qBittorrent
Torznab Search Engine

V1.0 - 18/05/2021

Ngosang

Index

Goal 2
Use case 3
API integration 7
Example request (OK) 7
Example request (KO, bad params) 8
Example request (KO, bad URL) 8
Response fields 8
Categories 9
Future 10

Goal

The current implementation of qBittorrent's search engine is in bad shape and it's hard to maintain.

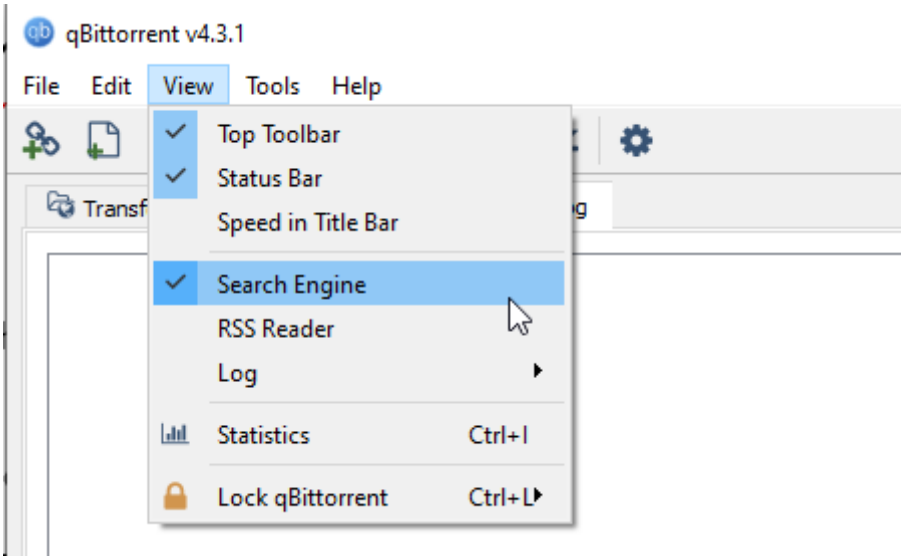
The goal is to replace the search engine in qBittorrent with a Torznab client implementation.

With that solution qBittorrent will delegate the search function to a 3rd party program. qBittorrent won't have to:

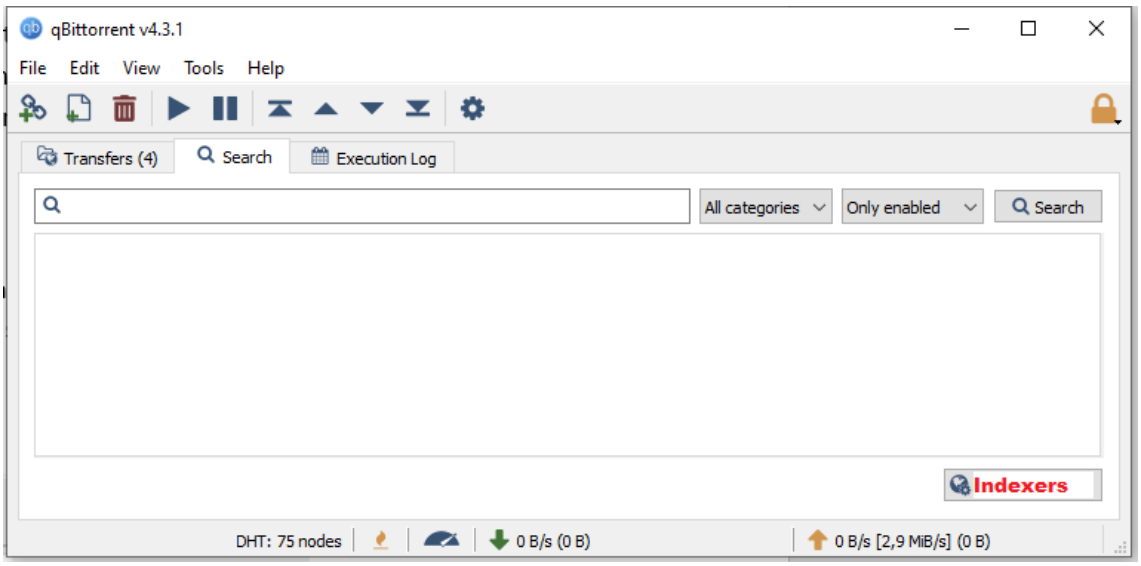
- maintain integrations with external torrents sites
- maintain Python code, search plugins or third-party plugins
- download Python installer or test Python installation
- run Python scripts (call search plugins)

Use case

In a clean installation of qBittorrent the Search functionality is disabled by default. The feature is optional. The user can enable it in the menu. This is the current behavior.

