Pushshift Documentation

Release 4.0

Jason M. Baumgartner

Feb 11, 2018

Contents

Preface

The pushshift.io Reddit API was designed and created by the /r/datasets mod team to help provide enhanced functionality and search capabilities for searching Reddit comments and submissions. The project lead, /u/stuck_in_the_matrix, is the maintainer of the Reddit comment and submissions archives located at https://files.pushshift.io and lead architect for the Pushshift API project.

What is the purpose of this API?

The goal of this project is to provide a feature-rich API for searching Reddit comments and submissions and to give the ability to aggregrate the data in various ways to make interesting discoveries within the data. This RESTful API gives full functionality for searching Reddit data. With this API, you can quickly find the data that you are interested in and discover interesting correlations within the data.

How many objects are indexed on the back-end?

There are over **four billion** comments and submissions available via the search API.

Understanding the API

There are two main ways of accessing the Reddit comment and submission database. One is by using the API directly via https://api.pushshift.io/ and the other is through accessing the back-end Elasticsearch search engine via https://elastic.pushshift.io/ This document will focus on the first method and give a broad overview of all the parameters available when conducting a search. This document will also explore the use of more advanced API parameters to utilize more focused searches.

Using the https://api.pushshift.io endpoints

There are two main endpoints used to search all publicly available comments and submissions on Reddit:

- https://api.pushshift.io/reddit/comment/search
- https://api.pushshift.io/reddit/submission/search

In the next section, we will explore how to perform more effective searches using the comment search endpoint.

Comments Search

To search comments, use the https://api.pushshift.io/reddit/comment/search endpoint. Let's start with a few examples and then go over the various parameters available when using this endpoint. One of the simplest searches is using just the q parameter. The q parameter is used to search for a specific word or phrase. Here is an example:

Search for the most recent comments mentioning the word "science"

https://api.pushshift.io/reddit/comment/search/?q=science

This will search the most recent comments with the term "science" in the body of the comment. This search is not case-sensitive, so it will find any occurence of the term "science" regardless of capitalization. The API defaults to sorting by recently made comments first. After performing this search, 25 results are returned. This is the default size for searches and can be adjusted using the size parameter. This will be discussed in further detail in the parameters section. Data is returned in JSON format and actual search results are included in the "data" key. There is also a "metadata" key that gives additional information about the search including total number of results found, how long the search took to process, etc. If aggregations are requested, all aggregation data is returned under the aggs key.

Comment Search Parameters

7.1 Time based Parameters

7.1.1 after

The "after" parameter allows you to restrict the comments returned from a search by epoch time. This parameter also supports some convenience methods via abbreviations for time. If you use an epoch time for the value of the "after" parameter, it will return all comments with a created_utc epoch time greater than that value. You can also use abbreviations such as 24h (24 hours), 90s (90 seconds), 7d (7 days), etc. As an example, if you wanted to return all comments containing the term "quantum" that were made in the past 24 hours, you would make the following API call:

https://api.pushshift.io/reddit/comment/search/?q=quantum&after=24h

7.1.2 before

The "before" parameter works exactly like the after parameter, except it will return comments made before the epoch time given. Also, like the "after" parameter, it accepts abbreviated values for time. As an example, if you wanted to search for comments containing the term "universe" that were at least 30 days old, you would make the following API call:

https://api.pushshift.io/reddit/comment/search/?q=universe&before=30d

7.1.3 reply_delay

The "reply_delay" can be used to search comments by the amount of time that elaspsed before the comment reply was made. For instance, if a comment is made at 3:00:00 pm and a reply to that comment was made at 3:01:53 pm, a total of 113 seconds elasped before the reply was made. Using the reply_delay parameter, you can find comments that were made within X second to the parent comment (or submission if the comment is a top level comment). This parameter is excellent for finding bot-like activity on Reddit. As an example, let's say you are a moderator of /r/politics and you want to see what bots are active in your subreddit over the past 24 hours. Using the "reply_delay" parameter along

with the "subreddit" and "after" parameter will allow you to see bot-like activity. Usually, most bots will reply within 30 seconds to the parent object (comment or submission).

