



Rate Limiting

To promote stability of Render's systems, the REST API's endpoints are rate limited. Sending too many requests to an endpoint will result in a 429 error being returned.

Current Rate Limits

Rate limits are shared for a set of endpoints as noted below in the table. For instance, GET requests across different resources share the same rate limit. All but one of these rate limits apply for your user except for deploys which is per service.

API Endpoints	Rate Limit
POST /v1/services	20 / hour
PATCH /v1/services POST /v1/services/{serviceID}/deploy POST /v1/services/{serviceID}/resume POST /v1/services/{serviceID}/suspend Deploy Hooks	10 / minute / service
POST /v1/customdomain POST /v1/customdomain/verify	50 / hour
POST /v1/jobs	2000 / hour
All other POST / PATCH / DELETE	30 / minute
All other GET	400 / minute

We may reduce limits at times to prevent abuse, or ensure better overall quality of service. We can also increase rate limits on a case by case basis if you are hitting them frequently. Please contact support if you'd like to discuss your rate limits.

Handling Rate Limits


To help you keep track of how many requests you have left at a given time, every request to the Render REST API returns a set of rate limiting response headers.

```
Bash

$ curl -I https://api.render.com/v1/owners
> HTTP/1.1 200 OK
> Content-Type: application/json; charset=utf-8
> Ratelimit-Limit: 100
> Ratelimit-Remaining: 99
> Ratelimit-Reset: 45
```

Header Name	Description
Rate-Limit	Maximum requests you're permitted to make per time window
Ratelimit-Remaining	Number of requests remaining in the current rate limit window
Ratelimit-Reset	Time at which the current rate limit window resets in UTC Epoch Seconds

When querying the API, you should handle 429 status codes and setup a retry mechanism. We recommend using an exponential backoff schedule with random jitter to ensure all of your API requests can gracefully handle the failure.

 Updated about 2 months ago