



Exporting your users with the API

ON THIS PAGE

Prerequisites

- Zendesk account

- Zendesk API token

- Python

- Terminal

- Setting up your project

- Create the export script

- Export the users

- Customizing your CSV file

The [reporting tools](#) in Zendesk can export your users in CSV, JSON, or XML formats. However, these tools are not available to customers on Team plans.

As an alternative, customers on all Zendesk plans including Team plans can use the [Zendesk REST API](#) to export lists of their users.

In this how-to article, you'll create a small Python script that exports all the users in your account to a CSV file. You can then import the CSV file into your favorite spreadsheet application.

If you're uncomfortable working with code, you may want to ask for help from someone with greater technical proficiency or from your favorite AI chat bot.

Disclaimer: Zendesk provides this article for instructional purposes only. Zendesk doesn't provide support for the example code in this article. Zendesk doesn't support third-party technologies such as command-line tools or Python.

Prerequisites

Zendesk account

Ensure you have access to a Zendesk account and that you've been assigned the role of admin or agent in the account.

Zendesk API token

1. Sign in to your Zendesk account.
2. Navigate to the **Admin Center**.
3. Go to **Apps and integrations > APIs > Zendesk API**.
4. If not already done, click the **Settings** tab and enable **Token Access**.
5. Click **Add API Token**, give it a name, and save the token somewhere secure (you'll need it later).

Python

Python is a high-level, general-purpose programming language. It's one of the easiest languages to learn and use.

Download and install the latest version of Python for your operating system at <https://www.python.org/downloads/>.

Terminal

Terminal is a command line tool that's already installed in Windows and macOS.

Windows

1. Click the Start Menu (Windows icon) in the taskbar or press the Windows key on your keyboard.
2. Type "Terminal" in the search bar.
3. Right-click **Terminal** in the search results and choose **Run as administrator**.
4. Allow Terminal to run scripts by entering `Set-ExecutionPolicy -ExecutionPolicy Unrestricted` at the command line and pressing Enter.

You only have to run Terminal as an administrator and set the execution policy one time.

If you use Windows 10 or earlier, you can use the command prompt instead. Press **Windows + R** to open the Run dialog, then type `cmd` and press Enter.

macOS

1. Open Finder and go to Applications/Utilities.
2. Double-click Terminal.

Setting up your project

1. Create a folder called **export_users**.
2. Navigate to the folder using Terminal.

For example, if you created your **export_users** folder in the Documents/projects folder, then run one of the following commands in Terminal and press Enter:

- Windows: `cd Documents\projects\export_users`
- Mac or Linux: `cd Documents/projects/export_users`

The command `cd` means "change directory".

3. Create a virtual environment by entering one of the following commands in Terminal:

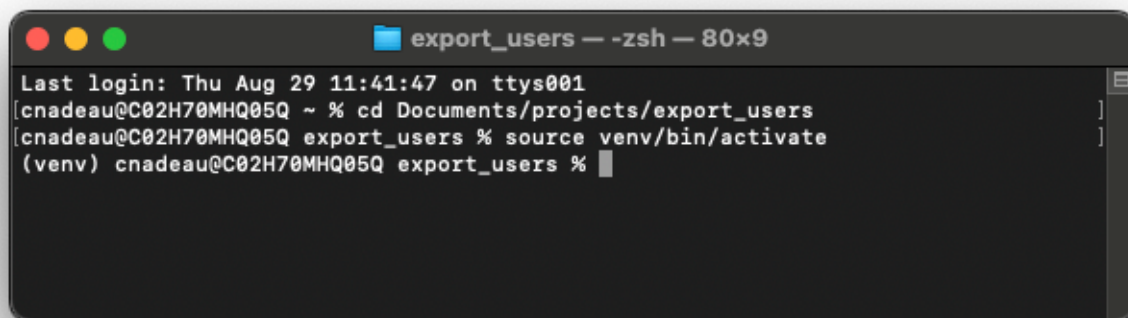
- Windows: `py -m venv venv`
- macOS or Linux: `python3 -m venv venv`

A virtual environment allows you to install Python libraries just for this project so that you don't clutter the rest of your system with multiple conflicting Python libraries.

