

API Glossary

This API Glossary will give you an overview of the commonly used terms you will see when reading more about our API integrations.

API

An Application Programming Interface lets disparate technology products and services communicate with each other via the internet, allowing them to complement each other without requiring each application to know all of the internal workings of the other.

API Key

The API Key is like a password that must be sent along with every API request to validate that the system is authorized to request data from our API.

Bandwidth

The amount of data being sent over the network between the two systems communicating.

Cache

Stores previously accessed data for easier, faster access later.

HTTP Response Status Codes

When you make a request to an endpoint, the server:

1. Looks at the information sent
2. Looks for the resource referenced in the request
3. Checks if you have permission to access it
4. Returns a response with a status code

Codes can be found [here](#).

Client

The system talking to the API.

CRUD

Create, Read, Update, Delete - common operations an API allows on a specific data object.

Definitions

The syntax for any properties you intend to use in the API development.

DELETE

A request from the partner system to remove data in a system. Dispatch does not currently support any DELETE operations.

Demo Environment

Also known as a staging or sandbox environment, a demo environment allows API users to test their integrations without placing real orders in a production system.

Endpoints

A URL that a partner application can use to request specific information from the Dispatch system. For example, Dispatch has an endpoint for requesting estimates and creating new orders.

FTP (File Transfer Protocol)

A method of exchanging files from one computer to another.

API Glossary

This API Glossary will give you an overview of the commonly used terms you will see when reading more about our API integrations.

GET

An operation that requests data from a server. A GET request can return a single piece of data, like one order or a list of items.

GIT

A technical tool used for tracking versions of code.

Github

A technical tool for teams to collaborate using Git.

Headers

Headers pass additional information with a request or response.

JSON

JSON (Javascript Object Notation) is the data format used to request and return information from a system. JSON is similar to XML but preferred by many developers because it has far fewer unnecessary characters, making it faster to send over the internet.

LAMP

Linux, Apache, MySQL, and PHP - an open-source operating system, web server software, database, and programming language that are commonly combined to build web applications and power APIs. Dispatch does NOT have a LAMP stack under the hood, but our API will communicate with a system using one.

Latency

The delay before a transfer of data begins following an instruction for its transfer. Also known as the total amount of time it takes for a request to complete. If a server is taking a long time to respond to a network request, you may hear, "we have poor latency."

Navigation

Links on a homepage that break down the other pages of a website.

Parameters

The specific data sent with the API call.

PATCH

A request from the partner system to partially update data in another system.

Paths

The resource URLs that API developers choose to expose their API. Each path can have a GET/PUT/POST/DELETE HTTP action defined.

POST

A request from the partner system that creates new data in the Dispatch system. For example, creating a new order.

Properties

Name value pairs that can be used in the API functionality.

API Glossary

This API Glossary will give you an overview of the commonly used terms you will see when reading more about our API integrations.

PUT

A request from the partner system to change data in a system. For example, editing an order that has been placed.

REST

REST (Representational State Transfer) APIs rely on an industry-standard set of architectural principles to guide the creation and use of the API. These include having stateless calls, returning standard error codes, and supporting endpoints for GET, POST, PUT, and DELETE operations. RESTful APIs power approximately 70% of the web.

SDK (Software Development Kit)

A kit or set of software tools, guidelines, and programs used by programmers and developers to create applications for a specific platform. Similar to a LIBRARY, but typically SDKs include more bells and whistles. Dispatch does not provide an SDK for interacting with the Dispatch APIs.

Sandbox/Staging

See DEMO.

Server

A computer that provides data to other computers.

SOAP

SOAP (Simple Object Access Protocol) is a type of API protocol that allows communication between devices that run Windows or Linux via XML popular in the enterprise space. SOAP is an older and stricter protocol than REST. Dispatch does NOT currently support a SOAP API.

Status Page

A status page provides an overview of the system status of an application or suite of applications.

Swagger

A tool for generating API documentation based on the code that powers the API.

Tags

Tags are metadata information about the API that consumers can use to search for the API.

Throttling/Rate Limiting

A tactic used by most API servers to keep traffic at a reasonable level. The API will start rejecting requests when the rate exceeds a certain amount.

Uptime

The time a server is in operation.

Webhook

Webhooks allow the Dispatch application to update data within the Partner application via calls to update data within that system, usually as a reaction to a specific, discrete event. Webhook APIs function similarly to a traditional REST API but in the opposite direction. In Dispatch's case, an example Webhook would update the status of a delivery within the partner's tool. Users configure their webhooks within Developer Settings. A typical API polls a system for data. A webhook allows that system to push updates out to anyone configured to listen on the other side.