

API DOCUMENTATION

# Getting started

In this guide you can learn the basics of using the ACLED API to access ACLED data.

## On this page

## Accessing the API

### Login/Register

Access to ACLED's API requires you to set up a myACLED account. Please visit the registration page to set yours up, if you haven't already.

Once you have an account set up, you can authenticate your API request in one of two ways:

- Cookie-based authentication (simple browser or tool-based access)
- OAuth token-based authentication (for programmatic access)

### Setting up an API call & building a URL

To execute an API call (i.e. to request data), you must first request an authentication token, using your myACLED credentials. You can then build a URL, including various parameters defining the subset of the data you want to view. You can use this URL to request data via your internet browser or through programming languages such as R or Python, a step which is covered in the individual endpoint tabs (e.g. ACLED endpoint section).

### Authentication - Who is requesting the data?

The authentication method you choose will depend on how exactly you want to access the data.

#### Cookie-based authentication

To view the data directly in your browser, you can take the following steps:

- Log in to <https://acleddata.com>
- Navigate to any API page in the browser, e.g. <https://acleddata.com/api/acled/read?limit=10>
- Response will be displayed directly (JSON by default)

If you are using an external tool, such as Postman, you can also use cookie-based authentication. The steps are as follows:

#### Overview

- Make an authentication with your user credentials
- Session is now logged in e.g. Postman
- Make subsequent calls to the API endpoint, e.g. <https://acleddata.com/api/acled/read?limit=10>

#### Detail

- Make authentication request

Make a POST request to <https://acleddata.com/user/login?format=json> with a JSON payload (in the body) of {'name': 'email address', 'pass': 'password'}

Example:

```
{
  "name": "EMAIL-HERE",
  "pass": "PASSWORD-HERE"
}
```

Example JSON response:

```
{
  "current_user": {
    "uid": "NUMERIC-ID-HERE",
    "name": "EMAIL-HERE"
  },
  "csrf_token": "TOKEN-HERE",
  "logout_token": "LOGOUT-TOKEN-HERE"
}
```

- Session is now logged in e.g. Postman

Your session is now created in the tool you are using, and you do not need to explicitly pass any authentication token in your API requests.

- Make subsequent calls to the API endpoint

Make a GET request to e.g. <https://acleddata.com/api/acled/read?limit=10>, and the relevant data will be returned.

#### OAuth method

If you are using a script to access the data via the API programmatically, you will need to authenticate via OAuth. The steps are as follows:

#### Overview:

- Request an Access Token
- Use the Access Token in your API request
- (Optional) Refresh when needed

#### Detail:

- Request an Access Token

Make a POST request to our authentication endpoint with your credentials, to receive an access token valid for 24 hours and a refresh token valid for 14 days. Use the following variables:

Variable name	Value
username	The user's email address
password	The user's password
grant_type	"password" ↔ hard-coded string
client_id	"acled" ↔ hard-coded string

Example CURL request:

```
curl -X POST "https://acleddata.com/oauth/token" \
-H "Content-Type: application/x-www-form-urlencoded" \
-d "username=SMALL.DOMAIN.COM" \
-d "password=YOUR_PASSWORD" \
-d "grant_type=password" \
-d "client_id=acled"
```

Example response (JSON):

```
{
  "token_type": "Bearer",
  "expires_in": 86400,
  "access_token": "ACCESS-TOKEN-HERE",
  "refresh_token": "REFRESH-TOKEN-HERE"
}
```

- Use the Access Token in your API request

Include your access token in the Authorization header of your API request. Format:

Header name	Value
Authorization	Bearer [INSERT TOKEN HERE]

Example request:

```
curl -H "Authorization: Bearer ACCESS-TOKEN-HERE" \
-X GET \
"https://acleddata.com/api/acled/read?limit=10"
```

- Refresh when needed

When your access token expires, you can use your refresh token to get a new one. This avoids having to re-enter your credentials.

Example request:

```
curl -X POST "https://acleddata.com/oauth/token" \
-H "Content-Type: application/x-www-form-urlencoded" \
-d "refresh_token=YOUR_REFRESH_TOKEN" \
-d "grant_type=refresh_token" \
-d "client_id=acled"
```

Example response:

```
{
  "token_type": "Bearer",
  "expires_in": 86400,
  "access_token": "ACCESS-TOKEN-HERE",
  "refresh_token": "REFRESH-TOKEN-HERE"
}
```

## The base URL - From whom are you requesting data?

An API call is simply a request sent from one computer to another. Accordingly, the first step in building the URL that you will use to execute your API call is specifying where you would like to send your request. To access ACLED data, your computer must send a request to ACLED's server, which you can specify using the ACLED API's base URL:

<https://acleddata.com/api/>

Every request you make to the ACLED API will start with this base URL.

