

Open in app ↗

Medium

Search

Write



★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



REST API Naming Conventions and Best Practices



Nadin Pethiyagoda · Follow

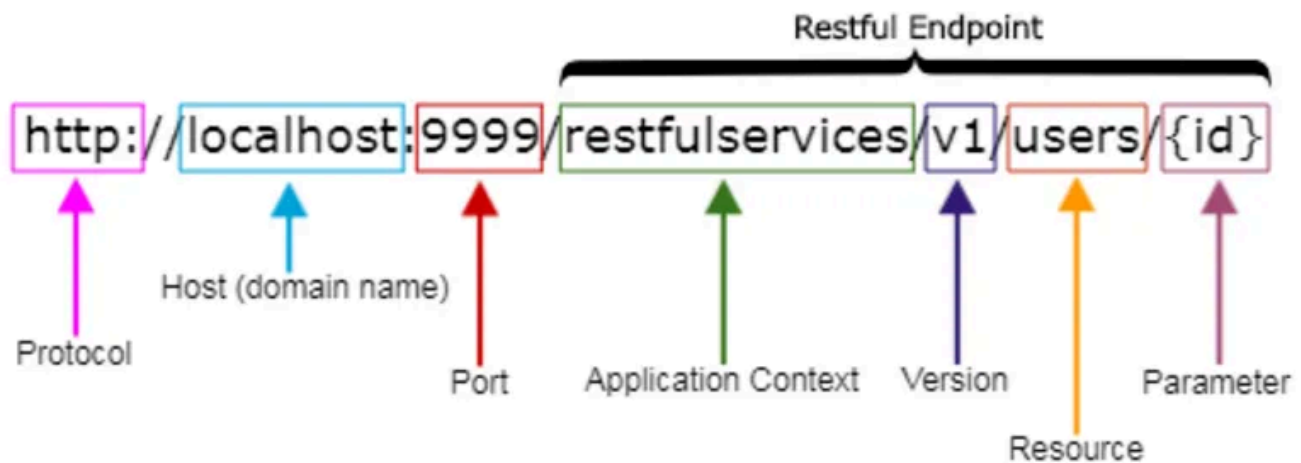
3 min read · Sep 12, 2022



320



6



The main data representation in REST is referred to as a resource. A properly named resource makes an API simple to use and intuitive. That same API, when implemented incorrectly, may feel complicated and be challenging to use and comprehend. The following article will assist you in getting started when constructing the resource URIs for your new API.

Use Nouns to represent resources / Not Verbs

Always make sure that your URIs are named with nouns to specify the resource instead of using verbs. The URIs shouldn't indicate any CRUD (Create, Read, Update, Delete) operations. Additionally avoid verb-noun combinations: hyphenated, snake_case, camelCase.

Bad examples:

```
http://api.example.com/v1/store/CreateItems/{item-id} ✗  
http://api.example.com/v1/store/getEmployees/{emp-id} ✗  
http://api.example.com/v1/store/update-prices/{price-id} ✗  
http://api.example.com/v1/store/deleteOrders/{order-id} ✗
```

Good examples:

```
http://api.example.com/v1/store/items/{item-id} ✓  
http://api.example.com/v1/store/employees/{emp-id} ✓  
http://api.example.com/v1/store/prices/{price-id} ✓  
http://api.example.com/v1/store/orders/{order-id} ✓
```

Use Pluralized Nouns for resources

Use plural when possible unless they are singleton resources.

Bad examples (Typical and Singleton resources):

`http://api.example.com/v1/store/item/{item-id}` ❌
`http://api.example.com/v1/store/employee/{emp-id}/address` ❌

Good examples (Typical and Singleton resources):

`http://api.example.com/v1/store/items/{item-id}` ✅
`http://api.example.com/v1/store/employees/{emp-id}/address` ✅

Use hyphens (-) to improve the readability of URIs

Do not use underscores. Separating words with hyphens will be easy for you and others to interpret. It is more user-friendly when it comes to long-path segmented URIs.