Here is an example call to the API using the above scenario. This API call will show comments that were made in less than 30 seconds over the past 24 hours to the subreddit /r/politics:

https://api.pushshift.io/reddit/comment/search/?subreddit=politics&reply_delay=%3C30&after=24h

This is especially powerful when used in tandem with the "aggs" parameter with a value of "subreddit":

https://api.pushshift.io/reddit/comment/search/?subreddit=politics&reply_delay=%3C30&after=24h&aggs=author&size=0

7.1.4 utc_hour_of_week

The "utc_hour_of_week" parameter is a parameter that is primarily meant as an aggregation method to show comment volume by hour of week (so that you could track trends and see when subreddits or specific authors were most active). The parameter itself can be used directly to limit comments by a specific hour of the week as well. The range is from 0 to 168 with 0 being midnight on Monday and 168 being the 23'rd hour of Sunday night.

7.1.5 utc_hour_of_day

The "utc_hour_of_day" parameter is a parameter that is primarily meant as an aggregation method to show comment volume over the course of a day. When using this parameter as an aggregation type, it shows when a subreddit or author is move active throughout a typical day.

7.2 Filter Parameters

7.2.1 size

The "size" parameter limits the number of objects returned within the data array. The parameter accepts an integer up to 500. This parameter is associated with the data array only and does not influence the number of results under aggregations when using the "aggs" parameter. Reference the "agg_size" parameter for limiting the size of aggregation results instead.

As a quick example, if you wanted to retrieve 25 comments that contained the term "universe," you would make the following API call:

https://api.pushshift.io/reddit/comment/search/?q=universe&size=25

7.2.2 filter

The "filter" parameter is used to limit the amount of information returned within objects contained in the data array. Let's say you wanted to do a comment search for the term "denver" and you only needed the author, score and subreddit fields. Using filter, you could restrict the API and only return those fields. This is an example using the filter parameter using the previous example:

https://api.pushshift.io/reddit/comment/search/?q=denver&filter=author,score,subreddit

7.2.3 sort

The "sort" parameter is used to sort results based on a given key. For comments, generally one would want to sort by the comment creation date or the comment scores. To use the sort parameter, you would specify the key used for the sort and then a colon and then the sort order using either "asc" or "desc". The following example does a search for "patriots" within the subreddit "nfl" and sorts the results by score descending (showing comments with the highest score):

https://api.pushshift.io/reddit/comment/search/?q=patriots&subreddit=nfl&sort=score:desc

7.2.4 length

The "length" parameter allows for restricting the results to comments above or below a certain character length. This is helpful for excluding short comments when searching for comments with more substance, etc. When using this parameter, simply set the value to specific length or use the "<" or ">" characters to select comments less than or greater than a certain length. For example, if you wanted to find comments in the subreddit "askhistorians" with a length greater than 500 characters, you could make an API call like this:

https://api.pushshift.io/reddit/comment/search/?subreddit=askhistorians&length=%3E500

7.2.5 user_removed

A boolean parameter that is true if a user removed their own comment.

7.2.6 mod_removed

A boolean parameter that is true if a moderator removed a user's comment.

7.2.7 nest_level

The nest level of a comment. A top level comment will have a nest level of 1. A comment that is a reply to a top level comment will have a nest level of 2 and so on.

7.3 Comment Attribute Parameters

7.3.1 q

This parameter will return comments matching the keyword or phrase matching the parameter value. The value can be a simple term or a complex phrase and is case-insensitive. For example, to find comments that mention the band Radiohead, one would make the following API call:

https://api.pushshift.io/reddit/comment/search/?q=radiohead

This parameter accepts many different options that can help narrow down the search to find specific comments. Here are some examples that show various ways to maximize the utility of this parameter when searching for specific comments.

Multiple terms (AND operation)

To find comments that match two different words, seperate the words using a "+" sign. The following would return comments containing the term "Radiohead" and the term "band":

https://api.pushshift.io/reddit/comment/search/?q=radiohead+band

Multiple terms (OR operation)

To find comments that match either of two different words, seperate the words using a "|" sign. The following would return comments containing the term "Radiohead" or the term "Nirvana":

https://api.pushshift.io/reddit/comment/search/?q=radiohead|nirvana

Negation

To find comments that match one word but not another word, use a "-" before the word you wish to exclude. For example, the following would return comments containing the term "Radiohead" but not the word music:

https://api.pushshift.io/reddit/comment/search/?q=radiohead-music

Exact Phrase

If you wanted to find an exact phrase, you can put the phrase in quotation marks. The following example will find comments that contain the phrase "band radiohead":

https://api.pushshift.io/reddit/comment/search/?q="band%20radiohead"

Complex Combinations

You can combine many of the previous types of operations and group them using parentheses to create advanced options for searching. As a more complicated example, let's say you wanted to search for comments containing "Nirvana" or "Music" but not the word "songs" or "group":

https://api.pushshift.io/reddit/comment/search/?q=(Nirvana|Music)-(songs+group)

7.3.2 author

This parameter will restrict the search to specific Reddit authors. Every Reddit comment has an author which means you can restrict your search results to specific people.

