LU11b - Postman Client/Frontend Simulation
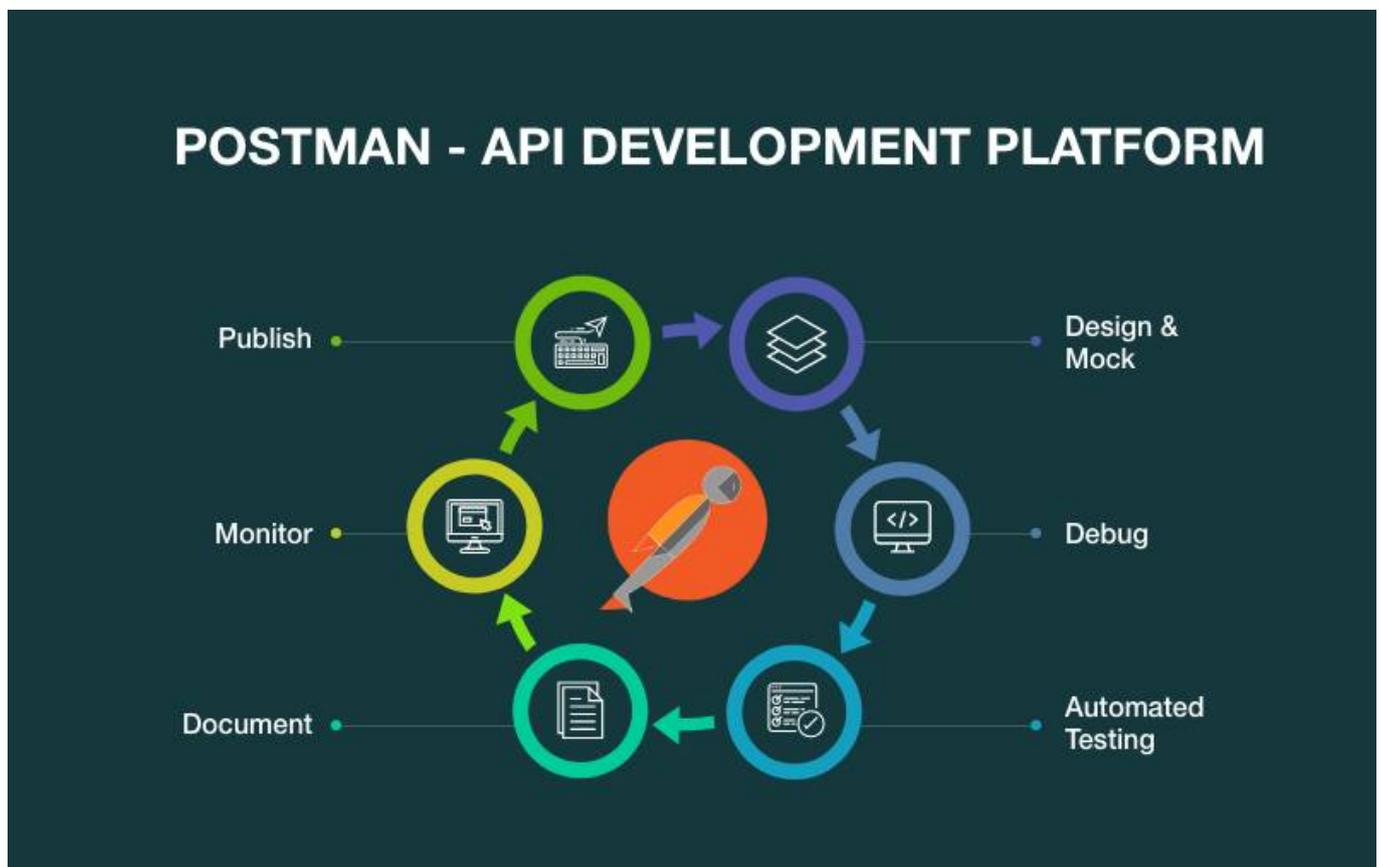
Learning Objectives

1. Know what POSTMAN is an describe the benefits of it
2. Describe how to use POSTMAN withing the module 290

Sources

- What is an API

Introduction

Postman is a popular application used for API development, testing, and documentation. It provides a user-friendly interface for creating and sending HTTP requests to APIs, allowing developers to interact with and test endpoints without needing a front-end application. With Postman, users can perform various HTTP methods—such as GET, POST, PUT, DELETE—and see responses directly in the interface, complete with status codes, headers, and body content. This functionality makes it invaluable for debugging, testing, and verifying API responses.

Additionally, Postman supports environment variables, automated testing with JavaScript, and the ability to save and organize requests in collections. These collections can be shared among team

members, which is particularly useful for collaborative API development and ensuring consistent testing across different development environments. Overall, Postman simplifies API interaction and enhances productivity in backend and frontend development.

Why using POSTMAN

In the context of developing and testing Node.js servers, Postman is an invaluable tool. Postman provides a graphical user interface (GUI) for developers to easily create, send, and analyze HTTP requests and responses, which is especially useful for testing APIs (Application Programming Interfaces) built with Node.js. Here's how Postman fits into the workflow:

1. **Request Creation and Testing:** With Postman, you can create HTTP requests (such as GET, POST, PUT, DELETE, etc.) to interact with your Node server. For example, if you have a REST API endpoint on your Node server, you can use Postman to test how it responds to different types of requests without needing a frontend interface. This speeds up backend testing by making it easy to customize and send specific data, headers, and authentication tokens to endpoints.
2. **Environment and Variable Management:** Postman allows developers to set up environments (like „development" or „production") and define variables to make requests repeatable and adaptable. For instance, if your Node.js server's URL changes across environments, you can set up environment variables in Postman to switch between them easily.
3. **Automated Testing and Collection Runners:** In addition to manual testing, Postman enables you to automate tests by writing JavaScript-based tests within the tool. These scripts can check for expected responses, validate data structures, or manage conditions based on response codes. Postman collections allow you to group and sequence API calls, making it possible to run a series of requests in a particular order—ideal for testing workflows or processes on a Node.js server.
4. **Debugging and Documentation:** Postman's interface provides detailed views of each request and response, including headers, status codes, and data. This makes it easy to troubleshoot errors in your Node server and verify that your responses match your API design. Plus, Postman offers tools for generating API documentation based on the requests you create, which you can share with teammates or external users to understand and interact with your API.

By streamlining API testing, Postman complements a Node.js development environment, allowing developers to quickly iterate, debug, and verify their server-side functionality.

Tutorials

Videotutorial Deutsch	**5 Minuten - API-Entwicklung mit Postman**
Videotutorial English	**Postman API Testing Tutorial Postman Tutorial For Beginners 2022**

Vocabulary

English	Deutsch
…	…

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/modul/m290/learningunits/lu09/theorie/02

Last update: **2024/11/05 13:14**