POSTMAN

Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster

DEFINITION OF API

Application Programming Interface: are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols.

Types of API protocols

- REST API. REST (Representational State Transfer) is a web services API.
- SOAP API. SOAP (simple object access protocol) is a well-established protocol, similar to REST in that it's a type of Web API. ...
- RPC API.
- Event-driven APIs, aka asynchronous APIs.

POSTMAN API PLATFORMS:

1. APM
2. CI/CD
3. CLOUD
4. SOURCE CONTROL

APM: Application performance monitoring is the collection of tools and processes designed to help IT professionals ensure that enterprise applications meet the performance, reliability and valuable user experience (UX) required by employees, partners and customers.

CI/CD: stand for continuous integration and continuous delivery/continuous deployment. In very simple terms, CI is a modern software development practice in which incremental code changes are made frequently and reliably.

CLOUD: is the delivery of different services through the Internet, including data storage, servers, databases, networking, and software. Cloud storage has grown increasingly popular among individuals who need larger storage space and for businesses seeking an efficient off-site data back-up solution.

USES OF POSTMAN

Postman can store and manage API specifications, documentation, workflow recipes, test cases and results, metrics, and everything else related to API

Using Postman, you can send a request to an endpoint, retrieve data from a data source, or test an API's functionality. You don't need to enter commands in a terminal or write any code.

Some of the common uses of Postman are:

- Sending various types of HTTP requests (GET, POST, PUT, PATCH, etc.) to an API endpoint and inspecting the response data and headers.
- Saving and organizing requests into collections and folders for easy access and reuse.
- Creating and running automated tests for APIs using scripts and variables.
- Generating and publishing documentation for APIs based on the requests and responses.
- Creating and using mock servers to simulate API responses without hitting the actual backend.
- Sharing and collaborating on API projects with team members or other developers.