ACLED API

User Guide

Richard Holmes

https://ampersandstudio.uk/

June 2020

Contents

ACLED API

Version 2.9

Introduction

The following document highlights the basic steps for interacting with the Acled 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. **You must include a query acknowledging that you have read the terms and conditions of use below.**

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 accepting the terms.
- 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

Terms & Conditions

Terms of Use and Attrition Policy

Background

Non-Commercial Licenses - ACLED's full dataset is available for use free of charge by non-commercial entities and organizations (e.g., non-profit organizations, government agencies, academic institutions) using the data for non-commercial purposes, subject to these Terms of Use. Non-commercial licenses may also be granted to for-profit media outlets or journalists citing ACLED's content in works of journalism; provided that such works are made available to the general public and benefit public discourse on the topic, subject to ACLED's prior, written approval.

Commercial Licenses – All other commercial exploitation of ACLED content (including, but not limited to, any sale, licensing, sublicensing, or other distribution of the data, in whole or in part, or any incorporation of the data or analysis, in whole or in part, in a product for sale, license, sublicense, or other distribution, to third parties for profit, or use of the data or analysis by a commercial entity for situational awareness) is prohibited, except as expressly agreed by ACLED in a written Commercial License Agreement executed by ACLED and the licensee. Commercial entities seeking a commercial license to use the full ACLED dataset must submit a request to ACLED at admin@acleddata.com, including the details of the requested use. ACLED reserves the right to withhold approval in its sole discretion. Any approval granted is limited strictly to the use described in the corresponding request and/ or approval (which will supersede the request in the case of any inconsistency or conflict). All commercial licensees are subject to ACLED's Terms & Conditions. By including "Accept" in the query, they are accepting the Terms & Conditions as outlined in the linked document.

Regardless of the type of license, all rights to use or exploit ACLED data or analysis are conditioned on the licensee's adherence to the Attribution Requirements outlined below.

Crawling the ACLED website or API is allowed if done in accordance with the provisions of our robots.txt file, but scraping is prohibited.

No user is permitted to (i) use ACLED's data or analysis in any manner that may harm, target, oppress or defame ACLED, the data subjects, or any group or population, or cause any of the foregoing to be harmed, targeted, oppressed or defamed; (ii) provide, permit or allow direct access to any of ACLED's original/raw data or analysis; (iii) use any of the data or analysis to create, develop, support or provide benchmarking for any dataset, product or platform similar to, or in competition with, or would create a functional substitute for, any of ACLED content, products or platform; and (iv) provide, permit or allow access to any of the data or analysis by ACLED's competitors.

Attribution Requirements

1. **If using ACLED data in any way**, **direct or manipulated,** the data must be clearly acknowledged. Acknowledgement should include (1) a footnote with the full citation which includes a/the link to ACLED's website (see below for examples), (2) in text citation/acknowledgement, stating where the data you use are from and that ACLED data are publicly available, and (3) clear citation on any and all visuals making use of ACLED data.

2. **If generating a data file for public or private use, and presenting those data to another party**, the ACLED data included must be directly acknowledged in a source column, including ACLED's full name and a link: "Armed Conflict Location & Event Data Project (ACLED); https://www.acleddata.com."

3. **To reference the ACLED codebook, please cite as follows (substituting for the correct year):**

   "ACLED. (2017). "Armed Conflict Location & Event Data Project (ACLED) Codebook, 2017.""

4. **If using ACLED data in an academic paper or article, please cite as follows:**

   "Raleigh, Clionadh, Andrew Linke, Håvard Hegre and Joakim Karlsen. (2010). "Introducing ACLED-Armed Conflict Location and Event Data." Journal of Peace Research 47(5) 651-660."

5. **If referring to figures or statistics published in ACLED analysis, infographics, working papers, etc.,** please cite the individual analysis piece or paper, including the author(s), using the following format:

Hart, Tom, and Lauren Blaxter. (23 November 2018). "Ceasefire Divisions: Violations of the Truce with Gaza Lead to Rising Political Pressures in Israel." Armed Conflict Location & Event Data Project (ACLED). <https://www.acleddata.com/2018/11/23/ceasefire-divisions-violations-of-the-truce-with-gaza-lead-to-rising-political-pressures-in-israel/>.

If the ACLED piece does not have an author recognized (often the case for pieces on ACLED methodology), the citation should identify ACLED as the author.