Inclusive search

To find comments by one author, simply set the value of the author parameter to that author's name. The field is not case-sensitive and allows you to include multiple authors seperated by a comma. This example will find comments by the author "spez" or "automoderator":

https://api.pushshift.io/reddit/comment/search/?author=spez,automoderator

Exclusive search

You can also use this parameter to return all comments *not* made by specific authors. Using the previous example, if you wanted to return all comments that were not made by automoderator or spez, you would put a "!" before the name. Example:

https://api.pushshift.io/reddit/comment/search/?author=!automoderator,!spez

7.3.3 author_flair_css_class

Parameter to filter comments based on the author's flair css class.

7.3.4 author_flair_text

Parameter to filter comments based on the author's flair text.

7.3.5 subreddit

This parameter will restrict the search to specific subreddits. Every Reddit comment is associated with a submission which is associated with a subreddit.

Inclusive search

To find comments within a subreddit or multiple subreddits, set the value of the subreddit parameter to the subreddit(s) that you are interested in. This field is not case-sensitive and allows you to include multiple subreddits seperated by a comma. This example will find comments within the subreddit askscience:

https://api.pushshift.io/reddit/comment/search/?subreddit=askscience

Exclusive search

You can also use this parameter to return all comments *not* within a subreddit or multiple subreddits. Using the previous example, if you wanted to return all comments that were not made within askscience, you would put a "!" before the subreddit name. Example:

https://api.pushshift.io/reddit/comment/search/?subreddit=!askscience

7.3.6 score

The score parameter allows you to search for comments with a specific score or range of scores. This parameter is helpful in finding higher quality comments (although a high score comment isn't necessarily always a quality comment). As an example, this API call will find comments with the term "boston" with a score greater than 500:

https://api.pushshift.io/reddit/comment/search/?q=boston&score=%3E500

7.3.7 gilded

Like the score parameter, this allows you to search for comments with a certain amount of gildings. To find a comment that contains the term "amazing" and has been gilded (no matter how many times), you would make the following API call:

https://api.pushshift.io/reddit/comment/search/?q=amazing&gilded=%3E0

You could also search comments and sort by the gilded parameter to return comments with many gildings ranked in descending order:

https://api.pushshift.io/reddit/comment/search/?q=amazing&sort=gilded:desc

7.3.8 distinguished

Parameter to retreieve comments based on the type of user ("moderator", "admin", etc.)

7.3.9 id

Parameter to retrieve specific comments by their id.

7.3.10 link_id

Parameter to retrieve comments within a specific submission.

7.3.11 edited

A boolean parameter that is true if a user made an edit to their comment.

7.3.12 parent_id

A parameter that gives the parent id of a comment (which could be another comment or a submission if the comment is a top level comment).

7.4 Aggregation Parameters

7.4.1 agg

The agg parameter is used to create aggregations. (This needs to be expanded . . .)

Search parameters for comments

There are numerous additional parameters that can be used when performing a comment search. Let's go over them and provide examples for each.

Parameter	Description	Accepted Values	Example Usage
q	Search term or phrase	String / Quoted String for phrases	q=radio head
ids	Get specific comments via their ids	Comma-delimited base36 ids	ids=ce2 31,ce23 2,ce233
size	Number of results to return within the data array	0 to 500 (Int)	size=10 0
fields	Only return specific fields under the data array	comma-delimited string	fields= subredd it,auth or
sort	Sort results using a specific key (key:direction where direction is "asc" or "desc")	sortable key:"as c" or "desc"	sort=sc ore:des c
aggs	Return aggregation(s) summary	author, link_id, created_utc, subreddit	aggs=li nk_id, author
author	Restrict to a specific author(s)	Comma-delimited string	author= david,b illy,to m (only include these authors)
subreddit	Restrict to a specific subreddit(s)	Comma-delimited string	subredd it=asks cience, science
after	Return results after this date	N/A	Epoch value or Integer + "s,m,h, d" (i.e. 30d for 30 days)
before	Return results before this date	N/A	Epoch value or Integer + "s,m,h, d" (i.e. 30d for 30 days)
frequency	Used with the aggs parameter when set to created_utc	N/A	"second ", "minute ", "hour", "day"

