

A dark blue map of the world is shown in the background. The text is overlaid in a bright yellow color. The text is arranged in two lines, with the first line being 'Using cartoframes' and the second line being 'with GeoDB instances'. The text is in a bold, sans-serif font.

**Using cartoframes
with GeoDB instances**

Authentication

To use `cartoframes`, you need to authenticate with your GeoDB instance. For this you need the following information:

- Your master API key.
- Your username.
- Your GeoDB server instance's URL.

Find your API key

The screenshot shows the 'Your API keys' page in the Google Cloud Platform console. The page has a blue header with navigation links: Home, Maps, and Data. A search bar and the Google logo are on the right. On the left, there's a sidebar with links: Profile, Account, Connected Apps, Organization settings, and Developer Settings. The main content area is titled 'The simplest way to access your private data'. Under 'Default API Keys', there's a 'Master' key highlighted with a red box. The key is partially obscured by a redaction box, with the visible part being '...d658deab65d3b25e0'. A red arrow points from the text 'Copy master API Key or create a custom one for a given dataset' to this key. To the right of the key is a 'Regenerate' link. Below the key is a warning icon and text: 'For testing and development only! This API key can be used to perform any API request without restriction!'. Under 'Default public', there's a 'default_public' key with a redaction box. Below this, there's a 'Custom API Keys' section with a 'NEW API KEY' button. At the bottom, there's a table of custom API keys with columns for key name, scope (MAPS, SQL), and actions (Regenerate, Delete). One key is visible with the name 'events_api' and scope 'SQL'.

Home Maps Data Search

Profile
Account
Connected Apps
Organization settings
Developer Settings

The simplest way to access your private data

Default API Keys

Master
[Redacted] d658deab65d3b25e0 [Copy] Regenerate
⚠ For testing and development only! This API key can be used to perform any API request without restriction!

Default public
default_public [Copy]

