

8/09/2020

Name : Mahima Bisht

Email Id = mahima.bisht@capgemini.com

EmpId = 46010069

△ HTTP Methods

- GET
- POST
- DELETE
- OPTIONS

△ HTTP Status Codes

- 200 - OK - request successful
- 201 - created
- 204 - No Content
- 301 - Moved Permanently
- 304 - Not modified
- 401 - Unauthorized / Unauthenticated
- 404 - Not found
- 500 - Internal Server error
- 503 - Service Unavailable

△ Cache control header - it tells/specifies time that a resource will be fresh. So that cache data can be reuse

- helps in minimizing traffic.

△ Options method - used to receive a response that tells which about the different methods or what it supports and what not.

△ Delete Method - used to delete a repository

08/09/2020

Name: Mahima Bisht
Emp Email: mahima.bisht@apgemini.com
Emp Id: 46010069

- In response, we will a lot of things including status codes and statusline.

```
int code = response.getStatusLine().  
getStatusCode();
```

- default number of connections in Apache is 2 connection.

- response.close(); will close the connection.

- HttpClient and HttpResponse are not class these are interfaces.

- MIME TYPE & Media Type are found in a response.

- Header is called - Content type.
→ it tells media type + encoding.

△ Http Headers - meta data.
- few headers are:
→ Status
→ Server
→ Cache-Control
→ Content type
→ Date
→ ETag.

- We can get value of headers by using:
getHeader(response, "server");
- it will return a "string"

08/09/2020

Name = Mahima Bisht
EmailId = mahima.bisht@
capgemini.com
EmpId = 46010069

API test automation

△ POSTMAN

- Started as a chrome extension
- later, became desktop application.
- web API testing tool.
- tests are written in JS.
- can modify methods while inspecting API.
- tests runs AFTER sending request & receiving response.

△ SOAPUI

- desktop application
- acts as client talking to web API.
- sends http request & evaluate http responses.

△ Basic test using get method

- ~~not~~ need Apache HttpClient dependency.
- Create httpget obj as:
`HttpGet obj = new HttpGet("URL");`
- Create client as:
`HttpClient client = HttpClientBuilder.create().build();`
- execute the request as:
`client.execute(obj);`
- and assign it to response obj of Http-
response Class.