shown again.

The screenshot shows the Zappier interface. On the left is a sidebar with a 'Zappier' logo and navigation links: 'Rate card', 'Customer types', and 'Customers' (which is highlighted). The main area is titled 'Customers' with a subtitle: 'Assign types and per-customer deals. A multiplier override replaces the type multiplier for that customer.' Below this is a table with columns: ID, NAME, TYPE, MULTIPLIER OVERRIDE, and a Save button. The table contains six rows of customer data. Below the table is an 'Add customer' form with a 'name' input field, a 'free' type dropdown, and a 'Create' button. A notification at the bottom right states: 'Created cust_4cd7a1bc — API key: key-911041798400c80491634be5'. A 'Sign out' button is located at the bottom left of the sidebar.

ID	NAME	TYPE	MULTIPLIER OVERRIDE	
cust_0cd512cd	Marta	pro	—	Save
cust_1	Ada (free)	free	—	Save
cust_2	Grace (pro)	pro	—	Save
cust_3	Linus (business)	business	—	Save
cust_4cd7a1bc	Marta	pro	—	Save
cust_7eba346e	Marta	pro	—	Save

Add customer

name free

The new customer's API key is shown once in the notification — copy it immediately.

Created cust_4cd7a1bc — API key: key-911041798400c80491634be5

Sign out

Figure 9: API key shown once

The new customer appears in the table right away:

Step 9 — Explore the interactive API docs

Open <http://localhost:3000/docs> — full Swagger docs with a “Try it out” console. Click **Authorize** and paste a customer API key to make live calls from the browser.

Step 10 — Make an API call

Call the API with a customer key. Every priced response includes its **quote**, so the cost of every call is transparent:

Note how the quote reflects the walkthrough itself: the 6¢ price set in step 4, halved to 3¢ by Grace’s Pro multiplier.

- Z

Zapier
- Rate card
- Customer types
- Customers

Customers

Assign types and per-customer deals. A multiplier override replaces the type multiplier for that customer.

ID	NAME	TYPE	MULTIPLIER OVERRIDE	
cust_0cd512cd	Marta	pro	—	Save
cust_1	Ada (free)	free	—	Save
cust_2	Grace (pro)	pro	—	Save
cust_3	Linus (business)	business	—	Save

Add customer

name

free

Create

The new customer's API key is shown once in the notification — copy it immediately.

