SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with ‘A’ Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Information Technology

Course Name – 19IT503 Internet of Things

III Year / V Semester

Unit 3 – EVOLVING IoT STANDARDS & PROTOCOLS

Topic 3- Representational State Transfer (REST)

Representational State Transfer (REST)

REST

- REST was first described in 2000 by Roy Fielding in his University of California dissertation which analyzed a set of web focused software architecture principles for distributed computing.
- It defines a set of architectural principles by which one can design WS
- It focus on a system's resources, including how resource states are addressed and transferred over HTTP by a plethora of clients written in different languages.
- REST is an architectural style of large-scale networked software that takes advantage of the technologies and protocols of the World Wide Web;
- It describes how distributed data objects, or resources, can be defined and addressed, stressing the easy exchange of information and scalability

Representational State Transfer (REST)

A REST-based WS follows four basic design principles:

- Use HTTP methods explicitly.
- Be stateless.
- Expose directory structure-like URIs.
- Transfer XML, JavaScript Object Notation (JSON), or both.

- A web API that obeys the REST constraints is informally described as RESTful

- RESTful web APIs are typically loosely based on HTTP methods to access resources via URL-encoded parameters and the use of JSON or XML to transmit data

The REST architectural constraints are
Client-Server, Stateless, Cache-able, Layered System, Uniform Interface, Code on Command

Representational State Transfer (REST)

Making Requests

REST requires that a client make a request to the server in order to retrieve or modify data on the server.

A request generally consists of:

- an HTTP verb, which defines what kind of operation to perform
- a header, which allows the client to pass along information about the request
- a path to a resource
- an optional message body containing data

HTTP Verbs

There are 4 basic HTTP verbs we use in requests to interact with resources in a REST system:

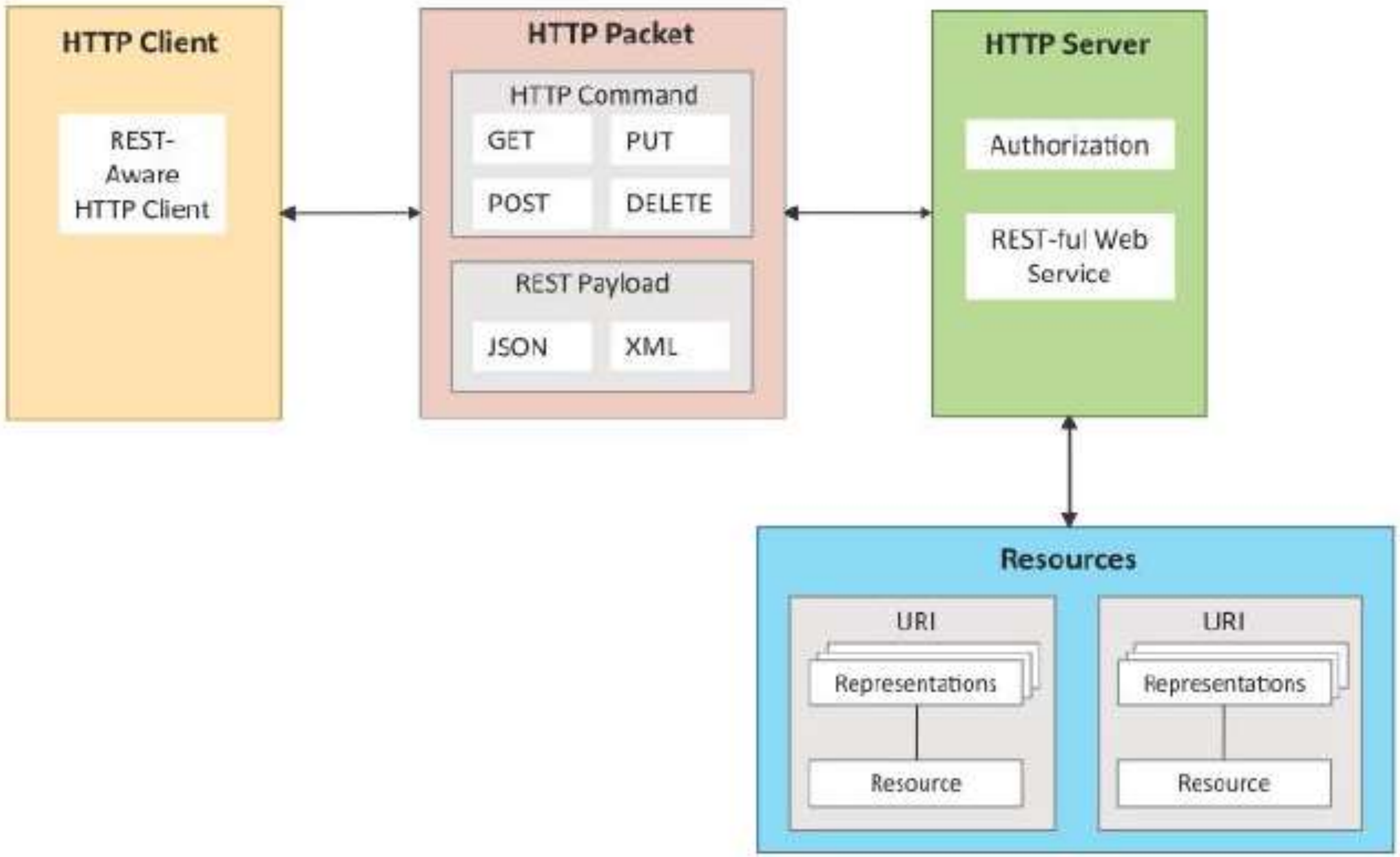
GET — retrieve a specific resource (by id) or a collection of resources

