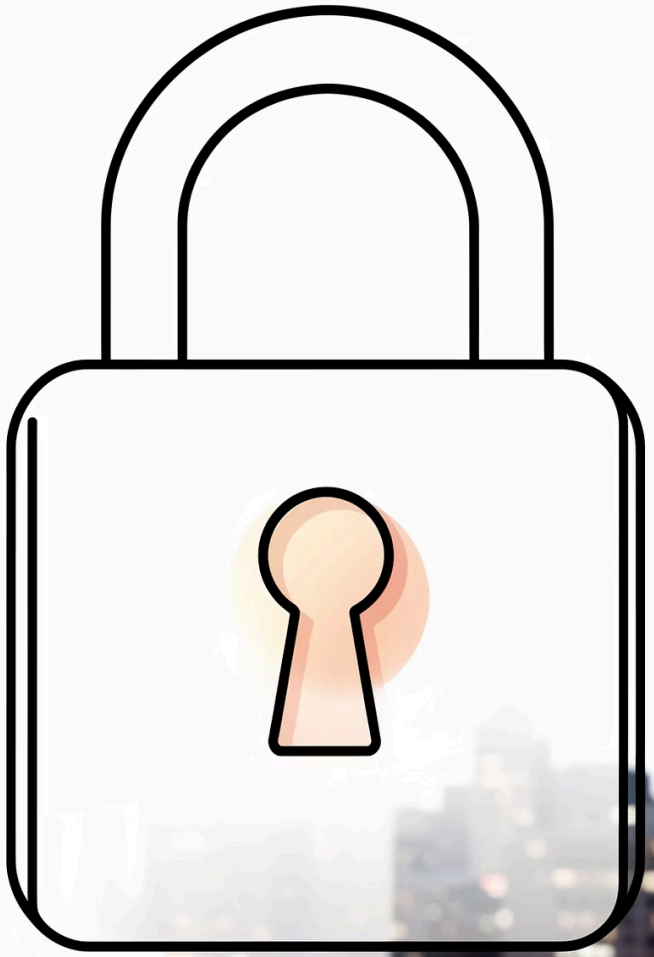
QA Workshop - Day 10

For BIT 5th semester students (Professional Class)

Instructor: **Neha Rouniyar**

API Basics: How to Keep Your Apps Safe

Learn how your apps can connect well and safely, making them easier to build and test.



CORE CONCEPT

Authentication: Proving Who You Are

Authentication is the first step when your app talks to another app (API). It's like asking, **"Who are you?"** The API checks your identity before letting you do anything.

- The API always checks if you're a real, signed-in user.
- If it doesn't know you, it says 'no.' This keeps data safe.

Username & Password

Your usual username and password.

Email & One-Time Code

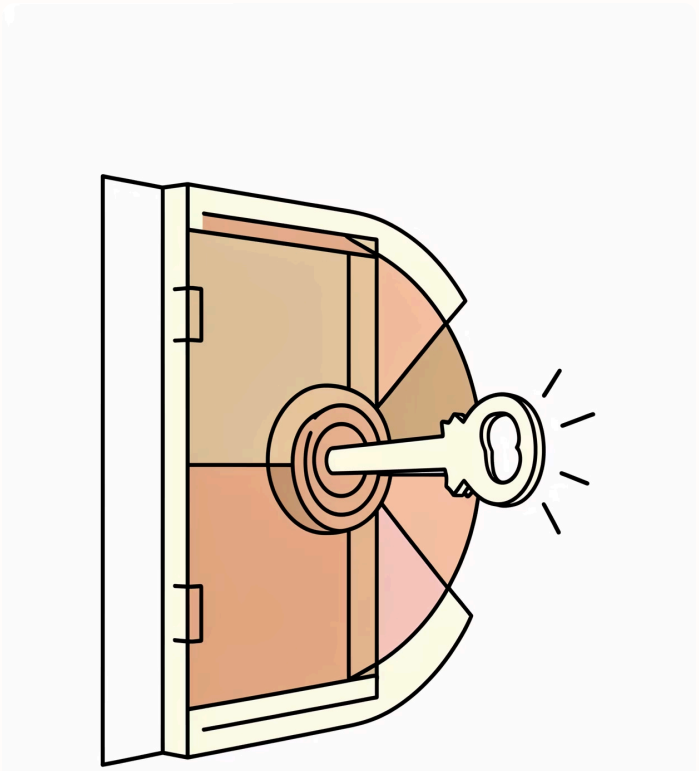
A temporary code sent to your email or phone for safety.

Access Key

A special key you get after you sign in.

NEXT STEP

Authorization: What You Can Do



First, the system knows who you are (authentication). Then, it asks, **"What can you do?"** This is authorization. It decides what you can see or do on the app or website.