8.1 Getting comments based on id

You can retrieve comments directly by using the ids parameter. To get a batch of comments by their id, use the following example:

Retrieve three comments using their base 36 id values

https://api.pushshift.io/reddit/comment/search?ids=dlrezc8,dlrawgw,dlrhbkq

8.2 Using the subreddit parameter

There are quite a few parameters to review, so let's start by providing some more complex examples and how to use the parameters above. Let's continue with the previous example above and expand on our "science" keyword search. What if we wanted to search for the term "science" but restrict it to a specific subreddit? By using the subreddit parameter, we can do that:

Search for the most recent comments mentioning the word "science" within the subreddit /r/askscience

https://api.pushshift.io/reddit/search/comment/?q=science&subreddit=askscience

8.3 Using the sort and size parameters

This will return 25 comments containing the term "science" but only from the /r/askscience subreddit. Since we didn't ask for a specific sort method, the most recent comments are returned (the sort parameter defaults to "desc"). What if we wanted the first comment ever to /r/askscience that mentioned the word "science"? We could use the sort and size parameters to handle that.

Search for the most recent comments mentioning the word "science" within the subreddit /r/askscience

https://api.pushshift.io/reddit/search/comment/?q=science&subreddit=askscience&sort=asc&size=1

This is the result:

```
{
    "data": [
        {
            "author": "MockDeath",
            "author_flair_css_class": null,
            "author_flair_text": null,
            "body": "Knowing more would definitely help.  I guess all you can do is
→find out if they know the basics like you said then take it from there.  That
→CO\u00b2 has the carbon turned to the isotope carbon14 in the upper atmosphere by
→cosmic radiation.  This causes a specific percentage of carbon in the atmosphere to
→be carbon14.\n\nNow we are carbon based life forms and we have to get the carbon we
→are built out of from some where.  We get it from eating plants, and the plants get
→it from absorbing CO\u00b2 from the air.  So so long as we are alive, we uptake new
→carbon14.  So this gives you a pretty good base line for dating.\n\nNow to fight
→arguments against carbon dating you could use the example of how we can see proton
→collisions in the LHC for sensitivity of our equipment.  Nuclear decay is very
→accurate in how fast it happens, this is why atomic clocks work to a much higher
→degree of accuracy than other methods of time keeping.  Also, you might want to
→make a general appeal for science.  Science works, that is why we have TV's, robots,
→ particle accelerators, satellites, computers, MRI and CAT scanners, nuclear power,
→etc etc.  Scientists are not just willy nilly making shit up, or these kinds of
→things wouldn't work.",
```

```
            "created_utc": 1270637661,
            "id": "c0nn9iq",
            "link_id": "t3_bne3u",
            "parent_id": "t1_c0nn5ux",
            "score": 2,
            "subreddit": "askscience",
            "subreddit_id": "t5_2qm4e"
        }
    ],
    "metadata": {
        "execution_time_milliseconds": 30.52,
        "results_returned": 1,
        "shards": {
            "failed": 0,
            "successful": 36,
            "total": 36
        },
        "size": 1,
        "sort": "asc",
        "sort_type": "created_utc",
        "timed_out": false,
        "total_results": 134785,
        "version": "v3.0"
    }
}
```

From the result returned, we can see that the first comment ever made to /r/science mentioning "science" happened on epoch date 1270637661, which translates to Wednesday, April 7, 2010 10:54:21 AM (GMT). Let's quickly go over the metadata pieces. We can see that the execution time for this search was around 30 milliseconds. There were a total of 36 shards searched and all were successful. The search did not time out (timed_out parameter) which is good. This is an attribute you may want to check if you use the API programmatically as some searches that are more complicated may sometimes time out. The total_results value is 134,785. This tells us the total number of comments in /r/askscience that mention the word science. Since we did not use the before or after parameters, this number represents the entirety of the comments made to /r/askscience.

