





# GraphQL Admin API reference

The Admin API lets you build apps and integrations that extend and enhance the Shopify admin.

This page will help you get up and running with Shopify’s GraphQL API.

## Client libraries

Use Shopify’s officially supported libraries to build fast, reliable apps with the programming languages and frameworks you already know.

 <b>React Router</b>	-
The official package for React Router applications.	
<ul style="list-style-type: none"><li>■ <a href="#">Docs</a></li><li>■ <a href="#">npm package</a></li><li>■ <a href="#">GitHub repo</a></li></ul>	
 <b>Node.js</b>	+
 <b>Ruby</b>	+
 <b>cURL</b>	+
<b>Other</b>	+

React Router

Node.js

Ruby

cURL



Shopify uses cookies to provide necessary site functionality and improve your experience. By using our website, you agree to our [privacy policy](#) and our [cookie policy](#).

OK

---

## Authentication

All GraphQL Admin API requests require a valid Shopify access token. If you use Shopify's [client libraries](#), then this will be done for you. Otherwise, you should include your token as a `X-Shopify-Access-Token` header on all GraphQL requests.

Public and custom apps created in the Partner Dashboard generate tokens using [OAuth](#), and custom apps made in the Shopify admin are [authenticated in the Shopify admin](#).

To keep the platform secure, apps need to request specific [access scopes](#) during the install process. Only request as much data access as your app needs to work.

Learn more about [getting started with authentication](#) and [building apps](#).

[React Router](#) [Node.js](#) [Ruby](#) [cURL](#)



```
1 import { authenticate } from "../shopify.server";
2
3 export async function loader({request}) {
4   const { admin } = await authenticate.admin(request);
5   const response = await admin.graphql(
6     `query { shop { name } }`,
7   );
8 }
```

---

## Endpoints and queries

GraphQL queries are executed by sending `POST` HTTP requests to the endpoint:

`https://{store_name}.myshopify.com/admin/api/2025-10/graphql.json`

Queries begin with one of the objects listed under [QueryRoot](#). The QueryRoot is the schema's entry-point for queries.

Queries are equivalent to making a `GET` request in REST. The example shown is a query to get the ID and title of the first three products.

Explore and learn Shopify's Admin API using [GraphiQL Explorer](#). To build queries and mutations with shop data, install [Shopify's GraphQL app](#).

**POST** [https://{store\\_name}.myshopify.com/admin/api/2025-10/graphql.json](https://{store_name}.myshopify.com/admin/api/2025-10/graphql.json)

[React Router](#) [Node.js](#) [Ruby](#) [cURL](#)



```
1 import { authenticate } from "../shopify.server";
2
3 export async function loader({request}) {
4   const { admin } = await authenticate.admin(request);
5   const response = await admin.graphql(
6     `#graphql
7     query getProducts {
8       products (first: 3) {
9         edges {
10           node {
11             id
12             title
13           }
14         }
15       }
16     `
17   );
18 }
```

## Rate limits

The GraphQL Admin API is rate-limited using calculated query costs, measured in cost points. Each field returned by a query costs a set number of points. The total cost of a query is the maximum of possible fields selected, so more complex queries cost more to run.

Learn more about [rate limits](#).

**{} Request**



```
1 {
2   products(first: 1) {
3     edges {
4       node {
5         title
6       }
7     }
8   }
9 }
```

```

3      "products": {
4        "edges": [
5          {
6            "node": {
7              "title": "Hiking backpack"
8            }
9          }
10       ]
11     }
12   },
13   "extensions": {
14     "cost": {
15       "requestedQueryCost": 3,
16       "actualQueryCost": 3,
17       "throttleStatus": {
18         "maximumAvailable": 1000.0,
19         "currentlyAvailable": 997,
20         "restoreRate": 50.0
21       }
22     }
23   }
24 }

```

## Status and error codes

All API queries return HTTP status codes that contain more information about the response.

### 200 OK

GraphQL HTTP status codes are different from REST API status codes. Most importantly, the GraphQL API can return a **200 OK** response code in cases that would typically produce 4xx or 5xx errors in REST.

### Error handling

The response for the errors object contains additional detail to help you debug your operation.

The response for mutations contains additional detail to help debug your query. To access this, you must request **userErrors**.

### Properties

Shopify uses cookies to provide necessary site functionality and improve your experience. By using our website, you agree to our [privacy policy](#) and our [cookie policy](#).

## 4xx and 5xx status codes

The 4xx and 5xx errors occur infrequently. They are often related to network communications, your account, or an issue with Shopify's services.



dev docs

Apps

Storefronts

Agents

References ▾



search + assistant



Help



Sign up

`{}` Sample 200 error responses

Throttled Internal

```
1 {
2   "errors": [
3     {
4       "message": "Query cost is 2003, which exceeds the single query max cost limit (
5
6   See https://shopify.dev/concepts/about-apis/rate-limits for more information on how
7   cost of a query is calculated.
8
9   To query larger amounts of data with fewer limits, bulk operations should be used i
10  See https://shopify.dev/tutorials/perform-bulk-operations-with-admin-api for usage
11  ",
12    "extensions": {
13      "code": "MAX_COST_EXCEEDED",
14      "cost": 2003,
15      "maxCost": 1000
```

## 4xx and 5xx status codes

The 4xx and 5xx errors occur infrequently. They are often related to network communications, your account, or an issue with Shopify's services.

Many errors that would typically return a 4xx or 5xx status code, return an HTTP 200 errors response instead. Refer to the [200 OK section](#) above for details.

### 400 Bad Request

The server will not process the request.

### 402 Payment Required

The shop is frozen. The shop owner will need to pay the outstanding balance to [unfreeze](#) the shop.

### 403 Forbidden

The shop is forbidden. Returned if the store has been marked as fraudulent.

## 423 Locked

The shop isn't available. This can happen when stores repeatedly exceed API rate limits or due to fraud risk.

## 5xx Errors

An internal error occurred in Shopify. Check out the [Shopify status page](#) for more information.

### Info

Didn't find the status code you're looking for? View the complete list of [API status response and error codes](#).

#### `{}` Sample error codes

400 402 403 404 423 500

```
1 HTTP/1.1 400 Bad Request
2 {
3   "errors": {
4     "query": "Required parameter missing or invalid"
5   }
6 }
```

Was this page helpful? [Yes](#) [No](#)

### Updates

Developer changelog  
Shopify Partners Slack  
Shopify Editions

### Legal

Terms of service  
API terms of use  
Privacy policy  
Partners Program Agreement

### Business growth

Shopify Partners Program  
Shopify App Store  
Shopify Academy

### Shopify

About Shopify  
Shopify Plus  
Careers  
Investors