REST Architecture

AGENDA

- REST Architecture
- The Main Components of REST
- The Different Methods of REST
- Conclusion

REST Architecture

❑ Basics of REST Architecture

REST is an acronym for REpresentational State Transfer and a software architectural style for distributed hypermedia systems. Every item of interest in REST is called a resource.

The key abstraction of information in REST is a resource. Any information that we can name can be a resource. For example, a REST resource can be a document or image, a temporal service, a collection of other resources, or a non-virtual object (e.g., a person).

REST Architecture

❑ Uniform Resource Identifier (URI)

REST uses resource identifiers to identify each resource involved in the interactions between the client and the server components.

A URI is a string which uniquely identifies a resource on the internet. URI can be of two types: Uniform Resource Locator (URL) or Uniform Resource Name (URN). The URL is used to display where a resource is located. For this reason, the URL is also utilised when surfing on the Internet to navigate to specific websites. In contrast, the URN is location-independent and permanently designates a resource. Thus, if URLs are primarily known in the form of web addresses, a URN can, for example, also appear as an ISBN to permanently identify a book.

REST Architecture

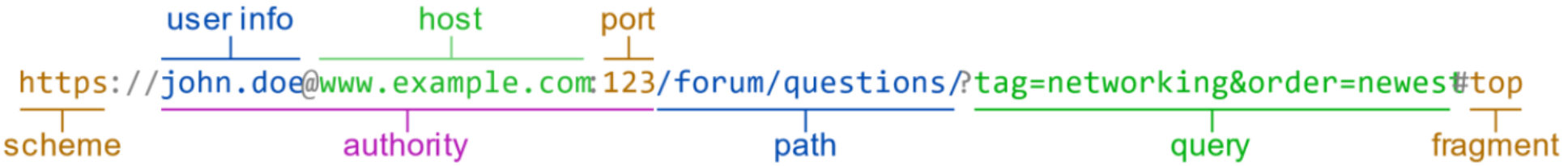
❑ Uniform Resource Identifier (URI)

A URI consists of up to five parts. However, only two of these are mandatory.

- **scheme: Gives information about the protocol being used.**
- **authority: Identifies the domains.**
- **path: Shows the exact path to the resource.**
- **query: Represents a request action.**
- **fragment: Refers to a partial aspect of a resource.**

Only scheme and path must appear in every identifier. In the URI syntax, all components are listed successively and separated by specific, predefined characters.

Structure of a URI (a URL)



REST Architecture

❑ Characteristics of REST Architecture

REST interfaces allow a web-based resource to be accessed from any other client application on a network. The resource could be an actual object on the server, or it could be a verb (a method) to be called on the server. Some of the most important characteristics of REST architecture are

- **Uniform interface**
- **Client-server architecture**
- **Statelessness**
- **Cacheable**
- **Layered System**
- **Code on Demand (Optional)**

The Main Components of REST

❑ Components of REST

The REST interfaces (messages) contain different components. Some of the main benefits of APIs include the following:

- **Request URI**
- **Request Headers**
- **Response Headers**
- **Response Codes**
- **Response Body**

The Main Components of REST

- Request URI

The Request URI is also know as Resource Path (request target). The path to the resource (object) to be acted upon. The ID of the resource must be provided in the path.

For example, the following resource path identifies a specific transaction (resource) in our database: https://apitest.cybersource.com/pts/v2/payments/

The main part of the resource path begins after the host (also know as Top Level Domain (TLD)), apitest.cybersource.com. /pts/v2/payments is the address (resource) on the CyberSource end that processes transaction detail requests. CyberSource returns a request ID that provides transaction details you can use for follow-up transactions, queries, or reference.

The Main Components of REST

❑ Request Headers

The header is a collection of fields and their associated values that provide information about the message to the receiver. Think of it as metadata about the message. Here are these fields :

- **Accept: specify desired media type of response**
- **Accept-Language: specify the desired language of response**
- **Date: date/time at which the message was originated**
- **Host: host and port number of a requested resource**
- **If-Match: conditional request**
- **Referrer: URI of previously visited resource**
- **User-Agent: identifier string for a web browser or user agent.**
- **Cookie: matched cookies for state maintenance**

The Main Components of REST

- Request Body

A POST or PUT request has a body. A GET request has no body. REST uses JSON (JavaScript Object Notation) as its content format. This body can contain detailed data, large documents which usually cannot be sent as part of a URL.

