REST Resources

The initial concept of the RESTful API is its resource and its functionality. It acts as an object of OOP (Object Oriented Programming) language or a database entity. As the resources are recognized, they're determined using a standard format so that the server can transmit the resource in the specific developer's format.

Table of Contents

What is The REST Resources?

A resource can be defined as a vital element to be referenced within a client-server system. REST architecture treats all of its content as a resource, which includes Html Pages, Images, Text Files, Videos, etc. Access to resources is provided by the REST server where REST client is used for accessing as well as modification of resources. All of its resources get identified via URI, which is abbreviated as Uniform Resource Identifier.

Here are some possible resources:

- Version 1.0.3 of any app release.
- The latest edition of any software.
- A weblog entry for August 28, 2019.
- A road map of Indore.
- Some information on salmon fish.
- Sales numbers for Q4-2018.
- A relationship between 2 acquaintances - Alice and Bob.
- The list of open vulnerabilities in the database.
- User profile information.

Representation of REST Resources

There are various forms of representing REST resources such as XML, text-based, JSON, etc. But, the two most popular forms of representing resources are using JSON and XML. REST doesn't pose any restriction for a specific format in representing resources in REST. It all depends on the choice and requirement of the project and developer. Hence, you can use any format for representing resources in REST. Here are two samples of a single example in that is implemented in RESTful web services:

The following example is shown in the XML format that shows the user's profile information:

```
<root>
   <Data>
      <UserGUID>4f1764fe-b3d3-ce7c-0f08-0ef09d834b7e</UserGUID>
      <FullName>Alex</FullName>
      <ProfilePic>http://localhost/9c0977f4-877a-8138-4fbb-a524edf20437.jpg</ProfilePic>
   </Data>
   <Message>Success</Message>
   <ResponseCode>200</ResponseCode>
</root>
```

The same example of REST resource is now shown using JSON format:

```
{
  "ResponseCode": 200,
  "Message": "Success",
  "Data": {
    "UserGUID": "4f1764fe-b3d3-ce7c-0f08-0ef09d834b7e",
    "FullName": "Alex",
    "ProfilePic": "http:\/\/localhost\/9c0977f4-877a-8138-4fbb-a524edf20437.jpg"
  }
}
```

Disadvantages with XML for Formatting

There are few disadvantages with XML for formatting the REST resources. These are:

- **Lacking in data-type**: XML elements do not express the type of data or the order to employ it. For this, one would have to utilize XMLSchema.
- **Lacking in lists**: There is no way of natively expressing the lists. This makes it confusing whether a certain element is a list or an object.

For the Development of Good and Standard Rest Resources

Some fundamental points need to be kept in mind when you are constructing a REST resource. These are:

- **Completeness in meaning**: Your format should represent the resource completely that you intend to create. The resource can be either simple or complex, or a nested one, which comprises a resource containing another resource.
- **Understandable structure**: You should frame the REST resource in such a way that it can be understandable by the server as well as the client so that the REST can utilize the format of your resource.
- **Linkability**: There may be some situations where your REST may have a resource that is linked to another resource. In such situations, you have to manage and structure it properly.

Next Chapter : REST Methods ›