

ACLED Religion API

User Guide



Richard Holmes

<https://ampersandstudio.uk/>

February 2021

Contents

ACLED RELIGION API

Introduction	3
<i>API Access Detail</i>	3
<i>API Developer Account</i>	3
<i>Sample API Calls and Responses</i>	4
<i>Response Format</i>	4
<u>RELIGION</u>	
Commands	5
<i>Read</i>	5
Query Filters	8
<i>Limit Query & Pagination</i>	11
<i>Limit Fields Returned</i>	11
<i>Complex Queries</i>	11

ACLED RELIGION API

Version 1.0

Introduction

The following document highlights the basic steps for interacting with the Acled Religion API. The API is RESTful in nature and is accessed via Basic HTTP(S) authentication.

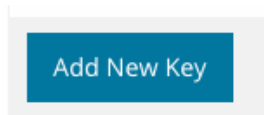
API Access Detail

The full URL for accessing the API is <https://api.acleddata.com/{data}/{command}>, where “data” represents the type of data to be collected and “command” represents the action to be executed. See below for details regarding possible data types and usages.

API Developer Account

You must register for an API key in order to access the API. To register please visit <https://developer.acleddata.com/>.

Please note you should register from the device you intend to access the API from as your IP address will be linked to your account and may determine your access to the API. On completing the registration form you may be required to wait for approval or may have limited access to the API, until your account has been approved. The dashboard will display any notifications that relate to your account, that should be completed before proceeding to create an API Key. Once you are eligible to create a key you will see the following button is active, within the dashboard.



Clicking the button will generate a random API key for your account. **You will only be shown the key one time and must copy and save it in a secure location.** If you fail to copy the key you will need to revoke the active key and generate a new key. This does not affect your usage of the API or reset your account usage limit.

Sample API Calls and Responses

API calls may be made in any standard browser or using any programmatic language that is capable of making HTTP requests. The samples below demonstrate the URL to be called to make the request.

The following points should be noted:

- & All requests will be denied without a key and email address.
- & This API only uses the GET or POST request. A GET request is advised whereby all data sent will be contained within standard Query String parameter formats and URLencoded.
- & All responses from the API shall be formatted as JSON unless specifically requested in either XML, CSV or TXT format.
- & TXT format returns a plain text csv string.
- & A limit of 500 lines of data (1000 rows for monadic) will be returned by default for ACLED and Actor data types. Larger requests may be made, however.
- & All fields, specific to the data type, will be returned by default. Reduced field lists can be requested for some data types.
- & ACLED data is returned in date order DESC (starting with the latest).

Response Format

By default the response is returned in JSON format but it's possible to return the response in XML, CSV and TXT format too. In order to return another format you simply add the relevant extension to the end of the command name so the query would look like the following:

Format	HTTP Request Format	MIME Type
JSON	https://api.acleddata.com/{data}/{command}	application/json
XML	https://api.acleddata.com/{data}/{command}.xml	text/xml
CSV	https://api.acleddata.com/{data}/{command}.csv	text/csv
TXT	https://api.acleddata.com/{data}/{command}.txt	text/plain

RELIGION

This data call returns the main dataset

Commands

Read

In order to retrieve the data you must make a GET or POST request to the following URL:

https://api.acleddata.com/religion/read?key={api_key}&email={email}

Returned Data (json only)

Attribute Name	Type	Description
status	int	A number representing the request status
success	boolean	A boolean representation on the success of the call
last_update	int	The number of hours since the last update to the data
count	int	The number of data rows returned
messages	array	An array of information messages that may require future action
data	array	The rows of data returned. For details of attributes returned in each row, see below.
filename	string	The filename that will be used for csv calls
error	array	The details of the error with a status as an integer and message as a string

Returned Data (json, xml, txt, csv)

Attribute Name	Type	Description
data_id	int	A unique id for the row of data
iso	int	A numeric code for each individual country. Referenced here - ISO Country List
event_id_cnty	string	An individual identifier by number and country acronym
event_id_no_cnty	string	An individual numeric identifier
event_date	date	The date the event occurred in the format: yyyy-mm-dd

Attribute Name	Type	Description
year	int	The year the event occurred.
time_precision	int	A numeric code indicating the level of certainty of the date coded for the event
event_type	string	The type of conflict event
sub_event_type	string	The type of conflict sub event
harass_event_type	string	The type of harassment event
harass_sub_event_type	string	The type of harassment sub event
relig_context	string	The religious context(s) of the event
actor1	string	The named actor involved in the event
assoc_actor_1	string	The named actor allied with or identifying ACTOR1
relig_affil_1	string	The religious affiliation of ACTOR1 and/or ASSOC_ACTOR_1
inter1	int	A numeric code indicating the type of ACTOR1.
actor2*	string	The named actor involved in the event
assoc_actor_2*	string	The named actor allied with or identifying ACTOR2
relig_affil_2*	string	The religious affiliation of ACTOR2 and/or ASSOC_ACTOR_2
inter2*	int	A numeric code indicating the type of ACTOR2.
interaction	int	A numeric code indicating the interaction between types of ACTOR1 and ACTOR2
region	string	The region in which the event took place
country	string	The name of the country the event occurred in
admin1	string	The largest sub-national administrative region in which the event took place
admin2	string	The second largest sub-national administrative region in which the event took place
admin3	string	The third largest sub-national administrative region in which the event took place
location	string	The location in which the event took place
latitude	decimal	The latitude of the location
longitude	decimal	The longitude of the location

Attribute Name	Type	Description
geo_precision	int	A numeric code indicating the level of certainty of the location coded for the event
source	string	The source of the event report
source_scale	string	The scale of the source
notes	string	A short description of the event
fatalities	int	The number of reported fatalities which occurred during the event
fatalities_precision		A numeric code indicating the level of certainty of reports around the number of fatalities coded for the event
timestamp	int / date	The unix timestamp this data entry was last updated
iso3		A 3 character code representation of each country

* These attributes will be returned as a new data row if export type is monadic.