Bad examples:

`http://api.example.com/v1/store/vendormanagement/{vendor-id}` ❌
`http://api.example.com/v1/store/itemmanagement/{item-id}/producttype` ❌
`http://api.example.com/v1/store/inventory_management` ❌

Good examples:

`http://api.example.com/v1/store/vendor-management/{vendor-id}` ✅
`http://api.example.com/v1/store/item-management/{item-id}/product-type` ✅
`http://api.example.com/v1/store/inventory-management` ✅

Use forward slashes (/) for hierarchy but not trailing forward slash (/)

Forward slashes are used to show the hierarchy between individual resources and collections.

Bad example:

`http://api.example.com/v1/store/items/` ❌

Good examples:

`http://api.example.com/v1/store/items` ✅

Avoid using file extensions

They are unnecessary and add length and complexity to URIs.

Bad examples:

`http://api.example.com/v1/store/items.json` ❌
`http://api.example.com/v1/store/products.xml` ❌

Good examples:

`http://api.example.com/v1/store/items` ✅
`http://api.example.com/v1/store/products` ✅

Version your APIs

Always attempt to version your APIs. You can provide an upgrade path without making any fundamental changes to the existing APIs by versioning your APIs. You can also let users know that updated versions of the API are accessible at the following fully-qualified URIs.

```
http://api.example.com/v1/store/employees/{emp-id}
```

Introduction in any major breaking update can be avoided with the following /v2.

```
http://api.example.com/v1/store/items/{item-id}  
http://api.example.com/v2/store/employees/{emp-id}/address
```

Use query component to filter URI collection

You will frequently come into requirements that call for you to sort, filter, or limit a group of resources depending on a particular resource attribute. Instead of creating additional APIs, enable sorting, filtering, and pagination in the resource collection API and give the input parameters as query parameters to meet this requirement.

```
http://api.example.com/v1/store/items?group=124  
http://api.example.com/v1/store/employees?department=IT&region=USA
```

Examples:

GET — Read employee with employee id 8345

`example.com/employees/8345`

POST— Create an employee

`example.com/employees`

PUT— Update employee with employee id 8345

`example.com/employees/8345`

DELETE — Delete employee with employee id 8345

`example.com/employees/8345`

References:

REST Resource Naming Guide

In REST, the primary data representation is called resource. Having a consistent and robust REST resource naming...

restfulapi.net

RESTful API URL Naming Conventions and Best Practices

Today in this article will learn about RESTful API URL Naming Conventions and Best Practices. Today in this article, we...

www.thecodebuzz.com

Best Practices for RESTful API Design

In today's highly connected environment, RESTful APIs are dominating the fabric of the internet. Most of the...

avalides.com

Rest Api

Naming Conventions

Best Practices

Uri

Endpoints



Written by Nadin Pethiyagoda

212 Followers

Software Engineer

Follow



More from Nadin Pethiyagoda



N Nadin Pethiyagoda

Spring Web Services (SOAP)

In essence, web services are any software, application, or cloud technology that offers...

Sep 11, 2022 🖱️ 12



...



N Nadin Pethiyagoda

Dependency Management with BOM

It can be quite painful when it comes to deploying an application. It is a huge concer...

Oct 7, 2022 🖱️ 13 💬 1



...



N Nadin Pethiyagoda

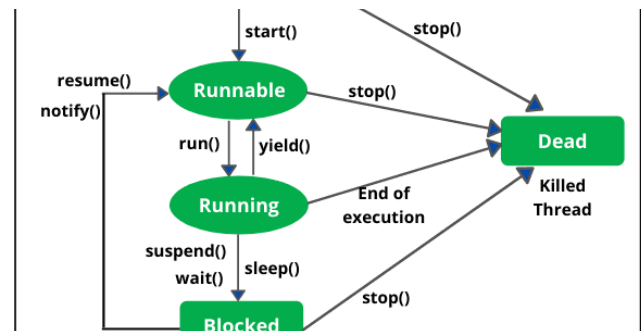
Ettercap and Wireshark: Why is HTTPS important?

This is rather an interesting topic to discuss because as engineers/senior engineers/tec...