6. **If using ACLED data in a visual, graphic, map or infographic of your own,** please attribute the source data prominently on the visual itself or within the key / legend.

7. **If you wish to reproduce or publish a graphic, graph or map ACLED has already published (rather than creating an original image using raw data)**, please cite the individual analysis piece or paper, including the author(s), using the following format:

   Hart, Tom, and Lauren Blaxter. (23 November 2018). "Ceasefire Divisions: Violations of the Truce with Gaza Lead to Rising Political Pressures in Israel." Figure 1, Armed Conflict Location & Event Data Project (ACLED). *<https://www.acleddata.com/2018/11/23/ceasefire-divisions-violations-of-the-truce-with-gaza-lead-to-rising-political-pressures-in-israel/>*. © 2018 ACLED All rights reserved. Used with permission from ACLED.

If the ACLED piece does not have an author recognized (often the case for pieces on ACLED methodology), the citation should identify ACLED as the author.

If you intend to use ACLED data or analysis in a manner not described in these Attribution Requirements, contact us directly at admin@acleddata.com for instruction regarding the attribution requirements.

If you have any other questions about the Terms of Use or Attribution Policy, or their application, please feel free to contact us directly at admin@acleddata.com.

ACLED

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/acled/read?terms=accept

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
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
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

Attribute Name	Type	Description
event_type	string	The type of conflict event
sub_event_type	string	The type of conflict sub event
actor1	string	The named actor involved in the event
assoc_actor_1	string	The named actor allied with or identifying ACTOR1
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
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
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
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 terms query must be included in all requests to indicate that you have read and accept the terms of use.**

Query Name	Type	Query String
terms	**=**	**accept**
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}
actor1	LIKE	?actor1={text}
assoc_actor_1	LIKE	?assoc_actor_1={text}
inter1	=	?inter1={number}
actor2	LIKE	?actor2={text}
assoc_actor_2	LIKE	?assoc_actor_2={text}
inter2	=	?inter2={number}
interaction	=	?interaction={number}
region	=	?region={number}
country	LIKE	?country={text}
admin1	LIKE	?admin1={text}
admin2	LIKE	?admin2={text}
admin3	LIKE	?admin3={text}
location	LIKE	?location={text}

Query Name	Type	Query String
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}
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
Foreign / Others	8

region	Numeric Code
Western Africa	1
Middle Africa	2
Eastern Africa	3
Southern Africa	4

region	Numeric Code
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

&   The ISO country list can be viewed here - <u>ISO Country Link</u>

&   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:

?terms=accept&event_id_cnty={text}&event_id_cnty_where=%3D
?terms=accept&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:
?terms=accept&limit=100 will return 100 rows of data (200 for monadic).

?terms=accept&page=1 will return the first 500 rows of data (1000 for monadic)
?terms=accept&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:
?terms=accept&fields=iso will return just the iso field.
?terms=accept&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:
?terms=accept&{fieldname}={text}:OR:{fieldname2}={text2}:OR:{fieldname3}={text3} will return data where field = text OR field2 = text2 OR field3 = text3.

?terms=accept&{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.

Actor

This data call returns the actors

Commands

Read

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

https://api.acleddata.com/actor/read?terms=accept

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
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
actor_name	string	The name of the actor
first_event_date	date	The date the first event for this actor occurred in the format: yyyy-mm-dd
last_event_date	date	The date the last event for this actor occurred in the format: yyyy-mm-dd
event_count	int	The number of events that have occurred with this actor

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 terms query must be included in all requests to indicate that you have read and accept the terms of use.**

Query Name	Type	Query String
terms	**=**	**accept**
actor_name	LIKE	?actor_name={text}
first_event_date	=	?first_event_date={yyyy-mm-dd}
last_event_date	=	?last_event_date={yyyy-mm-dd}
event_count	>=	?event_count={number}

  &   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.

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:
?terms=accept&actor_name={text}&actor_name_where=%3D
?terms=accept&last_event_date={yyyy-mm-dd|yyyy-mm-dd}&last_event_date_where=BETWEEN

Limit Query & Pagination

By default there is a limit of 500 rows of data returned. 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 rows of data.

Example:

?terms=accept&limit=100 will return 100 rows of data.