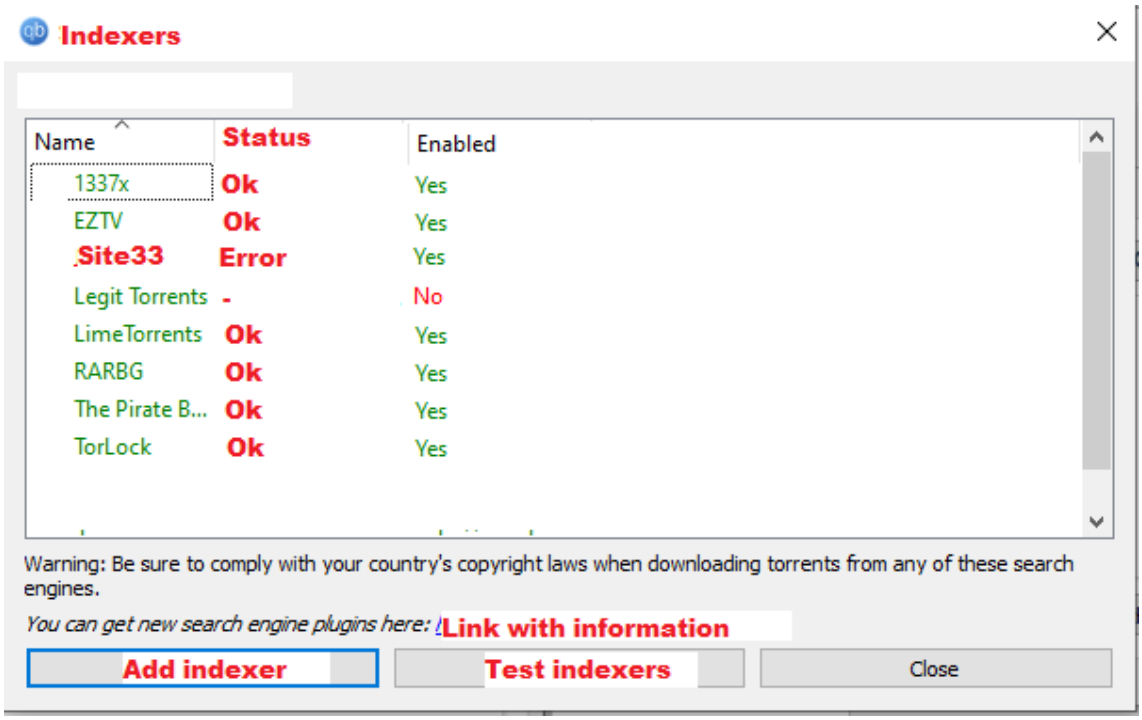


The search tab will keep the same layout as it has now. The categories could be removed in the first iteration but they are supported in Torznab and they are very useful.



By default, there are no Indexers configured. The user will have to configure at least 1 before searching. We can disable the Search button, show a message or something to alert the user.

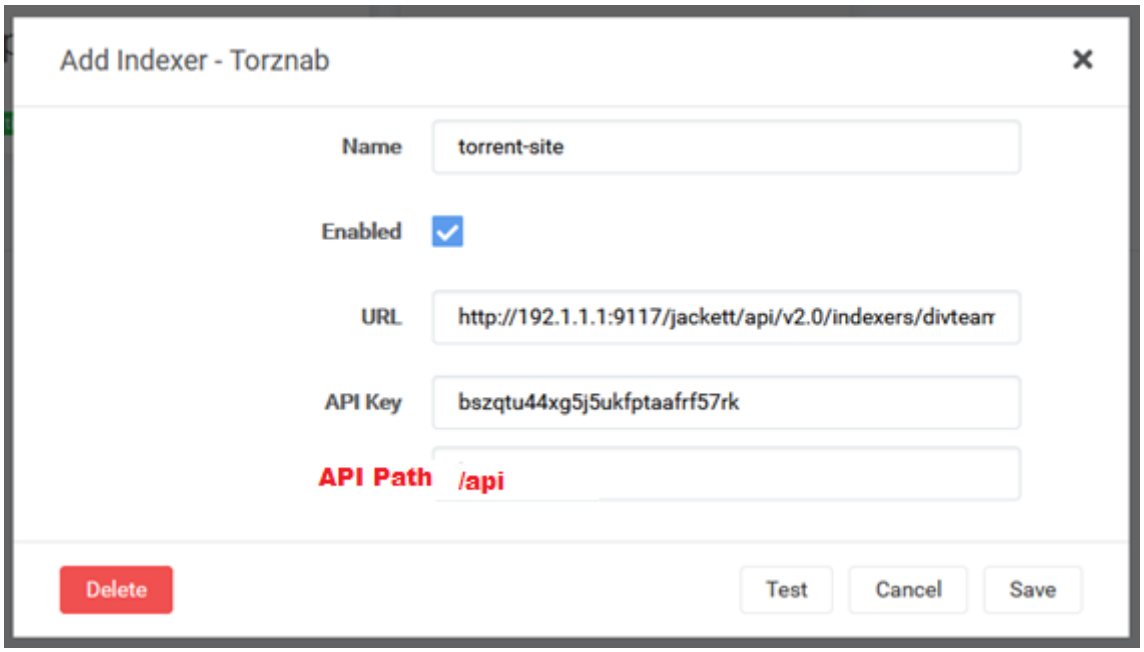
If the user click in the “indexers” button, we open a modal dialog to configure 1 or more indexers.