Aug 21, 2022 🖱️ 2 💬 1



...



N Nadin Pethiyagoda

Java Threads

A thread is simply a very lightweight process (a flow of execution), which is the smallest un...

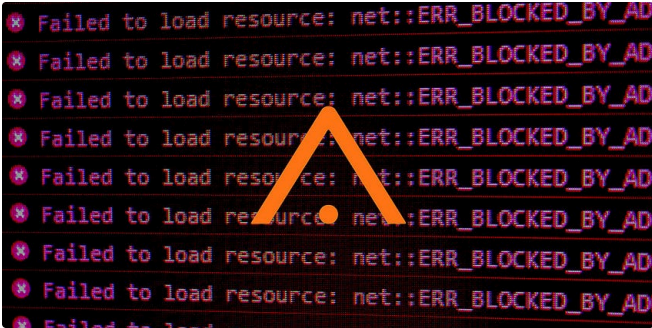
Sep 10, 2022 🖱️ 43 💬 1




...

See all from Nadin Pethiyagoda

Recommended from Medium



 Rory Murphy

API Error Handling: Techniques and Best Practices


API error handling is a critical aspect of software development, ensuring smooth...

Mar 1



 Jun 1  19K  328



 Alexander Nguyen in Level Up Coding

The resume that got a software engineer a \$300,000 job at Google.

1-page. Well-formatted.

Software Development Engineer

Mar. 2020 – May 2021

- Developed Amazon checkout and payment services to handle traffic of 10 Million daily global transactions
- Integrated Iframes for credit cards and bank accounts to secure 80% of all consumer traffic and prevent CSRF, cross-site scripting, and cookie-jacking
- Led Your Transactions implementation for JavaScript front-end framework to showcase consumer transactions and reduce call center costs by \$25 Million
- Recovered Saudi Arabia checkout failure impacting 4000+ customers due to incorrect GET form redirection

Projects


NinjaPrep.io (React)

- Platform to offer coding problem practice with built in code editor and written + video solutions in React
- Utilized Nginx to reverse proxy IP address on Digital Ocean hosts
- Developed using Styled-Components for 95% CSS styling to ensure proper CSS scoping
- Implemented Docker with Seccomp to safely run user submitted code with < 2.2s runtime

HeatMap (JavaScript)

- Visualized Google Takeout location data of location history using Google Maps API and Google Maps heatmap code with React
- Included local file system storage to reliably handle 5mb of location history data
- Implemented Express to include routing between pages and jQuery to parse Google Map and implement heatmap overlay

Lists



Natural Language Processing

1666 stories · 1242 saves

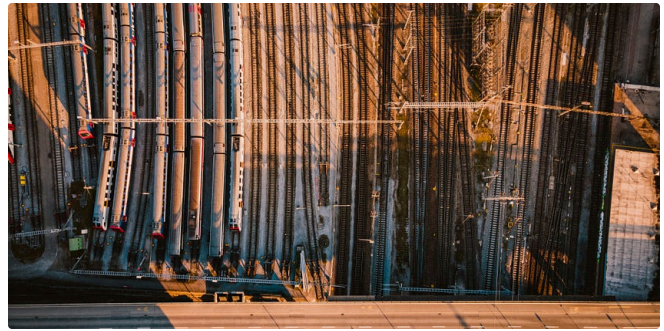


Mahtab Haider

GitHub Branching Strategy for Microservices Architecture: Best...

Introduction

Mar 13 🖱 4



Andrew Zuo

Async Await Is The Worst Thing To Happen To Programming

I recently saw this meme about async and await.



Jun 22



3K



179



Count(*)

vs

Count(1)



Vishal Barvaliya in Data Engineer

Count(*) vs Count(1) in SQL.

If you've spent any time writing SQL queries, you've probably seen both `COUNT(*)` and...



Mar 9



1.1K



32



Afan Khan in JavaScript in Plain English

Microsoft is ditching React

Here's why Microsoft considers React a mistake for Edge.



Jun 6



3.5K



75



See more recommendations