Query Filters

Each field can be searched to filter the returned data. By default each field will search by = or LIKE based on the table below. This can be changed by sending a new query string name value pair, where the name has ‘_where’ appended to it. The following table shows the default query type that will be used by each field. **The key and email address query must be included in all requests to indicate that you have registered your use of our API.**

Query Name	Type	Query String
key	=	{api_key}
email	=	{email address associated with key}
data_id	=	?data_id={number}
iso	=	?iso={number}
event_id_cnty	LIKE	?event_id_cnty={text}
event_id_no_cnty	LIKE	?event_id_no_cnty={text}
event_date	=	?event_date={yyyy-mm-dd}
year	=	?year={yyyy}
time_precision	=	?time_precision={number}
event_type	LIKE	?event_type={text}
sub_event_type	LIKE	?sub_event_type={text}
relig_context	LIKE	?relig_context={text}
relig_context	LIKE	?relig_context={text}
actor1	LIKE	?actor1={text}
assoc_actor_1	LIKE	?assoc_actor_1={text}
relig_affil_1	LIKE	?relig_affil_1={text}
inter1	=	?inter1={number}
actor2	LIKE	?actor2={text}
assoc_actor_2	LIKE	?assoc_actor_2={text}
relig_affil_2	LIKE	?relig_affil_2={text}
inter2	=	?inter2={number}
interaction	=	?interaction={number}

Query Name	Type	Query String
region	=	?region={number}
country	LIKE	?country={text}
admin1	LIKE	?admin1={text}
admin2	LIKE	?admin2={text}
admin3	LIKE	?admin3={text}
location	LIKE	?location={text}
latitude	=	?latitude={number}
longitude	=	?longitude={number}
geo_precision	=	?geo_precision={number}
source	LIKE	?source={text}
source_scale	LIKE	?source_scale={text}
notes	LIKE	?notes={text}
fatalities	=	?fatalities={number}
fatalities_precision	=	?fatalities_precision={number}
timestamp	>=	?timestamp={number/yyyy-mm-dd}
export_type	=	?export_type={text}
iso3	=	?iso3={text}

References

For some attributes a number is required instead of text. The following reference tables provides the numeric code to be used for certain content.

inter 1 / inter 2	Numeric Code
State Forces	1
Rebel Forces	2
Militia Groups	3
Communal / Identity Groups	4
Rioters	5
Protesters	6
Civilians	7

inter 1 / inter 2	Numeric Code
Foreign / Others	8

region	Numeric Code
Western Africa	1
Middle Africa	2
Eastern Africa	3
Southern Africa	4
Northern Africa	5
South Asia	7
Western Asia	8
Southeast Asia	9
Middle East	11
Europe	12
Caucasus and Central Asia	13
Central America	14
South America	15
Caribbean	16
East Asia	17
North America	18

- & The ISO country list can be viewed here - [ISO Country Link](#)
- & All LIKE queries will include a wildcard before and after the entered text.
- & Multiple queries can be searched with name/value pairs separated by &. Each field will be searched using AND so all arguments must match for data to be returned.
- & More complex queries can be searched to include the OR clause. See Complex Queries below.
- & If export_type is not included it will be dyadic. For monadic export you will need to include the export_type query.

To change the default query type you can add an additional name/value pair using the query name and appending ‘_where’ to the query name. The query value could be LIKE or %3D for ‘=’. Additional types of ‘<’, ‘>’ and ‘BETWEEN may also be used, representing less

than, greater than and between. The between requires the query name value to separate the 2 values with a |.

Example:

```
?key={api_key}&email={email address}&event_id_cnty={text}&event_id_cnty_where=%3D  
?key={api_key}&email={email address}&event_date={yyyy-mm-dd|yyyy-mm-dd}  
&event_date_where=BETWEEN
```

Limit Query & Pagination

By default there is a limit of 500 rows of data returned, 1000 rows if export_type = monadic. You can use the limit query name to change the default number. Setting limit as 0 will return all relevant data. It is likely returning all data will cause a timeout error and we therefore recommend using the page query instead. Each page will return 500 (1000 for monadic) rows of data.

Example:

```
?key={api_key}&email={email address}&limit=100 will return 100 rows of data (200 for monadic).
```

```
?key={api_key}&email={email address}&page=1 will return the first 500 rows of data (1000 for monadic)
```

```
?key={api_key}&email={email address}&page=2 will return the next 500 rows of data (1000 for monadic)
```

Limit Fields Returned

By default all fields will be returned for each line of data. You can use the fields query name to change the field items returned. Multiple fields can be requested by separating each one with a pipe (|).

Example:

```
?key={api_key}&email={email address}&fields=iso will return just the iso field.
```

```
?key={api_key}&email={email address}&fields=iso|fatalities will return the iso and fatalities data for each row.
```

Complex Queries

By default all fields must match for the data to be returned. In some instances more complex queries may be required to use the OR clause. This is possible by separating the

fields to join, by OR, with :OR: text. The main query item will be the first item in the join, followed by the other items split with :OR: . These can be used with other queries too.

Example:

?key={api_key}&email={email address}&{fieldname}={text}:OR:{fieldname2}={text2}:OR:{fieldname3}={text3} will return data where field = text OR field2 = text2 OR field3 = text3.

?key={api_key}&email={email address}&{fieldname}={text}:OR:{fieldname2}={text2}&country={country} will return data where field = text OR field2 = text2 AND country = country.

All items wrapped in {} must be replaced with the actual fields or text required. The curly brackets must also be removed from the query.