There could be 0…N indexers configured. Each indexer could be enabled or disabled by the user. Disabled means: 1) it won't be available for search. 2) it won't be tested if you click test indexers.

The status column and the "test indexers" button is a nice to have feature and it can be included in the future. If the user clicks in "test indexers" button, we will make a search with q="" for enabled indexer and check the HTTP response code. We update the status column.

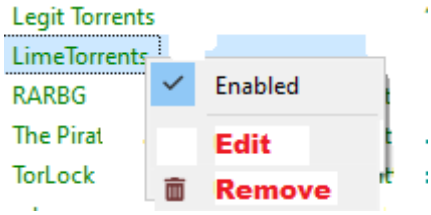
The user will be able to add 1 indexer clicking in "Add indexer" button, we show a configuration dialog.



Enabled checkbox and delete button could be in the contextual menu as it's now. All 4 text fields are mandatory. The API path should be filled with "/api" by default and the others

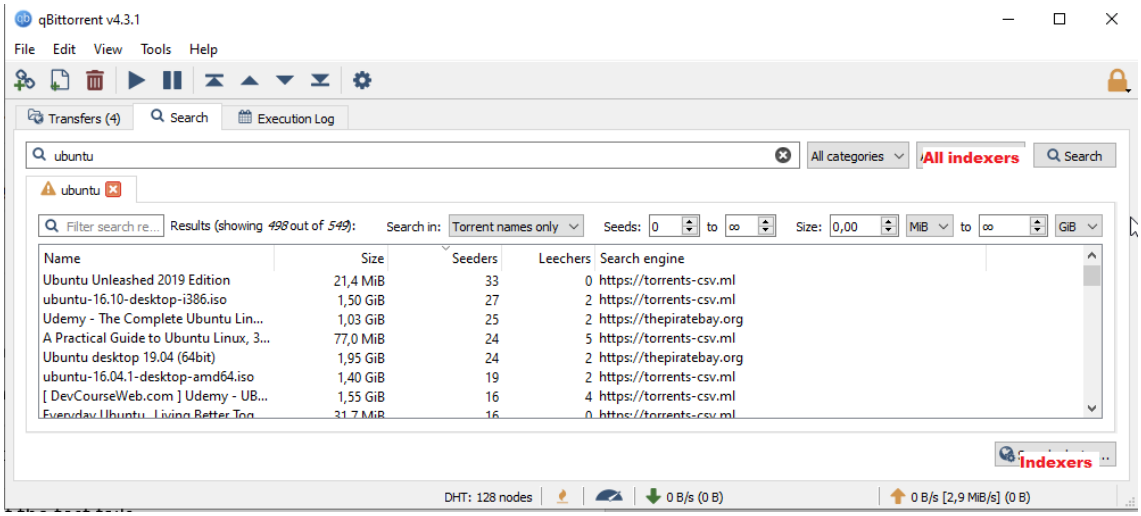
empty. The name has to be unique to avoid problems with other configured indexers. The “test” button is not mandatory but It’s a good idea. In the user clicks the “test” button we make a query with the configured fields and the search term q=”” and check the HTTP response code. The user can click save even if the test fails.

If the user click in one configured indexer with the right button, the contextual menu shows these options:



If the user double clicks in one configured indexer, we show the edit dialog. The edit dialog is the same as “add indexer”. The user can change the fields, test the indexer and save.

After there is at least 1 indexer configured. The user can search on them. The search layout is unchanged.



When the user click search in all indexers:

1) We create a thread pool or threads to make HTTP requests in parallel. We create a new tab. We add a spinner or some notification for the user, the search is in progress.