?terms=accept&page=1 will return the first 500 rows of data
?terms=accept&page=2 will return the next 500 rows of data

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:

?terms=accept&field={text}:OR:field2={text2}:OR:field3={text3} will return data where field = text OR field2 = text2 OR field3 = text3.

?terms=accept&field={text}:OR:field2={text2}&event_count={number} will return data where field = text OR field2 = text2 AND event_count = number.

Actor Type

This data call returns the actor types

Commands

Read

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

https://api.acleddata.com/actortype/read?terms=accept

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
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
actor_type_id	int	The id of the actor type
actor_type_name	string	The name of the actor type
first_event_date	date	The date the first event for this actor type occurred in the format: yyyy-mm-dd
last_event_date	date	The date the last event for this actor type occurred in the format: yyyy-mm-dd
event_count	int	The number of events that have occurred with this actor type

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 terms query must be included in all requests to indicate that you have read and accept the terms of use.**

Query Name	Type	Query String
terms	**=**	**accept**
actor_type_id	=	?actor_type_id={number}
actor_type_name	LIKE	?actor_name={text}
first_event_date	=	?first_event_date={yyyy-mm-dd}
last_event_date	=	?last_event_date={yyyy-mm-dd}
event_count	>=	?event_count={number}

    &   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.

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:
?terms=accept&actor_type_name={text}&actor_type_name_where=%3D
?terms=accept&last_event_date={yyyy-mm-dd|yyyy-mm-dd}&last_event_date_where=BETWEEN

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:
?terms=accept&field={text}:OR:field2={text2}:OR:field3={text3} will return data where field = text OR field2 = text2 OR field3 = text3.

?terms=accept&field={text}:OR:field2={text2}&event_count={number} will return data where field = text OR field2 = text2 AND event_count = number.

Country

This data call returns the countries

Commands

Read

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

https://api.acleddata.com/country/read?terms=accept

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
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
country	string	The name of the country
iso	int	The iso number of the country
iso3	string	The iso3 representation of the country
first_event_date	date	The date the first event for this actor type occurred in the format: yyyy-mm-dd
last_event_date	date	The date the last event for this actor type occurred in the format: yyyy-mm-dd
event_count	int	The number of events that have occurred with this actor type

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 terms query must be included in all requests to indicate that you have read and accept the terms of use.**

Query Name	Type	Query String
terms	**=**	**accept**
country	LIKE	?country={text}
iso	=	?iso={number}
iso3	=	?iso3={text}
first_event_date	=	?first_event_date={yyyy-mm-dd}
last_event_date	=	?last_event_date={yyyy-mm-dd}
event_count	>=	?event_count={number}

   &   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.

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:

?terms=accept&country={text}&country_where=%3D

?terms=accept&last_event_date={yyyy-mm-dd|yyyy-mm-dd}&last_event_date_where=BETWEEN

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:
?terms=accept&field={text}:OR:field2={text2}:OR:field3={text3} will return data where field = text OR field2 = text2 OR field3 = text3.

?terms=accept&field={text}:OR:field2={text2}&event_count={number} will return data where field = text OR field2 = text2 AND event_count = number.

Region

This data call returns the regions

Commands

Read

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

https://api.acleddata.com/region/read?terms=accept

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
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
region	int	The id of the region
region_name	string	The name of the region
first_event_date	date	The date the first event for this actor type occurred in the format: yyyy-mm-dd
last_event_date	date	The date the last event for this actor type occurred in the format: yyyy-mm-dd
event_count	int	The number of events that have occurred with this actor type

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 terms query must be included in all requests to indicate that you have read and accept the terms of use.**

Query Name	Type	Query String
terms	**=**	**accept**
region	=	?region={number}
region_name	LIKE	?region_name={text}
first_event_date	=	?first_event_date={yyyy-mm-dd}
last_event_date	=	?last_event_date={yyyy-mm-dd}
event_count	>=	?event_count={number}

& 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.

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:
?terms=accept&region_name={text}&region_name_where=%3D
?terms=accept&last_event_date={yyyy-mm-dd|yyyy-mm-dd}&last_event_date_where=BETWEEN

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:

?terms=accept&field={text}:OR:field2={text2}:OR:field3={text3} will return data where field = text OR field2 = text2 OR field3 = text3.

?terms=accept&field={text}:OR:field2={text2}&event_count={number} will return data where field = text OR field2 = text2 AND event_count = number.