Key Concepts

- Routes & Endpoints
- Requests
- Responses
- Schema
- Controller Classes

In this page we'll break down some of the key concepts and terms associated with the REST API: **Routes & Endpoints, Requests**, **Responses**, **Schema**, and **Controller Classes**. Each of these concepts play a crucial role in understanding, using, and extending the WordPress REST API, and each is explored in greater depth within this handbook.

Routes & Endpoints

In the context of the WordPress REST API a **route** is a URI which can be mapped to different HTTP methods. The mapping of an individual HTTP method to a route is known as an **endpoint**.

As an example, if we make a `GET` request to the URI `http://oursite.com/wp-json/` we are returned a JSON response showing what routes are available, and what endpoints are available within each route.

`/wp-json/` is a route, and when that route receives a `GET` request then that request is handled by the endpoint which displays what is known as the index for the WordPress REST API.

The route `wp-json/wp/v2/posts` by contrast has a `GET` endpoint which returns a list of posts, but also a `POST` endpoint which accepts authenticated requests to create new posts.

We will learn how to register our own routes and endpoints in the following sections.

[info]If you are using non-pretty permalinks, you should pass the REST API route as a query string parameter. The route `http://oursite.com/wp-json/` in the example above would hence be `http://oursite.com/?rest_route=/`.[/info]

If you get a `404` error when trying to access `http://oursite.com/wp-json/`, consider enabling pretty permalinks or try using the `rest_route` parameter instead.

Requests

A REST API request is represented within WordPress by an instance of the `WP_REST_Request` class, which is used to store and retrieve information for the current request. A `WP_REST_Request` object is automatically generated when you make an HTTP request to a registered API route.

The data specified in this object (derived from the route URI or the JSON payload sent as a part of the request) determines what response you will get back out of the API.

Requests are usually submitted remotely via HTTP but may also be made internally from PHP within WordPress plugin or theme code. There are a lot of neat things you can do using this class, detailed further elsewhere in the handbook.

Responses

Responses are the data you get back from the API. The `WP_REST_Response` class provides a way to interact with the response data returned by endpoints. Responses return the requested data, or can also be used to return errors if something goes wrong while fulfilling the request.

Schema

Each endpoint requires a particular structure of input data, and returns data using a defined and predictable structure. Those data structures are defined in the API Schema.

The schema structures API data and provides a comprehensive list of all of the properties the API can return and which input parameters it can accept.

Well-defined schema also provides one layer of security within the API, as it enables us to validate and sanitize the requests being made to the API. The Schema section explores this large topic further.

Controller Classes

Controller classes unify and coordinate all these various moving parts within a REST API response cycle. With a controller class you can manage the registration of routes & endpoints, handle requests, utilize schema, and generate API responses.

A single class usually contains all of the logic for a given route, and a given route usually represents a specific type of data object within your WordPress site (like a custom post type or taxonomy).

□□□ #2
□ □□□ □□□ 2025-06-04 23:42:29 UTC
□ □□□ □□□ 2025-06-04 23:42:29 UTC