Farkalit Usman

vXebia@bbynuat.com

|  |  |  |
| --- | --- | --- |
| **Version** | **Name** | **Descriptions** |
| 1.0 | Farkalit Usman | Initial document for REST API Naming Convention |
|  |  |  |

Abstract

REST API naming convention which is free from ambiguity and easier to user the end user.

REST API Naming Convention

Resource Naming Convention

REST API Resource Naming Convention

Table of Contents

[Objective 2](#_Toc18572646)

[Best Practices of REST API Resource Naming 2](#_Toc18572647)

[Consistency Convention 2](#_Toc18572648)

[Further Reading 3](#_Toc18572649)

# Objective

Primary data representation in REST is called “**resource**”. A meaning full and consistent naming strategy is definitely a good design decision. If it is named poorly then its uses become confusing and difficult.

1. A **resource can be a singleton or a collection**. For example, “users” is a collection resource and “user” is a singleton resource.
   1. We can identify “users” collection resource using the URI “/users”.
   2. We can identify a single “user” resource using the URI “/users/{userId}”.
2. A **resource may contain sub-collection resources** also. For example, sub-collection resource “accounts” of a particular “customer”
   1. Can be identified using the URN “/customers/{customerId}/accounts”.

Similarly, a singleton resource “account” inside the sub-collection resource “accounts”

* 1. Can be identified as follows: “/customers/{customerId}/accounts/{accountId}”.

# Best Practices of REST API Resource Naming

1. Use nouns to represent resources: Instead of using an action (verb) we should use noun (things) because noun has its properties and verb does not. For example
   1. Users of the Systems
   2. Users Accounts
   3. Account payment
   4. Network Devices safety
   5. Network Device Status
2. Its resource URI should be as follows:
   1. <http://www.bankboubyan.com/user-management/users>
   2. <http://www.bankboubyan.com/user-management/users/accounts>
   3. <http://www.bankboubyan.com/user-management/users/accounts/payment>
   4. <http://www.bankboubyan.com/device-management/safety/ios>
   5. <http://www.bankboubyan.com/device-management/config/status>

# Consistency Convention

For consistency we use the following rule for minimum ambiguity and maximum readability.

1. Never use CRUD functions name in URIs. Instead use HTTP Verb (GET, POST, PUT and DELETE).

For Example: <http://www.boubyan.com/users/get-accounts> (Not good)

For Example: <http://www.boubyan.com/users/create-accounts> (Not good)

Instead we use HTTP Verb as below:

HTTP GET <http://www.boubyan.com/users/accounts> (Get All user’s accounts)

HTTP GET <http://www.boubyan.com/users/accounts>/{accountId} (Get one user’s accounts)

HTTP POST <http://www.boubyan.com/users/account> (Create user’s accounts)

HTTP PUT <http://www.boubyan.com/users/account>/{accountId} (Update user’s accounts)

HTTP DELETE <http://www.boubyan.com/users/account>/{accountId} (Delete user’s accounts)

1. Use query component to filter the Collection

For Example: HTTP GET <http://www.boubyan.com/users?city=NEWYORK>

For Example: HTTP GET <http://www.boubyan.com/users?city=NEWYORK&status=active>

1. Use forward slash (/) for hierarchical relationships.

For Example: [http://www.boubyan.com/users/accounts/{accountId}](http://www.boubyan.com/users/accounts/%7baccountId%7d)

1. Do not use trailing forward slash in URIs.

For Example: <http://www.boubyan.com/users/accounts/> (remove last slash)

1. Use hyphen (-) to improve the readability.

For example: <http://www.boubyan.com/users/accounts/request-approval>

1. Do not use underscore ( \_ )
2. Use lower case letters instead of capital
3. Do not use any extensions

For Example: (<http://www.boubyan.com/users/accounts/identity-card.pdf>)

# Further Reading

1. <https://restfulapi.net/resource-naming/>