8.4 Using the before and after parameters

Let's continue by using additional parameters to highlight the power of the search API. The before and after parameters allow you to restrict the time-frame for the search by giving an epoch timestamp for both. However, the API also understands more human-like values for the before and after parameters. You can use a number followed by the characters s,m,h,d (which stand for second, minute, hour and day) to limit the time-frame as well. Let's run through some examples.

If you wanted to do a search for "Rome" in the subreddit /r/askhistorians but limit it only to the past 30 days, you could use the after parameter with the value 30d (30 days).

Search the subreddit /r/askhistorians for comments mentioning Rome within the past 30 days

https://api.pushshift.io/reddit/search/comment/?q=rome&subreddit=askhistorians&after=30d

What if there was a recent news story three days ago, but we wanted to limit the search window between 4 days ago and 2 days ago? We could use both the before and after parameter to do so. In the next example, we will search for comments mentioning Trump that were made between 4 and 2 days ago and sort by ascending.

Search all subreddits for the term "Trump" and return comments made between 2 and 4 days ago

https://api.pushshift.io/reddit/search/comment/?q=trump&after=4d&before=2d&sort=asc

8.5 Using the fields parameter

Let's say you wanted to do a search for the last 150 comments, but you only need the author and body fields returned for each comment. Using the fields parameter, you can tell the API which pieces of information you want to filter. This is primarily to help reduce bandwidth if you are making a lot of requests and only need specific fields returned.

Here is an example using the fields parameter to search for the past 150 comments that mention "government" and only returning the author and body fields:

Search all subreddits for the term "government" and return comments with only the body and author keys

https://api.pushshift.io/reddit/search/comment/?q=government&size=150&fields=body,author

8.6 Using the author parameter

Using one of the examples above that searched for the first occurrence of the word "science" in the subreddit /r/askscience, we saw that the author of the comment was "MockDeath." What if we wanted to get the first 100 comments that "MockDeath" made to Reddit? We can use the author parameter, along with the sort and size parameters.

Search all subreddits and get the first 100 comments ever made by the user /u/MockDeath

https://api.pushshift.io/reddit/search/comment/?author=MockDeath&sort=asc&size=100

Using the aggs parameter

Aggregations is a powerful method to give summary data for a search. Using the aggs parameter, we can quickly create facets around specific parameters and see how data changes over time. The aggs parameter for comment searches accepts the following values: author, subreddit, reated_utc and link_id. We can do a lot of very cool things using this parameter, so let's dive into some examples.

9.1 Using the time frequency (created_utc) aggregation

Let's say we wanted to see the frequency of usage for the term "Trump" over time. We'd like to be able to see how many comments were posted per hour over the past 7 days for this term. Using aggregations and the aggs parameter, we can get that data quickly. Here's an example using this criteria:

Create a time aggregation using the term trump to show the number of comments mentioning trump each hour over the past 7 days

https://api.pushshift.io/reddit/search/comment/?q=trump&after=7d&aggs=created_utc&frequency=hour&size=0

We used the frequency parameter along with the aggs parameter to create hourly buckets to show the total number of comments mentioning Trump over the past 7 days. The size parameter was set to 0 because we are only interested in getting aggregation data and not comment data. The aggregation data is returned in the response under the key aggs -> created_utc. Here is a snippet of the first part of the return:

```
{
    "aggs": {
        "created_utc": [
            {
                "doc_count": 685,
                "key": 1502406000
            },
            {
                "doc_count": 1238,
                "key": 1502409600
            },
            {
```

```
                "doc_count": 1100,
                "key": 1502413200
            },
```

The doc_count value is the total number of comments containing the term "trump." The key value is the epoch time for that particular bucket. In this example, the first bucket has an epoch time of 1502406000 which corresponds to Thursday, August 10, 2017 11:00:00 PM. This key value is the beginning time of the bucket, so in this example, 685 comments contain the term "trump" between the time Thursday, August 10, 2017 11:00:00 PM and Thursday, August 10, 2017 12:00:00 PM. The frequency parameter allows you to create buckets per second, minute, hour, day, week, month, year. Using this aggregation, you could use the data to create a chart (i.e. Highcharts) and graph the activity of comments for specific terms, authors, subreddits, etc. This is an extremely powerful data analysis tool.

