1 Web Application Architecture, HTTP, REST API, and Postman Training Module

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. This is the foundation for data communication for the World Wide Web. It is very important that we understand the HTTP clearly before we start looking at what is a REST API. The term REST stands for Representational State Transfer. It is an architectural style that defines a set of rules to create Web Services and Postman is a software development tool. It enables people to test calls to APIs.

1.1 Learning Objectives

- Web Application Architecture:
    - Client Server Architecture
    - MVC Architecture
    - Single Tier, 2-Tier, 3-Tier, N-Tier Applications
    - Peer-Peer Architecture (Optional)
- HTTP:
    - Request
    - Response
    - Methods
    - Status Codes
    - Headers
- REST:
    - Constraints/Properties
    - Resource
    - Resource and URI Design Guidelines
- Postman:
    - Send Requests
    - Authorize Requests
    - Collections
    - Variables
    - Environments
    - Tests (Optional)

1.2 Learning Resources

1.2.1 Web Application Architecture

- Web Architecture overall Web/PDF Tutorial (Mandatory):
- 3 Tier Architecture In-depth Web/PDF Tutorial (Mandatory):
- Client-Server-Model Web/PDF Tutorial (Optional Read)

1.2.2 HTTP

- Web/PDF Tutorial (Mandatory)
- Quick Overview Video Tutorial (Optional)

1.2.3 Rest

- Web/PDF Tutorial 1
- Web/PDF Tutorial 2
- Video Tutorial (Optional):

1.2.4 Postman

- Video Tutorial
- Web/PDF Tutorial

1.3 REFERENCES

- https://docs.github.com/en/free-pro-team@latest/github/authenticating-to-github/creating-a-personal-access-token
- https://developer.github.com/v3/repos/

1.4 ASSIGNMENT

- Generate User access token using your GitHub Account
- Create a postman collection and name it as Git API Testing
- Create a postman environment and name it as Git API Testing
- Create variables user token and repository name in the above environment
- Create a private user repository using the variables from above created environment and GitHub API.
- Get the repository details
- Delete the repository using the variables from above created environment and GitHub API