

# 5 best practices for naming Api endpoints

#api #naming #techtalks



Ricardo Carneiro 2 · فبراير ٨ min read

Nowadays web apis are the normal way to communicate between applications or even internal applications such as micro services. Because it's an open book it allows for people for write what they want, how they want ignoring good practices and consistency. And **consistency is key!** It allows us to speak the same language, understand easily how things work, integrate with other team members and even companies.

The goal of this article is to define a set of best practices to enable much further comprehension of creating web apis. Here at [Rezult IO](#) we tend to follow the industry standards for everything we can. Procedures, methods, Apis, Protocols, etc.

Here's a couple of thoughts for best practices.

## Uris as resources as nouns

One of the most recognisable characteristics of REST is the predominant use of nouns in URIs. RESTful URIs should not indicate any kind of CRUD (Create, Read, Update, Delete) functionality.

Example: use `/api/users/{id}` instead of `/api/getUser`

## Pluralized resources



Example: use `api/users` or `/api/users/{id}/address`

## Forward slashes for hierarchy

As shown in the examples above, forward slashes are conventionally used to show the hierarchy between individual resources and collections.

Example: `/users/{id}/address` clearly falls under the `/users/{id}` resource which falls under the `/users` collection.

## Lowercase letters and dashes

By convention, resource names should use exclusively lowercase letters. Similarly, dashes (-) are conventionally used in place of underscores (\_).

Example: `/users/{id}/pending-orders` instead of `/users/{id}/Pending_Orders`

## No abridging

Avoid abridging endpoint/resource names. What with modern-day technology, there's really no need. In fact, abridged names can actually create confusion in your API, as developers struggle to guess (and sometimes understand) the names you've chosen.

Example: `/users/{id}/phone-number` instead of `/users/{id}/tel-no`

## Consistency is key!

Consistency is an endpoint naming principle that deserves special recognition. No matter how closely you follow our above suggestions, your API will always feel clumsy if names are inconsistent. Always use the same name(s) to refer to a given concept within your API.



5



2



5



that endpoint names are consistent with names used in the documentation and, if applicable, in the application itself.

Example: /users should be documented under the users resource instead of the getUser method.

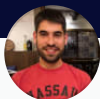
## In the end of the day

A handful of the most influential conventions and best practices for naming API endpoints — RESTful and otherwise. From the more technical issues, such as showing resource hierarchy, to the more linguistic ones, such as using American English, this concise collection of naming principles is sure to please even the strictest developers! For more fine-grained API design conventions, we recommend the API Stylebook curated by the API Handyman. It has loads of quality information, showcasing API design style guides from many companies.

### Discussion

[Subscribe](#)

DEV

[Code of Conduct](#) • [Report abuse](#)**Ricardo Carneiro**

Father of one, Founder of @ResultIO, Software Engineer 📺 PT MX 📄 result.io

[Follow](#)

5



2



5



## LOCATION

Mexico City

## JOINED

١١ نوفمبر ٢٠١٩

## Trending on DEV 🔥



How do you build Forms?

#discuss #productivity #forms



How I use Notion as a developer

#productivity #tutorial #writing



How to Use Airtable as a Production Database (Analyzing Airtable Performance)

#airtable #api #database #autocode



5



2



5