Sign out

Figure 10: Customer created

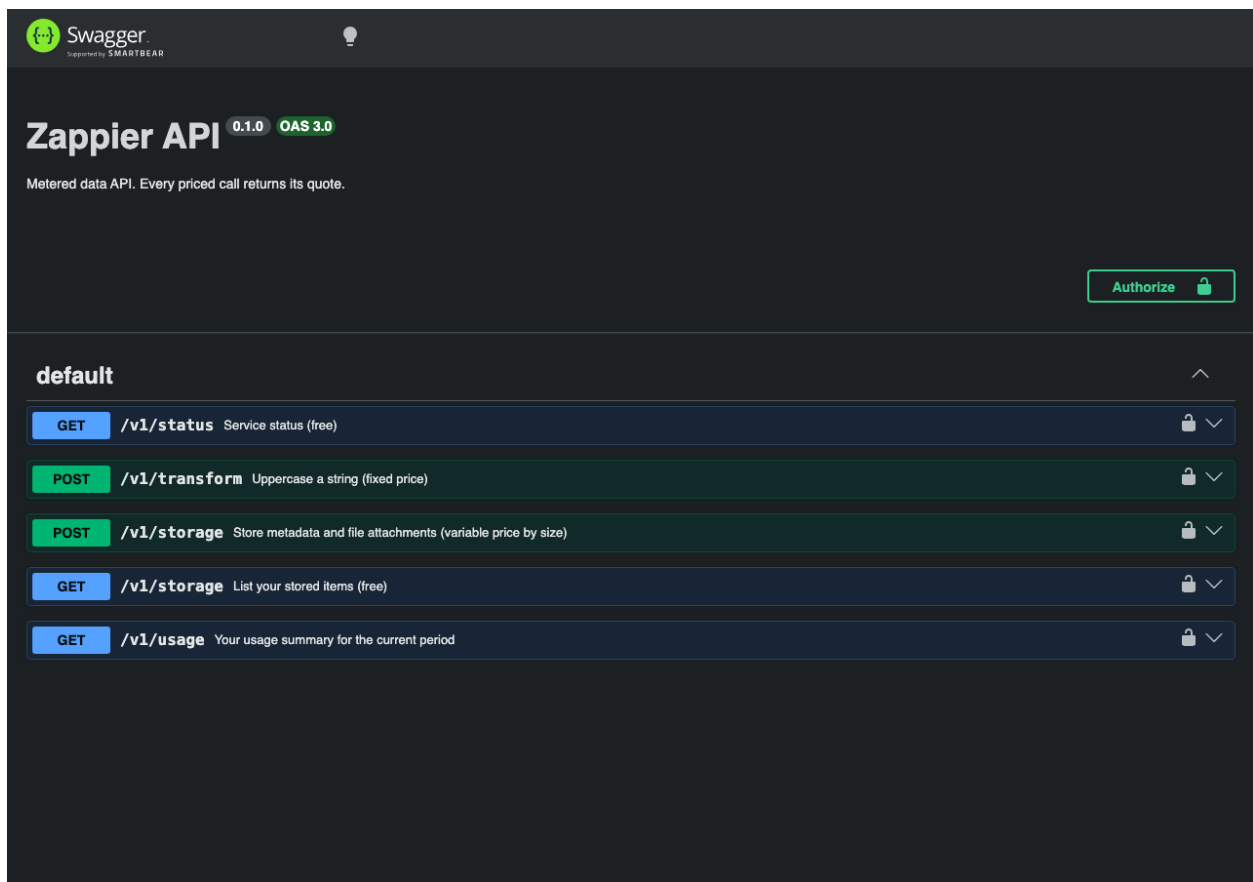


Figure 11: API docs

```
curl -X POST -H 'x-api-key: key-grace' \
-H 'content-type: application/json' \
-d '{"text":"hello zapper"}' \
http://localhost:3000/v1/transform

{
  "output": "HELLO ZAPPIER",
  "quote": {
    "endpointId": "transform",
    "listCents": 6,           ← the 6¢ price we set in step 4
    "totalCents": 3,        ← Grace is Pro: 6 × 0.5
    "breakdown": { "baseCents": 6, "metadataCents": 0, "attachmentCents": 0 }
  }
}
```

Figure 12: API call with quote

Step 11 — Check usage and credits

Customers can check their own month-to-date usage anytime:

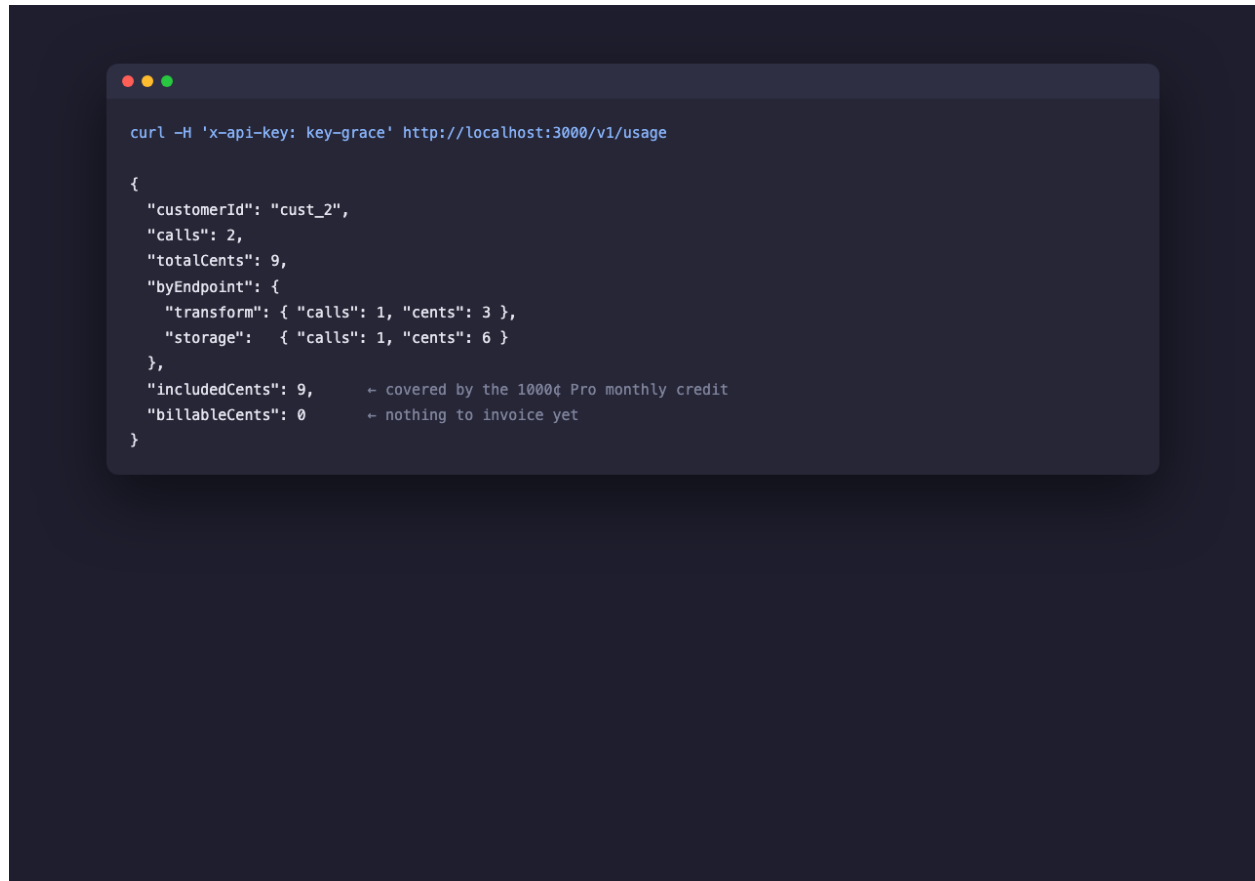


Figure 13: Usage summary

`includedCents` is covered by the type's monthly credit; `billableCents` is what the daily billing job would report to Stripe right now.

Where to go next

- **Daily billing** runs automatically at 06:17 ET — see `USER-MANUAL.md` §6 for the Stripe meter/product/price setup and how to verify a run.
- **Troubleshooting:** `USER-MANUAL.md` §9.
- **Internals:** `DEVELOPER.md`.