# List of available scripts

# Guide: Bulk User Invitations to ActivityInfo via Python Script

## 1. Introduction

This document provides a comprehensive guide on how to use the provided Python script to automate the process of adding multiple users to an ActivityInfo database. The script reads user information (name and email) from a CSV file and invites them to the specified database with a pre-defined role.

## 2. Prerequisites

Before you begin, ensure you have the following:

* **Python 3:** The script is written in Python. If you don't have it installed, download it from [python.org](https://www.python.org/downloads/).
* **Requests Library:** The script uses the requests library to communicate with the ActivityInfo API.
* **ActivityInfo Account:** You need an account with permission to add users to the target database.
* **API Token:** A personal API token to authenticate your requests.
* **User Data File:** A CSV file containing the names and email addresses of the users you want to add.

## 3. Setup Instructions

Follow these steps to set up and configure the script for use.

**Step 1: Install the requests Library**

If you don't have the requests library installed, open your terminal or command prompt and run the following command:

Bash

pip install requests

**Step 2: Prepare the User CSV File**

You need a CSV file with the user data. The script is configured to look for a file named users.csv.

* The CSV file **must** have a header row.
* The first column must be the user's **name**.
* The second column must be the user's **email**.

**Example users.csv file:**

Code snippet

name,email

John Doe,john.doe@example.org

Jane Smith,jane.smith@example.com

Peter Jones,peter.jones@sample.net

**Step 3: Save the Script**

Save the provided Python code into a file. For example, you can name it add\_users.py. **Place this file in the same directory as your users.csv file.**

**Step 4: Configure the Script**

Open the add\_users.py file in a text editor and modify the variables in the CONFIGURATION section at the top.

* API\_TOKEN: This is your personal access token for the ActivityInfo API.
  + **How to find it:** Log in to ActivityInfo, click your profile icon in the top right, go to "User Settings", and select the "API Tokens" tab to generate a new token.
  + ⚠️ **Important:** Treat your API Token like a password. Do not share it publicly.

Python

API\_TOKEN = "XXXXXXXXXXXXXX" # Replace with your actual token

* DATABASE\_ID: This is the unique identifier for the database you want to add users to.
  + **How to find it:** Navigate to your database's homepage. The ID is the string of letters and numbers in the URL. For example, in https://www.activityinfo.org/app/#/database/cqjj5vemfv8zsji3r, the ID is cqjj5vemfv8zsji3r.

Python

DATABASE\_ID = "cqjj5vemfv8zsji3r" # Replace with your database ID

* CM\_ADM\_ROLE\_ID: This is the identifier for the role you want to assign to the new users.
  + **How to find it:** Press F12 to open the developer window on the browser and then, go to "Database settings" > "User management" > "User management" in ActivityInfo. Find a user with the role you want to assign and check the role id. For example, *c2klv6bmewnkere2*.
  + **Note:** The script is currently set to use this CM\_ADM\_ROLE\_ID. Even though a READONLY\_ROLE\_ID variable exists, the code uses CM\_ADM\_ROLE\_ID when adding users. Change this value to the ID of your desired role.

Python

CM\_ADM\_ROLE\_ID = "c2klv6bmewnkere2" # Replace with your desired role ID

* USERS\_CSV\_PATH: The name of your CSV file. If you named your file something other than users.csv, update this variable.

Python

USERS\_CSV\_PATH = "users.csv" # Change if your CSV has a different name

## 4. Running the Script

Once you have completed the setup, you can run the script.

1. Open your terminal or command prompt.
2. Navigate to the directory where you saved the add\_users.py and users.csv files.
3. Run the script using the following command:

Bash

python add\_users.py

## 5. Understanding the Output

The script will print its progress in the terminal.

* **User Already Exists:** If a user from the CSV is already in the database, you will see:
* User john.doe@example.org already in database — skipping.
* **Successful Invitation:** If a user is successfully added, you will see:
* SUCCESS: jane.smith@example.com added as readonly
* **Error:** If the script fails to add a user, an error message will be displayed with details:
* ERROR: Could not add peter.jones@sample.net — status 400: {"message":"Invalid email address"}

This allows you to monitor the process and identify any issues that may arise.