



Pagination

ON THIS PAGE

- { Using cursor pagination
- { Using offset pagination
- { Offset pagination limit
- { Other pagination methods

Most list endpoints support two methods of paginating through results—cursor pagination and offset pagination. See the API documentation for an endpoint to determine which pagination methods it supports. If no pagination method is specified, the endpoint only supports offset pagination.

Zendesk recommends using cursor pagination instead of offset pagination where possible. Cursor pagination provides greatly improved performance when retrieving extremely large record sets. See [Comparing Cursor Pagination and Offset Pagination](#) for further details on the differences between the pagination methods.

Using cursor pagination

To use cursor pagination, include the `page[size]` parameter in the request parameters. This parameter is also used to specify the number of items to return per page. Most endpoints limit this to a maximum of 100. If you omit the `page[size]` parameter, offset pagination is used. See the API documentation for the specific resource.

Example

For example, a request to the tickets endpoint with the URL `https://example.zendesk.com/api/v2/tickets.json?page[size]=100` would return a response with the following format:

```
1  "tickets": [ ... ],
2  "meta": {
3    "has_more": true,
4    "after_cursor": "xxx",
5    "before_cursor": "yyy"
6  },
7  "links": {
8    "next":
9    "https://example.zendesk.com/api/v2/tickets.json?page[size]=100&page[after]=xxx"
10   ,
11   "prev":
12   "https://example.zendesk.com/api/v2/tickets.json?
13   page[size]=100&page[before]=yyy"
14 }
15 }
```

The values `xxx` and `yyy` are placeholder values that represent cursors.

Use the URL in the `links[next]` property to retrieve the next page of results. Using the above example, make a request to `https://example.zendesk.com/api/v2/tickets.json?page[size]=100&page[after]=xxx` to retrieve the next page of results.

In the Help Center API, the `next` link is also available in the "Link" HTTP response header. Example:

```
1  Link: <https://example.com/api/v2/help_center/en-us/articles?page%5Bafter%5D=xxx&
2  page%5Bsize%5D=100>; rel="next"
```

Repeat the above steps until the `meta[has_more]` property is false. This indicates there are no further records and you should stop paginating.

In the Help Center API, you can also paginate backwards. Follow the `links[last]` link to jump to the final page. Then, as long as `has_more` is true, follow the `prev` link to step backwards one page at a time until `has_more` is false. Like the `next` link, the `prev` link is also available in the "Link" HTTP response header.

For more detailed instructions and code examples, see [Paginating Through Lists Using Cursor Pagination](#).

Using offset pagination

Note: Beginning August 15, 2023, offset-based pagination requests over the first 10,000 records (100 pages) will result in a "400 Bad Request" error message. For more information, see [New limits for offset-based pagination](#).

If you do not opt-in to cursor pagination as described above, offset pagination will be used by default. A limit may apply when using offset pagination with high-volume requests. See [Offset Pagination limit](#).

By default, most list endpoints return a maximum of 100 records per page. You can change the number of records on a per-request basis by passing a `per_page` parameter in the request URL parameters. Example: `per_page=50`. However, you can't exceed 100 records per page on most endpoints.

When the response exceeds the per-page maximum, you can paginate through the records by incrementing the `page` parameter. Example: `page=3`. List results include `next_page` and `previous_page` URLs in the response body for easier navigation:

```
1  {
2    "users": [ ... ],
3    "count": 1234,
4    "next_page": "https://account.zendesk.com/api/v2/users.json?page=2",
5    "previous_page": null
6  }
```

Stop paging when the `next_page` attribute is `null`. For more information, see [Paginating Through Lists Using Offset Pagination](#)

Inaccuracies may be introduced in pages because of the limitations of API pagination. For more information and options, see [Limitations of API pagination](#).

Some lists can be ordered by transmitting a `sort_order=desc` or `sort_order=asc` parameter to the endpoint. Whether a specific list can be ordered is specified in the documentation for that specific resource.

Some endpoints limit the page number, or rate limit requests above a certain page number. See the documentation for the specific endpoint.

Offset pagination limit

Requests that use offset pagination are limited to the first 100 pages and 10,000 resources.

If a request exceeds this limit, you'll get a **400** HTTP status code. The response body will include details about switching from offset- to cursor-based pagination.

Cursor-based pagination does not have any page depth or resources limit.

Other pagination methods

Certain endpoints use different pagination methods:

- Incremental export endpoints use a variant of cursor pagination with some differences. See [Cursor-based incremental exports](#) in [Using the Incremental Exports API](#) article for more information.
- The List All Ticket Audits endpoint uses a variant of cursor pagination with some differences. See [Pagination](#) in the List All Ticket Audits documentation.

Join our developer community

 [Forum](#)  [Blog](#)  [Slack](#)

Zendesk 181 Fremont Street, 17th Floor, San Francisco, California 94105

[Privacy Policy](#)  [Terms & Conditions](#)  [System Status](#)  [Cookie Settings](#)