9.2 Using the subreddit aggregation

What if you wanted to not only get the frequency of specific comment terms over time, but also wanted to see which subreddits were the most popular for a given term over that time period? Here's an example of using the aggs parameters to show which subreddits had the most activity for a specific term.

Create a subreddit aggregation using the term trump to show the top subreddits mentioning trump over the past 7 days

https://api.pushshift.io/reddit/search/comment/?q=trump&after=7d&aggs=subreddit&size=0

Here is a snippet of the result:

```
{
    "aggs": {
        "subreddit": [
            {
                "bg_count": 66,
                "doc_count": 44,
                "key": "lovetrumpshaters",
                "score": 0.6666666666666666
            },
            {
                "bg_count": 20,
                "doc_count": 9,
                "key": "Denmark_Uncensored",
                "score": 0.45
            },
            {
                "bg_count": 51,
                "doc_count": 16,
                "key": "WhoRedditHatesNow",
                "score": 0.3137254901960784
            },
```

The subreddit aggregation will return the total number of comments in that subreddit that mention the query term (doc_count) as well as the total number of comments made to that subreddit during that time period (bg_count). This not only will show you which subreddits mentioned Trump the most often, but it also gives you normalized results so that you can also see what percentage of that subreddit's comments contained the search term. If you were to simply rank the subreddits by which subreddits mentioned the search term "trump" the most often, the results would be biased towards subreddits that also contain the most activity in general. Using this approach, you can see both the raw count and also the normalized data.

9.3 Using the submission (link_id) aggregation

The API also allows aggregations on link_id, which is another very powerful method to see which submissions are the most popular based on a specific search term. Continuing with the examples above, let's give a scenario where this would be extremely helpful. Within the past 24 hours, numerous big stories have dropped concerning Donald Trump. You would like to use the API to see which submissions are related to Trump based on the number of comments mentioning him within the submissions. We can again use the aggs parameter and set it to link_id to get this information quickly. Let's proceed with another example:

Show submissions made within the past 24 hours that mention trump often in the comments

https://api.pushshift.io/reddit/search/comment/?q=trump&after=24h&aggs=link_id&size=0

This will return under the aggs -> link_id key an array of submission objects. The doc_count gives the total number of comments for each submission that mention the search term ("trump") and the bg_count give the total number of comments made to that submission. This is a great way to quickly find submissions that are "hot" based on a specific search term or phrase.

9.4 Using the author aggregation

The API also allows you to create aggregations on authors so you can quickly see which authors make the most comments for a specific search term. Here is an example of using the author aggregation:

Show the top authors mentioning the term "Trump" over the past 24 hours

https://api.pushshift.io/reddit/search/comment/?q=trump&after=24h&aggs=author&size=0

```
{
    "aggs": {
        "author": [
            {
                "doc_count": 605,
                "key": "grrrrreat"
            },
            {
                "doc_count": 329,
                "key": "AutoModerator"
            },
            {
                "doc_count": 168,
                "key": "autotldr"
            },
            {
                "doc_count": 73,
                "key": "SnapshillBot"
            },
```

The author aggregation will show you which authors make the most comments containing a specific query term. From the example above, a lot of the top authors mentioning the term "Trump" are actually bots.

9.5 Combining multiple aggregations at once

Using the aggs parameter, you can combine multiple aggregations and get a lot of facet data for a specific term. Using the examples above, we can combine all of the calls into one call and show the top submissions over the past 24 hours,

the frequency of comments per hour mentioning Trump, the top authors posting about Trump and the top subreddits that have had comments made mentioning Trump.

Show aggregations for authors, submissions, subreddits and time frequency for the term "Trump" over the past 24 hours

https://api.pushshift.io/reddit/search/comment/?q=trump&after=24h&aggs=author,link_id,subreddit,created_utc&frequency=hour&size=0

Searching Submissions

To search for submissions, use the endpoint https://api.pushshift.io/reddit/search/submission/ endpoint. Let's start with a few examples and then go over the various parameters available when using this endpoint. Do to a simple search, the q parameter is used to search for a specific word or phrase. Here is an example:

Search for the most recent submissions mentioning the word "science"

https://api.pushshift.io/reddit/search/submission/?q=science

This will search for the most recent submissions with the word science in the title or selftext. The search is not case-sensitive, so it will find any occurence of science regardless of capitalization. The API defaults to sorting by the most recently made submissions first. After running this search, 25 results are returned. This is the default size for searches and can be changed by using the size parameter. This will be discussed in further detail in the parameters section. Data is returned in JSON format and results are included in the "data" key.