- It only lets you do things based on your job or permission.
- It's like a bouncer. It stops people from doing wrong things or seeing private info.

Remember: You must first prove who you are (authenticate). Only then does the system decide what you can do (authorize).

Admin User

Can do anything: make, read, change, or delete users.

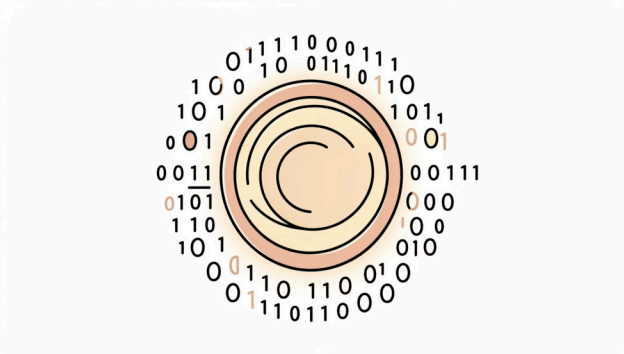
Normal User

Can do less. Can only look at info, not change or delete it.

MECHANISM

Tokens: Your Digital Pass

Imagine a token as a digital pass. When you log in, the server gives it to you. This pass shows who you are and what you can do. You won't need to type your password every time.



- After you log in, the server creates a special pass for you. Your app (like your web browser or phone app) gets this pass.
- Every time your app talks to the server, it sends your pass with its request.
- The server checks this pass. It makes sure you're still logged in and can do what you ask.

> Example: `Authorization: Bearer abc123xyz`

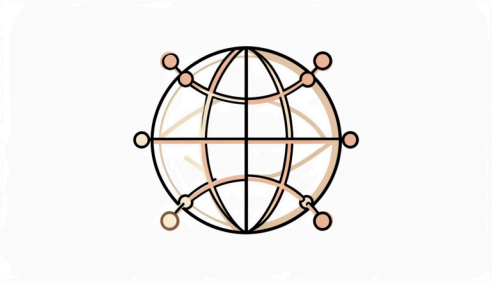
This pass makes things easier and safer. Your app talks to the server faster. The server doesn't need to check your password every time.

FOUNDATION

Base URL: Your API's Main Address

Think of an API like a building. The Base URL is its main street address.

To find a specific part of the API (an "endpoint"), you always start from this main address. It helps you find things easily.



Why Use a Base URL?

- **Less Typing:** You don't need to write the full address each time you access different parts of the API.
- **Simple Changes:** If the API address ever changes, you only update the Base URL, not every link.
- **Easy Switching:** It helps you switch easily between test, development, and live versions of the API.

> **Example Base URL:** `https://gorest.co.in/public/v2`
>
> **Then, specific API parts (endpoints) could be:** `/users`, `/posts`, `/comments`

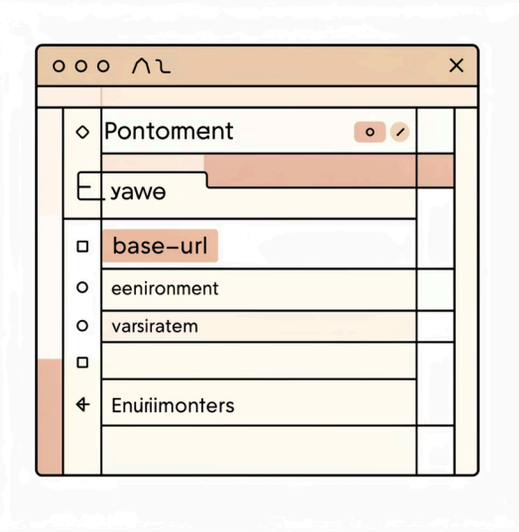
POSTMAN WORKFLOW

Shortcuts in Postman: Variables

Postman has "variables." Think of them as special boxes to store important information. They make testing your API simpler and smarter.

Instead of typing the same address or secret code many times, use a variable. It changes automatically!

Why Use Them?



- **Less Typing:** Your requests stay clean. You make fewer mistakes.
- **Change Fast:** Easily switch between test, development, and live systems. No need to change every request manually.
- **Work Faster:** Speeds up what you do. Use the same settings for everything.

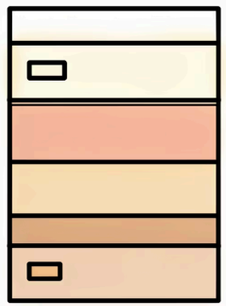
> **For example:** You can write `{{base_url}}/users`.
>
> Here, `base_url` is a variable set to `https://gorest.co.in/public/v2`. This means you don't type the long address each time!

REQUEST DETAILS

Headers: What Are They?

Headers are small notes sent with your API calls. They tell the server extra details about your request.

