RESTFUL Resource Naming Conventions

There are plenty of reasons to name API endpoints thoughtfully. Choosing sensible names for API endpoints can drastically smooth out the learning curve for new developers, helping them intuitively know what to look for and where to find it. With that in mind, we’re dedicating this article to more than **ten of the most effective best practices and conventions for naming API endpoints**.

### URIs as resources as nouns

Example: /users/{id} instead of /getUser

RESTful URIs should refer to a resource that is a thing (noun) instead of referring to an action (verb) because nouns have properties which verbs do not have – similar to resources have attributes.

### Pluralized resources

Example: /users (typical resource) or /users/{id}/address (singleton resource)

Next up is the question of whether resource names should be pluralized. Admittedly, this is a matter of preference; however, most API design experts would suggest you pluralize all resources unless they are singleton resources.

### Forward slashes for hierarchy

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

There aren’t any hard and fast rules [for hierarchy], only make sure the imposed structure makes sense to consumers of your services. As with everything in the craft of Software Development, naming is critical to success.

### Punctuation for lists

Example: /users/{id1},{id2} to access multiple user resources

When there is no hierarchical relationship (such as in lists), punctuation marks such as the semicolon, or, more frequently, the comma should be used.

### Query parameters where necessary

Example: /users?location=USA to find all users living in the United States

In order to [sort or filter](https://nordicapis.com/everything-you-need-to-know-about-api-pagination/) a collection, a REST API should allow query parameters to be passed in the URI.

### Lowercase letters and dashes

Example: /users/{id}/pending-orders instead of /users/{id}/Pending\_Orders

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

## General Endpoint Naming Best Practices

The above naming conventions are typically associated with REST APIs. However, there are a good handful of general naming conventions you should stick to regardless of whether your API is RESTful or not! Here are just a few of them

### American English

Example: /airplanes instead of /aeroplanes

Stick to using American English for your endpoint/resource names, since it’s the dialect your international audience of developers is likely most familiar with. The exception to this would be if your API targets a specific national audience that predominantly uses a different dialect (e.g. British or Australian).

### Intuitive names (no jargon)

Example: /users/{id}/card-number instead of /users/{id}/pan

As an extension of this, avoid using jargon. More knowledgeable developers won’t have any trouble guessing the simpler variant of a word, but the average developer won’t be able to guess a technical term they haven’t heard before!

### No abridging

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

Leave file extensions (such as .xml) out of your URIs. We’re sorry to say it, but they’re ugly and add length to URIs. If you need to specify the format of the body, instead use the Content-Type header.

### No trailing forward slash

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

The trailing slash must not have specific semantics. Resource paths must deliver the same results whether they have the trailing slash or not.

## 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.