**No 1**

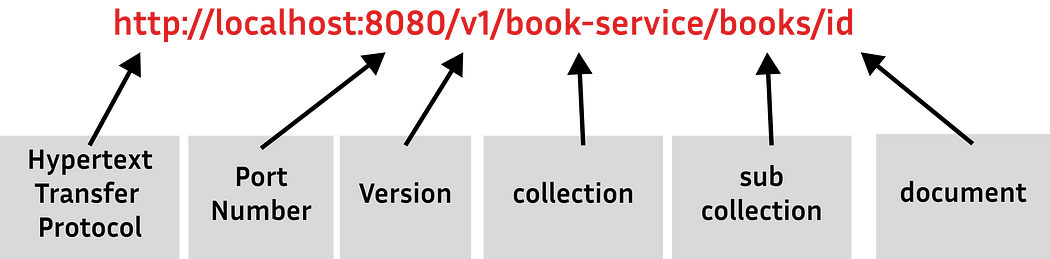
I basically researched about Client server Architecture this is basically where the server and client are separate entities. The client is responsible for the user interface and user interaction while the server handles the data processing and logic.

The clients and servers can evolve without changing the server and vice versa.

Real life example is the mobile app interacts with Restful API servers to fetch data or perform interactions.

**No3**

**Uniform Resource Identifiers.**

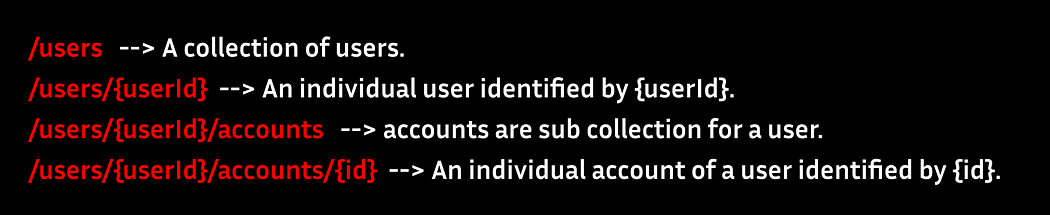


URIs play a significant part in identifying and routing resources. URIs are used to address API-exposed resources. When resources have proper names in URI, an API becomes natural and simple to use. If not implemented correctly, the same API can be difficult to operate and understanding.

The resource is the information that is to be manipulated or accessed using URIs.

Example: -  **Events, Products, Orders, Articles, Customers**

This is basically an example of the hierarchy of the way they are arranged

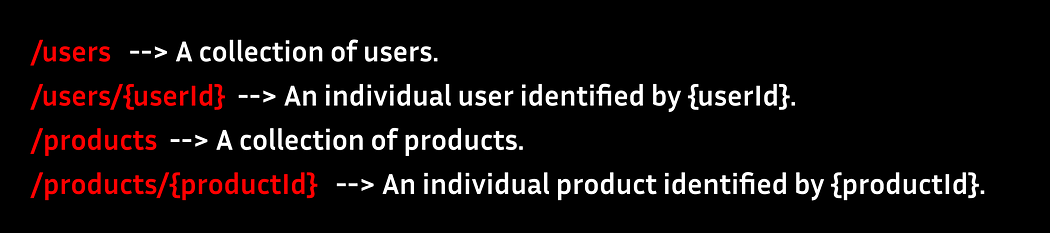


Key Principles when writing URIs

**1. Use nouns to represent resources**

RESTful URIs should refer to a resource that is a thing (noun) rather than an action (verb), because nouns have properties that verbs do not, and resources have attributes. *Examples: Teacher (****Teacher has a name, id, age, qualifications etc.****), Apple (****Apple has a color, taste, expiry date etc.****)*

*An example of using nouns*



# 

# **2. Consistency is the key**

Use consistent resource naming conventions and URI formatting to minimize uncertainty and maximize readability and maintainability. ***Use forward slash (/) to indicate hierarchical relationships.*** Used in the URI’s path to denote a hierarchical relationship between resources.

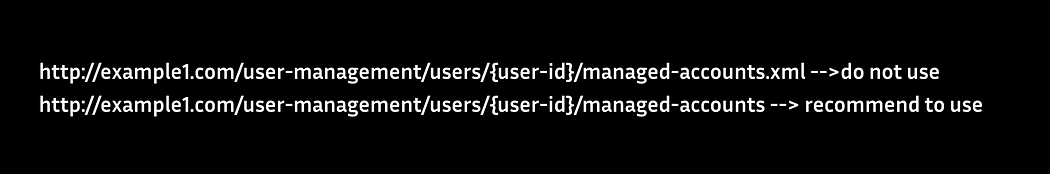
**Do not use trailing forward slash (/) in URIs like at the end don’t add the last slash**

***Use hyphens (-) to improve the readability of URIs.*** To make your URIs easy to understand, use the hyphen (-) character to improve the readability of names in long-path segments.

***Use lowercase letters in URIs.*** When convenient, lowercase letters should be consistently preferred in URI paths.

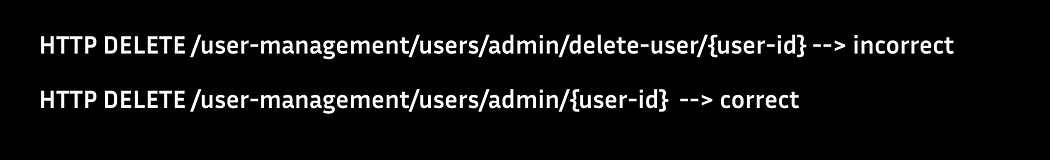
**3. Do not use file extensions**

File extensions look bad and do not add any advantage. And no reason to keep them.



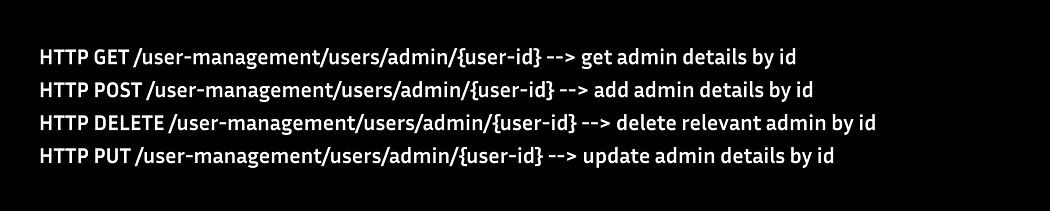
**Do not Use Verbs in the URI**

It is not appropriate to use verbs in REST URIs as REST uses nouns to represent resources, and HTTP methods (GET, POST, PUT, DELETE, etc.) to perform actions on those resources. In cases, where we need to perform some action that does not apply naturally to the definition of resources, we can create the custom URIs that can be considered nouns/resources and perform an action over them



**4. Never use CRUD function names in URIs**

We should not use a URI to demonstrate a CRUD function. URIs are simply used to identify resources and do not allow for any unique actions on them. We should use HTTP request methods to indicate which CRUD function is performed.



Reference:

https://ekanayakasalitha.medium.com/what-is-rest-891dadf5c1c9