## The endpoint - Which ACLED dataset would you like to access?

After stating that you would like to request data from ACLED's server, you must also specify which data (or "endpoint") you are requesting. In practical terms, which ACLED dataset are you interested in accessing? Available endpoints/datasets include:

- ACLED
- Deleted
- CAST

Be sure to click through ACLED's endpoint links to understand your options. If you want to access the core dataset of political violence, demonstration, and strategic development events, then you should use the ACLED endpoint.

Once you have selected an endpoint, you can add it to the base URL. For example, if you are requesting data from the ACLED endpoint, your URL should now look like this:

<https://acleddata.com/api/acled/>

## The response format - How would you like your data to be formatted?

An important element to consider when setting up your API call is the format of the data you want to receive. As covered in the Advanced concepts section, there are four file types which you can request: JSON, CSV, and XML.

You can specify your desired data format by adding read?format + {the extension of the format you want}. The only exception is JSON format, which is the default option, and therefore only requires you to write read?

For example, if you want your data in CSV format, your URL should now appear as:

<https://acleddata.com/api/acled/read?format=csv>

## Query filters - Which data do you want?

The URL you have built will now return data. However, the ACLED API has a default row limit of 5000, meaning this URL will execute an API call that only returns the first 5000 rows of the requested dataset. (For more information on this, see the Limits section below)

In almost all cases, users only need a subset of the entire dataset.

Instead of arbitrarily requesting the first 5000 rows of data, you can use query filters to specify which data you want to receive.

For instance, imagine you are interested in investigating events in the country of Georgia between 2022-01-01 and 2023-02-01. Rather than downloading the entire dataset and then finding the relevant events within it, you can use query filters to request only the specific events that meet your criteria. In this case, you should use the event\_date and country query filters. You can learn the basics of applying query filters in the examples below, but for a more in-depth guide, visit our ACLED endpoint section.

You can start by applying a query filter to specify the country of interest. The country query filter is an "=" type, which will match your request to an exact value in the dataset. For instance, if you include country=Georgia in your URL, you will receive all events in which the country column exactly matches 'Georgia'.

<https://acleddata.com/api/acled/read?format=csv&country=Georgia>

Next, you can specify your desired date range. You can specify a particular date by using the event\_date filter, which is also of "=" type. For example, if you include ...event\_date=2023-02-01 in your URL, you will receive a dataset containing events that occurred on the 1st of February 2023.

In all likelihood, rather than requesting events from a specific date, you will want to request events occurring across a date range. You can do so by changing the query type. Specifically, you can add the where suffix to your query filter and then select the type of query you would like to use (e.g., BETWEEN allows you to filter for anything between two values separated by ). See [Advanced concepts section](#) for more details on building complex queries.

Your URL is now requesting only the desired and specific subset of ACLED data:

[https://acleddata.com/api/acled/read?format=csv&country=Georgia&event\\_date=2022-01-01|2023-02-01&event\\_where=BETWEEN](https://acleddata.com/api/acled/read?format=csv&country=Georgia&event_date=2022-01-01|2023-02-01&event_where=BETWEEN)

Remember: If you have more than one query filter (e.g. one for country and one for event\_date), they should be separated by the & symbol.

## Limits

Your URL is now usable. It contains the base URL, the endpoint, your authentication credentials, and appropriate query filters. However, even when using query filters, the returned dataset may be very large. If you do not need every event defined by your query filters, you can limit the size of the returned file. In particular, you can specify the number of events in the returned data by including a limit statement in your URL (e.g., limit=2000 means you will receive data for 2000 events). Without a 'limit' in your URL, ACLED's API will return a maximum of 5000 events. In cases where you are expecting events beyond the default limit, you are advised to use pagination which allows you to split your call into multiple smaller ones and ensure your call executes successfully. These calls do not count toward your API rate limits. For more on pagination and limits, see the [Elements of ACLED's API](#).

In the example you have been working through, you are requesting data from a very limited date range in a country with relatively few events. You therefore might only expect to receive several hundred rows of data - far fewer than ACLED's 5000 row default. You probably do not need to include a limit for such a restricted API call, but if desired you could add a reasonable limit to your URL:

[https://acleddata.com/api/acled/read?format=csv&country=Georgia&event\\_date=2022-01-01|2023-02-01&event\\_where=BETWEEN&limit=3000](https://acleddata.com/api/acled/read?format=csv&country=Georgia&event_date=2022-01-01|2023-02-01&event_where=BETWEEN&limit=3000)

## Executing the call

Congratulations - you have built your first URL to request ACLED data! You can now execute your API call and receive specific data from ACLED's API. The easiest way to execute your API call is to simply copy and paste the URL you have built into your browser. If your call is successful, your data will start downloading in the specified file format.