4. Activate the virtual environment by running one of the following commands:

- Windows: `venv\Scripts\activate`
- Mac or Linux: `source venv/bin/activate`

You should now see "(venv)" at the start of the line in Terminal.



```
export_users — zsh — 80x9
Last login: Thu Aug 29 11:41:47 on ttys001
[cnadeau@C02H70MHQ05Q ~ % cd Documents/projects/export_users
[cnadeau@C02H70MHQ05Q export_users % source venv/bin/activate
(venv) cnadeau@C02H70MHQ05Q export_users %
```

5. Install the requests library by running the following command:

```
python -m pip install requests
```

The requests library is a Python library that makes API requests easier.

Create the export script

1. Create a file named `export_users.py`.
2. Paste the following code into the file:

```
1  import csv
2  import time
3  import requests
4
5
6  ZENDESK_USER_EMAIL = 'your_zendesk_email'
7  ZENDESK_API_TOKEN = 'your_zendesk_api_token'
8  ZENDESK_SUBDOMAIN = 'your_zendesk_subdomain'
9
10 def main():
11     print('Getting users from Zendesk...')
12     users: list = []
13     url = f'https://{ZENDESK_SUBDOMAIN}.zendesk.com/api/v2/users.json'
14     params = {'page[size]': 100}
15     auth = f'{ZENDESK_USER_EMAIL}/token', ZENDESK_API_TOKEN
16     response = requests.get(url, params=params, auth=auth)
17     page = response.json()
18     users.extend(page['users'])
19     while page['meta']['has_more']:
20         params['page[after]'] = page['meta']['after_cursor']
21         response = requests.get(url, params=params, auth=auth)
22         if response.status_code == 429: # if rate limit reached
23             time.sleep(int(response.headers['retry-after']))
24             response = requests.get(url, auth=auth)
25         page = response.json()
26         users.extend(page['users'])
27
28     write_to_csv(users)
29
30     print('Done!')
31
32
33 def write_to_csv(users):
34     with open('users.csv', 'w', newline='') as file:
35         writer = csv.writer(file)
36         writer.writerow([
37             'Name',
38             'Email'
39         ])
40         for user in users:
41             writer.writerow([
42                 user['name'],
```

```
43         user['email']
44     ])
45
46
47 if __name__ == "__main__":
48     main()
```

Make sure to preserve the code's indentation, which matters in Python.

3. Edit the file to change the values of the three constant variables at the top of the file (the ones in upper case) to values that apply to you.

Export the users

1. If not already done, navigate to the project folder using Terminal:

- Windows: `cd Documents\projects\export_users`
- Mac or Linux: `cd Documents/projects/export_users`

2. If not already done, activate the virtual environment:

You should see "(venv)" at the start of the line in Terminal. If not, activate the virtual environment:

- Windows: `venv\Scripts\activate`
- Mac or Linux: `source venv/bin/activate`

3. Export your users by running the following command:

```
python export_users.py
```

The script will create a **users.csv** file in your project folder.

4. Import the CSV file into your favorite spreadsheet application.

Customizing your CSV file

You can add more columns to the CSV file by adding items to the two lists in the `write_to_csv()` function definition. The first list defines the heading row in your CSV file. The second list defines the rest of the rows in the file.

The number and order of the items in the two lists must match exactly to correctly display the data in your CSV file.

The following example adds a CSV column for the user 'role':

```
1 def write_to_csv(users):
2     with open('users.csv', 'w', newline='') as file:
3         writer = csv.writer(file)
4         writer.writerow([
5             'Name',
6             'Email',
7             'Role'
8         ])
9     for user in users:
10        writer.writerow([
11            user['name'],
12            user['email'],
13            user['role']
14        ])
```

Each item in each list must end with a comma except for the last list item.

The terms 'name', 'email', and 'role' are defined in the API reference docs. They correspond to possible columns in your users table. For all possible columns you can add to your CSV file, see the following reference in the Users API docs:

<https://developer.zendesk.com/api-reference/ticketing/users/users/#json-format>

Join our developer community

 [Forum](#)  [Blog](#)  [Slack](#)

Zendesk 181 Fremont Street, 17th Floor, San Francisco, California 94105

[Privacy Policy](#) [Terms & Conditions](#) [System Status](#) [Cookie Settings](#)