Search parameters for submissions

There are numerous additional parameters that can be used when performing a submission search. Let's go over each of them now and provide examples for each one.

Paramet er	Descrip tion	Default	Accepte d Values
ids	Get specifi c submiss ions via their ids	N/A	Comma-d elimite d base36 ids
q	Search term. Will search ALL possibl e fields	N/A	String / Quoted String for phrases
q:not	Exclude search term. Will exclude these terms	N/A	String / Quoted String for phrases
title	Searche s the title field only	N/A	String / Quoted String for phrases
title:n ot	Exclude search term from title. Will ex-clude these terms	N/A	String / Quoted String for phrases
selftex t	Searche s the selftex t field only	N/A	String / Quoted String for phrases
selftex t:not	Exclude search term from selftex t. Will exclude these terms	N/A	String / Quoted String for phrases
size	Number of results to return	25	Integer <= 500
fields	One return specifi c fields (comma de-limit ed)	All Fields	String or comma-d elimite d string (Multip le values allowed)
sort	Sort results in a specifi c order	"desc"	"asc", "desc"
sort_t ype	Sort by a specifi c attribu te	"created _utc"	"score" , "num_c omments ", "create d_utc"
aggs	Return aggrega tion summary	N/A	["autho r", "link_ id", "create d_utc" , "subred dit"]
author	Restric t to a specifi c author	N/A	String or comma-d elimite d string (Multip le values allowed)
subredd it	Restric t to a specifi c subredd it	N/A	String or comma-d elimite d string (Multip le values allowed)
after	Return results after this date	N/A	Epoch value or Integer + "s,m,h, d" (i.e. 30d for 30 days)
before	Return results before this date	N/A	Epoch value or Integer + "s,m,h, d" (i.e. 30d for 30 days)
score	Restric t results based on score	N/A	Integer or > x or < x (i.e. score=> 100 or score=< 25)
num_co mments	Restric t results based on number of com-ment s	N/A	Integer or > x or < x (i.e. num_co mments= >100)
over_1 8	Restric t to nsfw or sfw content	both allowed	"true" or "false"
is_vid eo	Restric t to video content	both allowed	"true" or "false"
locked	Return locked or unlocke d threads only	both allowed	"true" or "false"
stickie d	Return stickie d or unstick ied content only	both allowed	"true" or "false"
spoiler	Exclude or include spoiler s only	both allowed	"true" or "false"
contest _mode	Exclude or include content mode submiss ions	both allowed	"true" or "false"
frequen cy	Used with the aggs paramet er when set to created _utc	N/A	"second ", "minute ", "hour", "day"

11.1 Get all comment ids for a particular submission

This call is very helpful when used along with Reddit's API. When there are large submissions with thousands of comments, it is often difficult to get all the comment ids for a submission. This call will return an array of comment ids when a submission id is passed to it. The endpoint is: https://api.pushshift.io/reddit/submission/comment_ids/{base36 submission id}

This call will return a data key with an array of comment ids. You can then retrieve the actual comment information from this API or the Reddit API. If the submission is fairly new, it is better to use the Reddit API to get the most current score for the comments.

Retrieve all comment ids for a submission object

https://api.pushshift.io/reddit/submission/comment_ids/6uey5x

List of Endpoints

Endpoin t	Descrip tion	Status
/reddit /search /commen t/	Search Reddit Comment s	Active
/reddit /search /submis sion/	Search Reddit Submiss ions	Active
/reddit /submis sion/co mment_ ids/{ba se36-su bmis-sio n-id}	Retriev e comment ids for a submiss ion object	Active
/reddit /analyz e/user/ {author -name}	Analyze a Reddit user's activit y	In Developm ent
/reddit /term/f requenc y/{term }	Analyze a term based on activit y	In Developm ent
/reddit /search /all/	Search Both Comment and Submiss ions	In Developm ent
/reddit /trendi ng/peop le	Find out who is trendin g on Reddit	In Developm ent
/reddit /search /links	Find releven t links being shared on Red-dit	In Developm ent

CHAPTER 13

To be continued (Currently under active development) . . .