**REST API**

Menu

->Introduction

->REST vs HTTP

->Rest Api Methods

->Status Codes

**Introduction :**

REST Architecture Style application, a web application need to design the business logic(resides in a server e.g. Tomcat,Apache HTTP,JBoss) with all set of object entities(e.g. Customer is an entity) and possible operations(e.g.‘Retrieve Customer Information based on a customer id’) on them. Those possible operations with these entities should be designed with four main operations or methods namely- Create,Retrieve,Update,Delete(CRUD). These entities (e.g. Customer is an entity) called as resources and these are presented or represented in a form e.g. JSON or XML or something else. We have Client(Browsers) who calls Create,Retrieve,Update,Delete(CRUD) methods to perform the appropriate function on such resource resides in the Server.

**REST vs HTTP:**

Basically the REST is a architectural web design where we use the few actions of rest on http .In simple terms REST is a specific way of approaching the design of big systems (like the web).It's a set of 'rules' (or 'constraints').HTTP is a protocol that tries to obey those rules.

**REST API METHODS**

As discussed before rest uses the characteristics of http here we use the methods used by HTTP

* GET
* POST
* PUT
* DELETE
* PATCH

|  |  |  |  |
| --- | --- | --- | --- |
| METHODS | CRUD | **ENTIRE COLLECTION (E.G. /USERS)** | SPECIFIC ITEM (E.G. /USERS/123) |
| GET | Read | 200 (OK), list of users. Use pagination, sorting and filtering to navigate big lists. | 200 (OK), list of users. Use pagination, sorting and filtering to navigate big lists. |
| POST | Create | 201 (Created), ‘Location’ header with link to /users/{id} containing new ID. | Avoid using POST on single resource |
| PUT | Update/Replace | 404 (Not Found), unless you want to update every resource in the entire collection of resource. 200 (OK) or 204 (No Content) | Use 404 (Not Found), if ID not found or invalid. |
| PATCH | Partial Update/Modify | 404 (Not Found), unless you want to modify the collection itself. | 200 (OK) or 204 (No Content). Use 404 (Not Found), if ID not found or invalid. |
| DELETE | Delete | Delete 404 (Not Found), unless you want to delete the whole collection — use with caution. | 200 (OK). 404 (Not Found), if ID not found or invalid |

**Idempotent Methods**

The term idempotent is used more comprehensively to describe an operation that will produce the same results if executed once or multiple times. This is a very useful property in many situations, as it means that an operation can be repeated or retried as often as necessary without causing unintended effects. The methods GET, HEAD, PUT and DELETE are declared idempotent methods. Other methods OPTIONS and TRACE SHOULD NOT have side effects so both are also inherently idempotent.

**Status Codes**

|  |  |
| --- | --- |
| **CATEGORY** | **DESCRIPTION** |
| **1xx: Informational** | Communicates transfer protocol-level information. |
| **2xx: Success** | Indicates that the client’s request was accepted successfully. |
| **3xx: Redirection** | Indicates that the client must take some additional action in order to complete their request. |
| **4xx: Client Error** | This category of error status codes points the finger at clients. |
| **5xx: Server Error** | The server takes responsibility for these error status codes. |

**List of Status codes and its description**

100 =Continue

101 =Switching Protocols

200 =OK

201 =Created

202 =Accepted

203 =Non-Authoritative Information

204 =No Content

205 =Reset Content

206 =Partial Content

300 =Multiple Choices

301 =Moved Permanently

302 =Found

303 =See Other

304 =Not Modified

305 =Use Proxy

306 =(Unused)

307 =Temporary Redirect

400 =Bad Request

401 =Unauthorized

402 =Payment Required

403 =Forbidden

404 =Not Found

405 =Method Not Allowed

406 =Not Acceptable

407 =Proxy Authentication Required

408 =Request Timeout

409 =Conflict

410 =Gone

411 =Length Required

412 =Precondition Failed

413 =Request Entity Too Large

414 =Request-URI Too Long

415 =Unsupported Media Type

416 =Requested Range Not Satisfiable

417 =Expectation Failed

500 =Internal Server Error

501 =Not Implemented

502 =Bad Gateway

503 =Service Unavailable

504 =Gateway Timeout

505 =HTTP Version Not Supported

**Conclusion :**

REST is a type of data transfer that is built upon the architecture of the HTTP protocol. It allows you to easily send and retrieve data between two different services using XML or JSON.

When structuring your web applications, it’s usually good practice to build them using RESTful architecture. This means that collections and resources are easily recognised and can form the basis of building a RESTful API.