Imagine them as little sticky notes attached to your request, telling important things.



What Headers Tell Us:

- **Who You Are:** Your digital ID, like a login code.
- **What You Send:** The type of data you're sending, like text or JSON.
- **What You Want Back:** The type of data you expect in return.

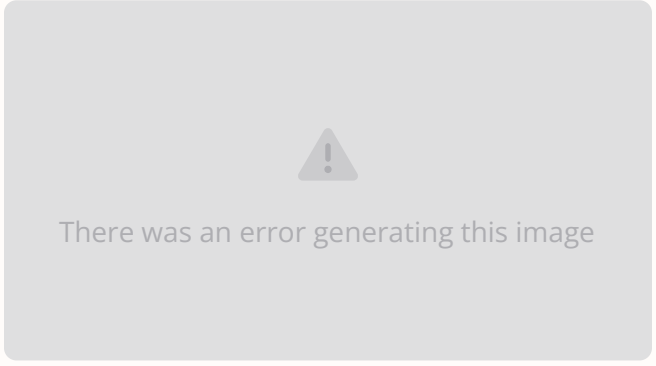
> ### Some Common Headers
>
> - `Content-Type: application/json`
> - `Authorization: Bearer your_token_here`
> - `Accept: application/json`

TARGETING RESOURCES

Path Parameters: How to Find One Item

Path parameters are bits of a web address (URL). They help you find just one thing from a list of many.

Think of them as codes right in the web address. They tell the computer exactly what you need.



When to Use Path Parameters:

- **For one item:** Like a person's page, a product's info, or one order.
- **For tidy data:** Great when your info is sorted, like files in folders.
- **For unique IDs:** When each item has its own special number or name.

> **Example:** `GET /users/1`
>
> Here, `1` is the path parameter. It asks for user number 1.

REFINING DATA

Query Parameters: Find & Filter Data

Query parameters are extra details you add to a web address (URL). They come after a question mark (?). They tell a website exactly what information you want.

These details help you get specific data without changing the main web address. It's a simple way to ask for what you need.

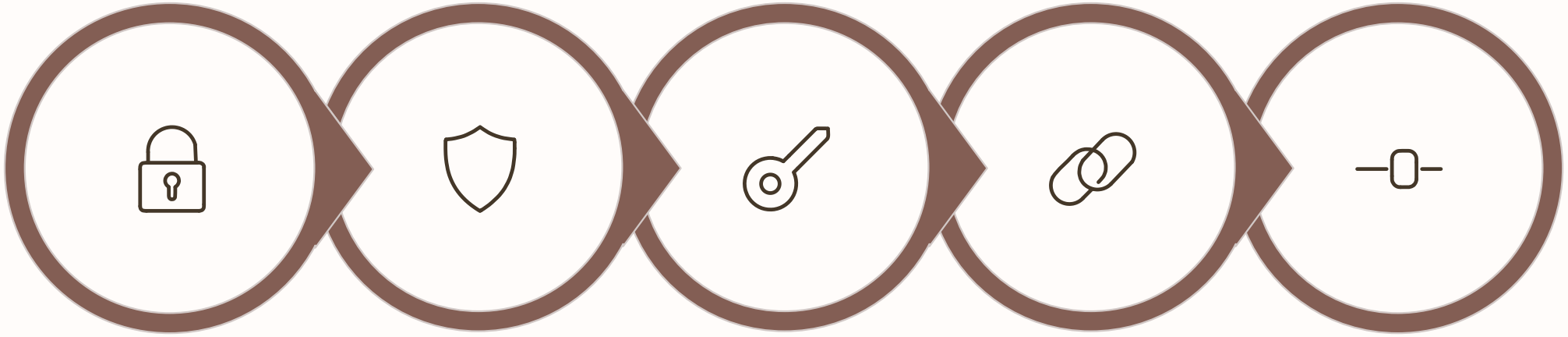
When to use Query Parameters:

- **Get data in parts:** Ask for data page by page if there's too much. (Example: `page=1&size=10` means "page 1, 10 items").
- **Filter results:** Only see items that meet certain rules. (Example: `status=active` means "show active items only").
- **Search for something:** Find items with a specific word or phrase. (Example: `search=keyword` means "find items with 'keyword'").
- **Sort your results:** Tell the website how to order the data. (Example: `sort=name&order=asc` means "sort by name, A to Z").

> **Example:** `GET /users?page=1&status=active`
>
> This asks for active users, showing only the first page of results.

SUMMARY

API Basics: A Quick Review



Who you are

What you can do

Your pass

Website address

Changeable parts

It's important to know these ideas to use and test APIs. Each part helps send information safely and correctly.

Now you can explore and test APIs!

Questions & Discussions??