POST — create a new resource

PUT — update a specific resource (by id)

DELETE — remove a specific resource by id

Representational State Transfer (REST)



Representational State Transfer (REST)

- In the header of the request, the client sends the type of content that it is able to receive from the server.

- This is called the Accept field, and it ensures that the server does not send data that cannot be understood or processed by the client.

- MIME Types, used to specify the content types in the Accept field, consist of a type and a subtype. They are separated by a slash (/).

- For example, a text file containing HTML would be specified with the type text/html.
- If this text file contained CSS instead, it would be specified as text/css.
- A generic text file would be denoted as text/plain. This default value, text/plain

Other types and commonly used subtypes:

- image — image/png, image/jpeg, image/gif
- audio — audio/wav, audio/mpeg
- video — video/mp4, video/ogg
- application — application/json, application/pdf, application/xml, application/octet-stream

Representational State Transfer (REST)

- For example, a client accessing a resource with id 23 in an articles resource on a server might send a GET request like this:

- GET /articles/23 Accept: text/html, application/xhtml

- The Accept header field in this case is saying that the client will accept the content in text/html or application/xhtml.

Paths

- Requests must contain a path to a resource that the operation should be performed on.
- In RESTful APIs, paths should be designed to help the client know what is going on.

Example

- GET flipkart.com/customers/223/MyOrders/12
- DELETE flipkart.com/customers/223/MyOrders/12

Representational State Transfer (REST)

Content Type
In cases where the server is sending a data payload to the client, the server must include a content-type in the header of the response.

This content-type header field alerts the client to the type of data it is sending in the response body.

For example, when a client is accessing a resource with id 23 in a topic resource with this GET Request:
GET snscourseware/IT/IoT/Topic/23 HTTP/1.1
Accept: text/html, application/xhtml

The server might send back the content with the response header:
HTTP/1.1 200 (OK)
Content-Type: text/html

Representational State Transfer (REST)

Response Codes

Responses from the server contain status codes to alert the client to information about the success of the operation.

Status code Meaning

200 (OK) This is the standard response for successful HTTP requests.

201 (CREATED) This is the standard response for an HTTP request that resulted in an item being successfully created.

204 (NO CONTENT) This is the standard response for successful HTTP requests, where nothing is being returned in the response body.

400 (BAD REQUEST) The request cannot be processed because of bad request syntax, excessive size, or another client error.

403 (FORBIDDEN) The client does not have permission to access this resource.

404 (NOT FOUND) The resource could not be found at this time. It is possible it was deleted, or does not exist yet.

500 (INTERNAL SERVER ERROR) The generic answer for an unexpected failure if there is no more specific information available.

THANK YOU