The Gorilla Dash REST API and GraphQL API: which one to use

Gorilla Dash has two developer APIs. How the REST API and the website GraphQL API differ, which one your project needs, and where the full documentation for each lives.

Gorilla Dash offers two APIs for developers. The **REST API** connects your organisation to other systems, such as a CRM, an automation tool or a reporting database. The **GraphQL API** powers websites that display your Gorilla Dash content. They use different credentials and different addresses, and each has its own documentation, so the first decision is which of the two you need.

Both APIs use the word **tribe**, as the rest of Gorilla Dash does. A tribe is one location, branch or business unit inside your organisation, and arguments such as `tribe_id` and `tribe_slug` select one.

The two APIs side by side

	REST API	GraphQL API
Built for	Integrations between Gorilla Dash and your other systems	Websites built for a Gorilla Dash organisation
Typical uses	Pushing leads in, reading contacts and enquiries out, keeping another system in sync	Showing pages, tribes, products, articles and menus on a website, and sending enquiries, bookings and orders back
Address	`https://api.gorilladash.com/api/v1`	`https://graphql.gorilladash.com/graphql`
Credentials belong to	Your organisation, or one tribe	One website set up in Gorilla Dash
How you sign requests	Two headers, `GorillaDash-Api-Key` and `GorillaDash-Api-Secret`, on every call	Exchange the website's credentials for an access token, then send it as `Authorization: Bearer`
Where to find the credentials	**Organisation Settings**, then **API Connections**	**Websites**, choose the website, then **Settings** and the **Website API** section
Full documentation	docs.gorilladash.com	gorilladash.com/graphql

Which one do I need?

Ask what the code you are writing is part of. If it is a website that Gorilla Dash has been set up to serve, use the GraphQL API. For anything else, use the REST API.

I want to...	Use
Send leads from a landing page, an ad platform or Zapier into Gorilla Dash	REST API
Copy contacts or enquiries into a CRM, an accounting package or a data warehouse	REST API
Show a list of tribes, with addresses and opening hours, on a site that is not a Gorilla Dash website	REST API
Build or extend the website for an organisation, using its pages, menus and products	GraphQL API

Add a store locator, a product catalogue or a blog to a Gorilla Dash website	GraphQL API
Take enquiries, appointment bookings or food orders on a Gorilla Dash website	GraphQL API

The credentials are not interchangeable. A REST API key is refused by the GraphQL API, and a website's GraphQL credentials are refused by the REST API.

Getting started with the REST API

1. Open **Organisation Settings** and choose **API Connections**. Copy the **Organisation API Key** and **Organisation API Access Token** shown under **Gorilla Dash API**. You need the **Edit Organisation Settings** permission to see this screen.
2. Open **Organisation Settings**, then **Gorilla Settings**, and check that **Disable RESTful API** is turned off.
3. Call `GET https://api.gorilladash.com/api/v1/ping` with the key in the `GorillaDash-Api-Key` header and the access token in the `GorillaDash-Api-Secret` header. A response with HTTP status 201 confirms that your credentials work.
4. Open the API reference at docs.gorilladash.com for every endpoint's fields, example requests and a console you can try calls from.

The other guides in this section explain how each REST endpoint behaves in practice. Start with the overview of the Gorilla Dash API, then the guide to API keys, which explains what an organisation key and a tribe key can each see.

Getting started with the GraphQL API

1. Open the organisation that owns the website, select **Websites**, and choose the website you are building. You need permission to manage websites in the organisation.
2. Select **Settings** and scroll to the **Website API** section. **API Website ID** is your OAuth client ID and **API Access Token** is your client secret.
3. On your website's server, send both values to `https://graphql.gorilladash.com/oauth/token` using the OAuth 2 client credentials grant. The response contains an access token that lasts 15 days.
4. Send your GraphQL queries to `https://graphql.gorilladash.com/graphql` with the header `Authorization: Bearer` followed by that access token.
5. Read the developer documentation at gorilladash.com/graphql. It has worked examples for each of these steps, and a reference for every query, mutation and type, generated from the live schema.

> **KEEP THE WEBSITE'S API ACCESS TOKEN ON YOUR SERVER**
> The API Access Token on the website's settings is a long-lived secret. Keep it in your server's environment, and never put it in JavaScript that is sent to the browser or in a public code repository. If it has been exposed, regenerate it from the same screen. Regenerating it stops every access token issued with the old one within a minute, so update your website's configuration straight away.

When an integration stops working

Each API has its own switch in **Organisation Settings**, then **Gorilla Settings**. **Disable RESTful API** refuses every REST call for the organisation and all of its tribes. **Disable client websites** refuses the GraphQL API to

every website the organisation owns. If a working integration or website suddenly stops returning data, check these two switches before anything else, then confirm that nobody has regenerated the credentials.

> **ASKING SUPPORT ABOUT A FAILING CALL**
> Tell us which API you are calling, the time of the request, and the HTTP status you received. For GraphQL, include the operation name, and give every query a name such as `query TribePage` so that we can find it. Never send us your API access token or secret.