2) There is 1 HTTP request for each enabled indexer.

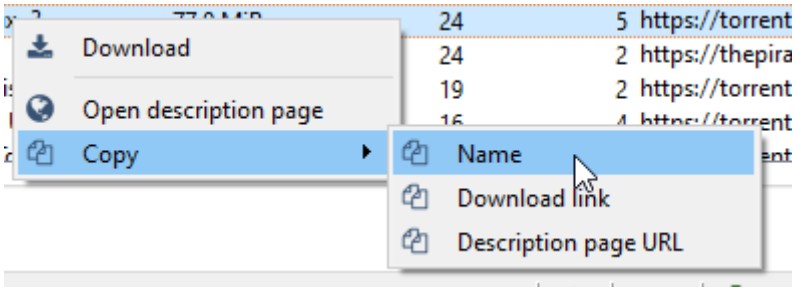
3) The response for each request could be OK or Error. We check the HTTP response code and response XML to check for errors.

3a) If the response is OK we parse the XML response and we add the results to the list without waiting for other requests to finish. We can add a log trace with the indexer name, query and number of results.

3b) If the response is Error we parse the response, log the error in the traces, show a warning icon or the error in the status bar. Errors in one response don’t affect the global search, we keep searching.

4) After all requests end, we change the spinner icon to notify the user.

Contextual menu in the search results is unchanged.



API integration

Torznab Spec => https://torznab.github.io/spec-1.3-draft/external/newznab/api.html
Torznab Spec 2 => https://nzbdrone.readthedocs.io/Implementing-a-Torznab-indexer/

We only have to implement Torznab search request:

https://torznab.github.io/spec-1.3-draft/external/newznab/api.html#search

GET {{API URL}}{{API path}}?t=search&apikey={{API Key}}&q={{search term}}

Search with empty "q" parameter returns latest torrents and it can be used to test the indexer.

Example request (OK)

GET
http://192.168.1.1:9117/jackett/api/v2.0/indexers/iptorrents/results/torznab/api?t=search&apikey=bszqtu44xg5j5ukfptd&q=linux

Response **200 OK**

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="1.0" xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:torznab="http://torznab.com/schemas/2015/feed">
  <channel>
    <atom:link href="http://192.168.1.1:9117/jackett/" rel="self"
type="application/rss+xml" />
    <title>IPTorrents</title>
    <description>Always a step ahead.</description>
    <link>https://iptorrents.com/</link>
    <language>en-us</language>
    <category>search</category>
    <item>
      <title>Link-Assistant v6 38 11 Linux Multilingual-NGEN</title>
      <guid>https://iptorrents.com/t/4330804</guid>
      <jackettindexer id="iptorrents">IPTorrents</jackettindexer>
      <comments>https://iptorrents.com/t/4330804</comments>
      <pubDate>Sat, 15 May 2021 14:07:02 +0200</pubDate>
      <size>230686720</size>
      <files>49</files>
      <grabs>27</grabs>
      <description> Uploaded by: TvTeam</description>
      <link>http://192.168.1.1:9117/jackett/dl/iptorrents/?jackett_
ZLcjd6OE1JRmJDXzRFZXNYcnRIVHlGMF96eWh2MWhkMXVwVVlYOVQ3dUo5YVI0eC1RUm
5wUjdWMFJWNDI3NFljUnBLYmFtMThhYm50T3hUdWZxRTdXMFFCSnRtWTBVeG9mMVBma1
9vQQ&amp;file=Link-Assistant+v6+38+11+Linux+Multilingual-NGEN</link>
      <category>4010</category>
      <category>100001</category>
      <enclosure
url="http://192.168.1.1:9117/jackett/dl/iptorrents/?jackettdWMFJWNDI
3NFljUnBLYmFtMThhYm50T3hUdWZxRTdXMFFCSnRtWTBVeG9mMVBma19vQQ&amp;file
=Link-Assistant+v6+38+11+Linux+Multilingual-NGEN" length="230686720"
type="application/x-bittorrent" />
      <torznab:attr name="category" value="4010" />
      <torznab:attr name="category" value="100001" />
      <torznab:attr name="seeders" value="13" />
      <torznab:attr name="peers" value="13" />
      <torznab:attr name="minimumratio" value="1" />
      <torznab:attr name="minimumseedtime" value="1209600" />
      <torznab:attr name="downloadvolumefactor" value="1" />
      <torznab:attr name="uploadvolumefactor" value="1" />
```