The Main Components of REST

❑ Response Headers

These are the commonly found response headers in the HTTP response:

- **Allow: list the REST methods supported by the request URI**
- **Content-Language: language of the response content**
- **Content-Type: media type of representation, where typical content types are text/html, application/json, etc.**
- **Content-Length: length in bytes of the representation**
- **Date: date/time at which the message was originated**
- **Expires: date/time after which the response is considered obsolete**
- **Last-Modified: date/time at which the representation was last changed.**
- **Set-Cookie: sets cookies for the domain.**

The Main Components of REST

❑ Response Codes

There are five categories of response codes. They are summarized below:

- **1XX: Informational**
- **2XX: Success**
  200 OK, 201 Created, 202 Accepted, 204 No Content
- **3XX: Redirection**
  300 Multiple Choices, 301 Moved Permanently, 302 Moved Temporarily, 304 Not Modified
- **4XX: Client Error**
  400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found
- **5XX: Server Error**
  500 Internal Server, Error 502 Not Implemented

The Main Components of REST

- Response Body

The response body is the content returned from the server in response to a REST method call. Typically, this response is in form of JSON. JSON stands for JavaScript Object Notation. This is a lightweight data format used for storing and transporting data. JSON is easy to understand and is self-explanatory.

An example of a JSON response is:


```
{
  "userId": 1,
  "id": 1,
  "title": "delectus aut autem",
  "completed": false
}
```

The Different Methods of REST

❑ Methods of REST

In the context of REST architecture, there are semantics analogous to the database operations (Create, Read, Update, Delete) for the key methods. This mapping provides a clear design principle for designing REST interfaces. The mapping can be roughly illustrated as below:

- **GET – read operation**
- **POST – create records**
- **PUT/PATCH – update records**
- **DELETE – delete records**
- **HEAD – return header**

The Main Components of REST

❑ GET Method

GET is the preferred method for accessing a resource on the internet. Hence it is equivalent to the Read operation in databases. Much like the SELECT clause in SQL, you request to return a set of data items. The format of a GET request varies in terms of the underlying URI. The URI format is dependent upon the application provider. The important characteristics of the GET request are outlined below.

A GET request:

- **is a read-only request**
- **does not modify any resource**
- **is idempotent, i.e., when you request the same resource twice, it should give the same response**
- **puts all the content in the URI**
- **typically, does not put anything in the body**
- **URI can take at best 64 KB of data**
- **is not encouraged when there is a need to send secured data.**

The Main Components of REST

❑ POST Method

POST is usually used when there is a need to send content of more than 64KB, even for a database read operation. The important characteristics of the POST request are listed below.

A POST request:

- **is used for sending secured data**
- **uses simple URIs but sends most content in the request body part**
- **as it typically corresponds to the database operation of creating a record, it is recommended when there is a need to create a new resource on the server**
- **is not idempotent**

The Main Components of REST

❑ PUT Method

As it typically corresponds to the database operation of updating a record, the PUT method is recommended when there is a need to update an existing resource on the server. The important characteristics of the PUT request are listed below.

A PUT request:

- **is idempotent**
- **may use content in URI or in BODY**
- **updates the entire record, so all the field values (updated) should be provided**

The Main Components of REST

- PATCH Method

PATCH is the same as PUT, except that you can do a partial update of the resource on the server, unlike PUT, where all the fields are updated. As a result, you can update one or two fields as well.

The Main Components of REST

- DELETE Method

The DELETE method is used to indicate to the server to delete a stated resource.

The Main Components of REST

- HEAD Method

The HEAD method is a special case of the GET request, where only the response headers are returned with no body. It is usually used to test response codes, or to test if the server is up and running.

Conclusion

In this presentation, we have covered the REST Architecture from scratch. Next, We will talk about SOA (Service-Oriented Architecture). The links to our resources are attached below

https://www.api-united.com/_files/ugd/b7b6d6_4003192bb0d942f6a5d45dbb1e7fafb5.pdf

https://www.baeldung.com/cs/rest-architecture

https://restfulapi.net/

https://developer.cybersource.com/api/soap-developer-guides/dita-flex/SAFlexibleToken/RESTComponents.html

MERCI

Pour votre attention