You can now examine the data you received (for the sake of better readability, we removed the notes column in this example, and reduced the number of rows):

event_id_cnty	event_date	year	time_precision	disorder_type	event_type	sub_event_type	actor1	as
GEO3998	2023-02-01	2023	1	Demonstrations	Protests	Peaceful protest	Protesters (Georgia)	La (G)
GEO3999	2023-02-01	2023	1	Demonstrations	Protests	Peaceful protest	Protesters (Georgia)	La (G) W (G)
GEO4002	2023-02-01	2023	1	Demonstrations	Protests	Peaceful protest	Protesters (Georgia)	
GEO4001	2023-01-31	2023	1	Demonstrations	Protests	Peaceful protest	Protesters (Georgia)	
GEO4000	2023-01-29	2023	1	Demonstrations	Protests	Peaceful protest	Protesters (Georgia)	
GEO3997	2023-01-26	2023	1	Demonstrations	Protests	Peaceful protest	Protesters (Georgia)	
GEO3994	2023-01-24	2023	1	Demonstrations	Protests	Peaceful protest	Protesters (Georgia)	Pn (R) Pn
GEO3996	2023-01-22	2023	1	Demonstrations	Protests	Peaceful protest	Protesters (Georgia)	Pn (U)
GEO3995	2023-01-22	2023	1	Demonstrations	Protests	Peaceful protest	Protesters (Georgia)	Pn (U)
GEO3993	2023-01-21	2023	1	Demonstrations	Protests	Peaceful protest	Protesters (Georgia)	Pn (R)

Congratulations! You have successfully built and executed an API call for ACLED's API! You are encouraged to visit the endpoint sections on this website (see links at the top of this [section](#)). There you will find walkthroughs, details, and examples of different tools for every endpoint in ACLED's API.

## Examples in Python

```
import requests
import json

# Function to get access token using username and password
def get_access_token(username, password, token_url):
    headers = {
        'Content-Type': 'application/x-www-form-urlencoded',
    }
    data = {
        'username': username,
        'password': password,
        'grant_type': 'password',
        'client_id': "acled"
    }

    response = requests.post(token_url, headers=headers, data=data)

    if response.status_code == 200:
        token_data = response.json()
        return token_data['access_token']
    else:
        raise Exception(f"Failed to get access token: {response.status_code} {response.text}")

# Get an access token
my_token = get_access_token(
    username="your_email@las1.com",
    password="your_password",
    token_url="https://acleddata.com/oauth/token",
)

# Option #1 (parameters in the url)
base_url = "https://acleddata.com/api/acled/read?format=json&country=Georgia:OR:country=Armenia&year=2"
# request base url with my_token
response = requests.get(
    base_url,
    headers={"Authorization": f"Bearer {my_token}", "Content-Type": "application/json"},
)

if response.json()["status"] == 200:
    print(
        "Request successful"
    )

# Option #2 (parameters as a dictionary)
parameters = {
    "country": "Georgia:OR:country=Armenia:OR:country=Azerbaijan",
    "year": 2021,
    "fields": "event_id_cnty|event_date|event_type|country|fatalities",
}

response_params_dic = requests.get(
    "https://acleddata.com/api/acled/read?format=json",
    params=parameters,
    headers={"Authorization": f"Bearer {my_token}", "Content-Type": "application/json"},
)

if response_params_dic.json()["status"] == 200:
    print(
        "Request successful"
    )
```

## Examples in R

```
library(httr2) # For handling API authorization and requests
library(jsonlite) # For handling the response of the API
library(dplyr) # For handling data

# Option #1 (parameters in the url)

# API url - "https://acleddata.com/api/acled/read?format=json&country=Georgia:OR:country=Armenia&year=2"
api_url <- "https://acleddata.com/api/acled/read?format=json&country=Georgia:OR:country=Armenia&year=2"
# Token authorization url
token_url <- "https://acleddata.com/oauth/token"

response <-
  # Prepare request for data from the API endpoint with parameters in the url
  request(api_url) %>%
  # Authorize credentials directly (not shown); if not, a password box will pop up
  # Can add a 'password' parameter, but it will be automatically stored so you do not have to reenter your password until the token req_oauth_password()
  req_auth_password(
    client = oauth_client("acled", token_url),
    username = "your_email@gmail.com"
  ) %>%
  # Execute the API request
  req_perform()

response_df_option1 <- resp_body_json(response, simplifyVector = TRUE)$data

# Option #2 (parameters as a list)

# Set up the list of parameters
parameters <- list(
  country = "Georgia:OR:country=Armenia",
  year = 2021,
  fields = "event_id_cnty|event_date|event_type|country|fatalities"
)

response <-
  # Prepare request for data from the API endpoint. Note that only the base url is provided this time
  request("https://acleddata.com/api/acled/read?format=json") %>%
  # Authorize credentials
  req_oauth_password(
    client = oauth_client("acled", token_url),
    username = "your_email@gmail.com"
  ) %>%
  # Add list of parameters
  req_url_query(list(parameters)) %>%
  # Execute the API request
  req_perform()

response_df_option2 <- resp_body_json(response, simplifyVector = TRUE)$data
```

## SHARE ON



## Related content