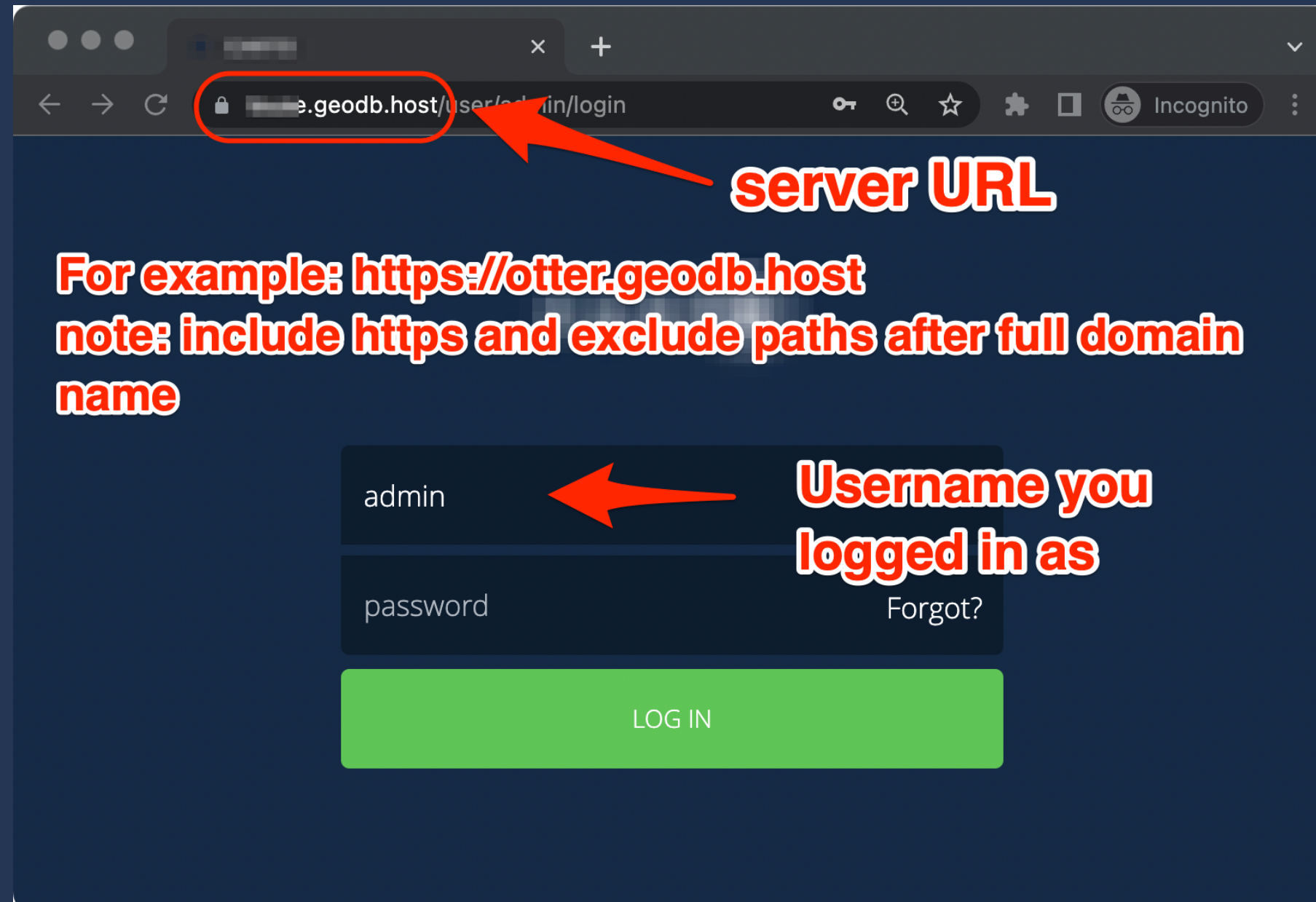
Custom API Keys

NEW API KEY

[Redacted] [MAPS] [SQL] [Copy] Regenerate Delete
[Redacted] [Copy]
[Redacted] [events_api] [SQL]

Copy master API Key or create a custom one for a given dataset

Find your username and server url



The image shows a web browser window with the address bar containing the URL `https://[redacted].geodb.host/user/admin/login`. A red circle highlights the domain part of the URL, and a red arrow points from the text **server URL** to it. Below the browser window, there is a login form with two input fields: the first contains the text `admin` and the second contains the text `password`. A red arrow points from the text **Username you logged in as** to the `admin` text in the first input field. To the right of the password field is a link labeled `Forgot?`. At the bottom of the form is a green button labeled `LOG IN`.

server URL

For example: `https://otter.geodb.host`
note: include https and exclude paths after full domain name

Username you logged in as

admin

password

Forgot?

LOG IN

Create a .env file

Create a .env file in your project's root directory and add your API key, username and server url. For example:

```
cat << EOF > .env
GEODB_MASTER_API_KEY=YOUR_API_KEY
GEODB_USERNAME=YOUR_USERNAME
GEODB_SERVER_URI=YOUR_SERVER_URL
EOF
```

Set credentials in your code

```
import os
from dotenv import load_dotenv
from cartoframes.auth import set_default_credentials
from cartoframes.auth import Credentials

load_dotenv()
GEODB_USERNAME=os.getenv( 'GEODB_USERNAME' )
GEODB_MASTER_API_KEY=os.getenv( 'GEODB_MASTER_API_KEY' )
GEODB_SERVER_URI=os.getenv( 'GEODB_SERVER_URI' )
auth_credentials = Credentials(
    base_url=f"{GEODB_SERVER_URI}/user/{GEODB_USERNAME}",
    api_key=GEODB_MASTER_API_KEY
)
set_default_credentials(auth_credentials)
```

Creating a dataset

We will create a geopandas dataframe from a geojson dataset and then upload it to GeoDB:

```
from cartoframes import to_carto
from geopandas import read_file

gdf = read_file('https://bit.ly/3Dc84I7')
gdf = gdf.drop(['cartodb_id'], axis=1)
to_carto(gdf, 'starbucks_brooklyn')
# Success! Data uploaded to table "starbucks_brooklyn" correctly
# 'starbucks_brooklyn'
```



Home



Maps



Data



Search



Your Datasets

Your Connections




Spatial Data Catalog



Your Datasets



New Dataset

Name ▾	Last Modified	Rows	Size	Builder Usage	Privacy
 starbucks_brooklyn	Updated 2 minutes ago	10	24 kB	0 maps	Private

Append data to an existing dataset

```
from cartoframes import to_carto
from geopandas import read_file

gdf = read_file('https://bit.ly/3Dc84I7')
gdf = gdf.drop(['cartodb_id'], axis=1)
# note that we are using the `if_exists` parameter
to_carto(gdf, 'starbucks_brooklyn', if_exists='append')
# Success! Data uploaded to table "starbucks_brooklyn" correctly
# 'starbucks_brooklyn'
```

starbucks_brooklyn ⋮

PRIVATE

ADD PEOPLE Updated 16 minutes ago

cartodb_id ↑ ⋮ number	the_geom ⋮ geometry	field_1 ⋮ number	name string
12	-73.96122, 40.57796	1	607 Brighton Bea
13	-73.98976, 40.61912	2	65th St & 18th Av
14	-74.02744, 40.63152	3	Bay Ridge Pkwy &
15	-74.00098, 40.59321	4	Caesar's Bay Sho
16	-73.99261, 40.68849	5	Court St & Dean
17	-73.87015, 40.65315	6	Target Gateway T
18	-74.03313, 40.61927	7	3rd Ave & 92nd S
19	-73.98452, 40.6915	8	Lam Group @ Sh
20	-73.94841, 40.63207 ⋮	9	33-42 Hillel Place



appended 10 records

Replace data in an existing dataset

```
from cartoframes import to_carto
from geopandas import read_file

gdf = read_file('https://bit.ly/3Dc84I7')
gdf = gdf.drop(['cartodb_id'], axis=1)
# note that we are using `if_exists='replace'`
to_carto(gdf, 'starbucks_brooklyn', if_exists='replace')
# Success! Data uploaded to table "starbucks_brooklyn" correctly
# 'starbucks_brooklyn'
```

Delete a dataset

```
from cartoframes import delete_table

delete_table('starbucks_brooklyn')
# Success! Table "starbucks_brooklyn" removed correctly
```