Pagination　　　　　　　　　　　　　　　　　⌄

Pagination

Endpoints that return a list of resources (such as [List services](#) and [List deploys](#)) use **cursor-based pagination** to keep response sizes manageable. If your first request to a list endpoint doesn't return all results, you can send a followup request for the next "page" of results based on a cursor you provide.

How to use

To fetch paginated results, you provide the following query parameters in your requests to a list endpoint:

Parameter	Description
`limit`	The maximum number of results to return in a single response (i.e., the size of a single "page"). Default: `20` Max: `100`
`cursor`	A unique string that corresponds to a position in the result list. If provided, the endpoint returns results that appear *after* the corresponding position.

When fetching your first "page" of results from a list endpoint, you *don't* include the `cursor` query parameter. The endpoint returns a list of resource objects, each paired with its cursor:

JSON

```json
[
  {
    "cursor": "cfQ74cE2sDI=",
    "service": {
      "id": "srv-xxxxx",
      "name" "Test Service",
      ...
    }
  },
```

```
    {
      "cursor": "mpFjFKeYgnw=",
      "service": {
        "id": "srv-xxxxx",
        "name" "A Second Service",
        ...
      }
    },
  ]
```

Note that `cursor` is not a *member* of the resource object, but instead is a sibling field.

If additional results are available, you can send another request to the endpoint. This time, set the `cursor` query parameter to the cursor of the *last* resource returned in the *previous* response, like so:

cURL

```
curl --request GET \
     --url 'https://api.render.com/v1/services?cursor=mpFjFKeYgnw=' \
     --header 'Accept: application/json' \
     --header 'Authorization: Bearer API_KEY_GOES_HERE'
```

This time, the endpoint returns results that immediately follow the cursor's position in the list. You can repeat this process until you've fetched all results.

Updated 11 months ago