```
    </item>
    <item>
      <title>Battle Axe Linux-rG</title>
      <guid>https://iptorrents.com/t/4330425</guid>
      <jackettindexer id="iptorrents">IPTorrents</jackettindexer>
      <comments>https://iptorrents.com/t/4330425</comments>
      <pubDate>Sat, 15 May 2021 09:19:02 +0200</pubDate>
      <size>137363456</size>
      <files>16</files>
      <grabs>30</grabs>
      <description> Uploaded by: TvTeam</description>

<link>http://192.168.1.1:9117/jackett/dl/iptorrents/?jackettxBVktj&a
mp;file=Battle+Axe+Linux-rG</link>
      <category>4050</category>
      <category>100045</category>
      <enclosure
url="http://192.168.1.1:9117/jackett/dl/iptorrents/?jackett_
&amp;file=Battle+Axe+Linux-rG" length="137363456"
type="application/x-bittorrent" />
      <torznab:attr name="category" value="4050" />
      <torznab:attr name="category" value="100045" />
      <torznab:attr name="seeders" value="12" />
      <torznab:attr name="peers" value="12" />
      <torznab:attr name="minimumratio" value="1" />
      <torznab:attr name="minimumseedtime" value="1209600" />
      <torznab:attr name="downloadvolumefactor" value="1" />
      <torznab:attr name="uploadvolumefactor" value="1" />
    </item>
  </channel>
</rss>
```

Example request (KO, bad params)

GET
http://192.168.1.1:9117/jackett/api/v2.0/indexers/iptorrents/results/torznab/api?t=search&apikey=bad&q=linux

Response **200 OK**

```
<?xml version="1.0" encoding="UTF-8"?>
<error code="100" description="Invalid API Key" />
```

Example request (KO, bad URL)

GET http://192.168.1.1:9117/jackett/api/v2.0/bad

Response **!= 200**

```
Body can be anything, not XML.
```

Response fields

Torznab fields are well defined in the specification but most of them are optional. Try to by permissive when parsing the response. Jackett follows the specification closely but other software doesn't. Response items could be extended with more fields and attributes and we have to ignore them.

The field mapping for qBittorrent will be: (qBittorrent UI => XML)

- Name => <title> (**mandatory**)
- Size => <size>
- Seeders => <torznab:attr name="seeders" value="13" />
- Leechers => subtracts <torznab:attr name="peers" and <torznab:attr name="seeders"
- Search engine / indexer => configured indexer name, not from the XML
- Download URL => <link> (**mandatory**)
- Description URL => <comments>

Jackett can provide .torrent file and magnet links for each result but this is not standard.

Be careful with optional fields.

Categories

Torznab defines a complex hierarchy of categories but we only have to implement root level.

Categories are numeric and they can be included in the search with the query parameter **&cat=4000** you can include several categories separated by comma **&cat=4000,5000**

Most categories are part of the Torznab standard and the numbers are fixed.

The category mapping for qBittorrent will be:

- All => No include the &cat= query parameter in the request
- "Console" 1000
- "Movies" 2000
- "Audio" 3000
- "PC" 4000
- "TV" 5000
- "XXX" 6000
- "Books" 7000
- "Other" 8000

The complete list of Torznab categories are here => https://github.com/Jackett/Jackett/blob/aca4a16baeabacffc9e322eb91c3a511e7da0ba3/src/Jackett.Common/Models/TorznabCatType.cs

Future

Some features for the future:

- Torznab specification allows more search parameters. For example, IMDB search, only torrents with seeds.
- Torznab response includes more fields that we are not showing in the UI. For example, published date, number of files, category, number of downloads, poster, info-hash, private tracker tags (freeleech). See the API response.
- Use the torrent info-hash to group results form different indexers into 1 result.
- Add more advanced features in indexer configuration. Example screenshot form Radarr.

Edit Indexer - Torznab

Enable Interactive Search ☑ Will be used when interactive search is used

URL 17/jackett/api/v2.0/indexers/divteam/results/torznab/

API Path /api

Path to the api, usually /api

Multi Languages

What languages are normally in a multi release on this indexer?

API Key bszqtu44xg5j5ukfptaafrf

Categories 2000

Drop down list, leave blank to disable all categories

Additional Parameters

Additional Newznab parameters

Remove year from search string ☐ Should Radarr remove the year after the title when searching this indexer?

Minimum Seeders 1

Minimum number of seeders required.

Seed Ratio

The ratio a torrent should reach before stopping, empty is download client's default

Seed Time minutes

The time a torrent should be seeded before stopping, empty is download client's default

Required Flags

What indexer flags are required?
More Info

Indexer Priority 25

Indexer Priority from 1 (Highest) to 50 (Lowest). Default: 25.

Delete | Test | Cancel | Save