

[⌕ Back to API Docs \(/help/api/rest/v4\)](#)

Legal Search API

/api/rest/v4/search/

Use this API to search case law, PACER data, judges, and oral argument audio recordings.

To get the most out of this API, see our coverage (/help/coverage/) and advanced operators (/help/search-operators/) documentation.

Basic Usage

This API uses the same GET parameters as the front end of the website. To use this API, place a search query on the front end of the website. That will give you a URL like:

```
https://www.courtlistener.com/q=foo
```

To make this into an API request, copy the GET parameters from this URL to the API endpoint, creating a request like:

```
curl -X GET \
  --header 'Authorization: Token 783e33a9097e90020f4dffddf0c738d9f360b340' \
  'https://www.courtlistener.com/api/rest/v4/search/?q=foo'
```

That returns:

```
{
  "count": 2343,
  "next": "https://www.courtlistener.com/api/rest/v4/search/?cursor=cz0yMzUuODcxMjUmcz04MDUzNTUzJnQ9byZkPTIwMjQtMDktMTY%3D&q=foo",
  "previous": null,
  "results": [
    {
      "absolute_url": "/opinion/6613686/foo-v-foo/",
      "attorney": "",
      "caseName": "Foo v. Foo",
      "caseNameFull": "Foo v. Foo",
      "citation": [
        "101 Haw. 235",
        "65 P.3d 182"
      ],
      "citeCount": 0,
      "cluster_id": 6613686,
      "court": "Hawaii Intermediate Court of Appeals",
      "court_citation_string": "Haw. App.",
      "court_id": "hawapp",
      "dateArgued": null,
      "dateFiled": "2003-01-10",
    }
  ]
}
```

That's the simple version. Read on to learn the rest.

Understanding the API

Unlike most APIs on CourtListener, this API is powered by our search engine, not our database.

This means that it does not use the same approach to ordering, filtering, or field definitions as our other APIs, and sending an HTTP `OPTIONS` request won't be useful.

Setting the Result type

The most important parameter in this API is `type`. This parameter sets the type of object you are querying:

Type	Definition
<code>o</code>	Case law opinions
<code>r</code>	List of Federal cases (dockets) with up to three nested documents. If there are more than three matching documents, the <code>more_docs</code> field for the docket result will be true.
<code>rd</code>	Federal filing documents from PACER
<code>d</code>	Federal cases (dockets) from PACER
<code>p</code>	Judges
<code>oa</code>	Oral argument audio files

For example, this query searches case law:

```
https://www.courtlistener.com/q=foo&type=o
```

And this query searches federal filings in the PACER system:

```
https://www.courtlistener.com/q=foo&type=r
```

If the `type` parameter is not provided, the default is to search case law.

Ordering Results

Each search `type` can be sorted by certain fields. These are available on the front end in the ordering drop down, which sets the `order_by` parameter.

If your sorting field has null values, those results will be sorted at the end of the query, regardless of whether you sort in ascending or descending order. For example if you sort by a date that is null for an opinion, that opinion will go at the end of the result set.

Filtering Results

Results can be filtered with the input boxes provided on the front end or by advanced query operators (</help/search-operators/>) provided to the `q` parameter.

The best way to refine your query is to do so on the front end, and then copy the GET parameters to the API.

Fields

Unlike most of the fields on CourtListener, many fields on this API are provided in camelCase instead of snake_case. This is to make it easier for users to place queries like:

```
caseName:foo
```

Instead of:

```
case_name:foo
```

All available fields are listed on the advanced operators help page (</help/search-operators/>).

To understand the meaning of a field, find the object in our regular APIs that it corresponds to, and send an HTTP `OPTIONS` request to the API.

For example, the `docketNumber` field in the search engine corresponds to the `docket_number` field in the `docket` API, so an HTTP `OPTIONS` request to that API returns its definition:

```
curl -X OPTIONS \
  --header 'Authorization: Token 783e33a9097e90020f4dffddf0c738d9f360b340' \
  "https://www.courtlistener.com/api/rest/v4/dockets/" \
  | jq '.actions.POST.docket_number'
```

After filtering through `jq` (<https://github.com/jqlang/jq>), that returns:

```
{
  "type": "string",
  "required": false,
  "read_only": false,
  "label": "Docket number",
  "help_text": "The docket numbers of a case, can be consolidated and quite long. In some instances they are too long to be indexed by postgres and we store"
```

Highlighting Results

To enhance performance, results are not highlighted by default. To enable highlighting, include `highlight=on` in your request.

When highlighting is disabled, the first 500 characters of snippet fields are returned for fields `o`, `r`, and `rd`.

Result Counts

`type=d` and `type=r` use cardinality aggregation to compute the result count. This enhances performance, but has an error of $\pm 6\%$ if results are over 2000. We recommend noting this in your interface by saying something like, "About 10,000 results."

Special Notes

A few fields deserve special consideration:

1. As in the front end, when the `type` is set to return case law, only published results are returned by default. To include unpublished and other statuses, you need to explicitly request them.
2. The `snippet` field contains the same values as are found on the front end. This uses the HTML5 `<mark>` element to identify up to five matches in a document.

This field only responds to arguments provided to the `q` parameter. If the `q` parameter is not used, the `snippet` field will show the first 500 characters of the `text` field.

This field only displays Opinion text content.

Monitoring a Query for New Results

To monitor queries for new results, use the Alert API (</help/api/rest/search/>), which will send emails or webhook events when there are new results.

Please Support Open Legal Data

These services are sponsored by Free Law Project (<https://free.law>) and users like you. We provide these services in furtherance of our mission to make the legal sector more innovative and equitable.

We have provided these services for over a decade, and we need your contributions to continue curating and enhancing them.

Will you support us today by becoming a member?

♥ Join FLP (<https://donate.free.law/forms/membership>)