



# RUMAD

Rutgers Mobile App Development

# Bonus Workshop 2: Postman/OpenAPI

**For this bonus workshop, we'll be covering:**

- OpenAPI Standard
- Using Postman for testing your APIs

The background is a solid dark red color. On the left side, there is a faint, stylized illustration of a smartphone. The phone's screen displays a white mask icon with two triangular eye cutouts. Below the screen, there is a white lightbulb icon. Several small white dots are scattered around the lightbulb. The word "Introduction" is written in a bold, white, sans-serif font, centered horizontally and partially overlapping the smartphone illustration.

# Introduction

# What are APIs?

## Application Programming Interface (API)

- Set of rules and definitions that allows for different pieces of software to communicate with each other
  - Defines how requests are made and what responses are expected (recall Weeks 2 & 3)
- Common in web applications (RESTful APIs, HTTP requests/responses, etc.)

# Importance of API Documentation and Testing

- **Documentation**
  - Provides a way of understanding how to use an API
- **Testing**
  - Ensures reliability and correctness of a program
- **OpenAPI** and **Postman** address these importances respectively



# **OpenAPI** **(Previously Swagger)**

# OpenAPI Documentation

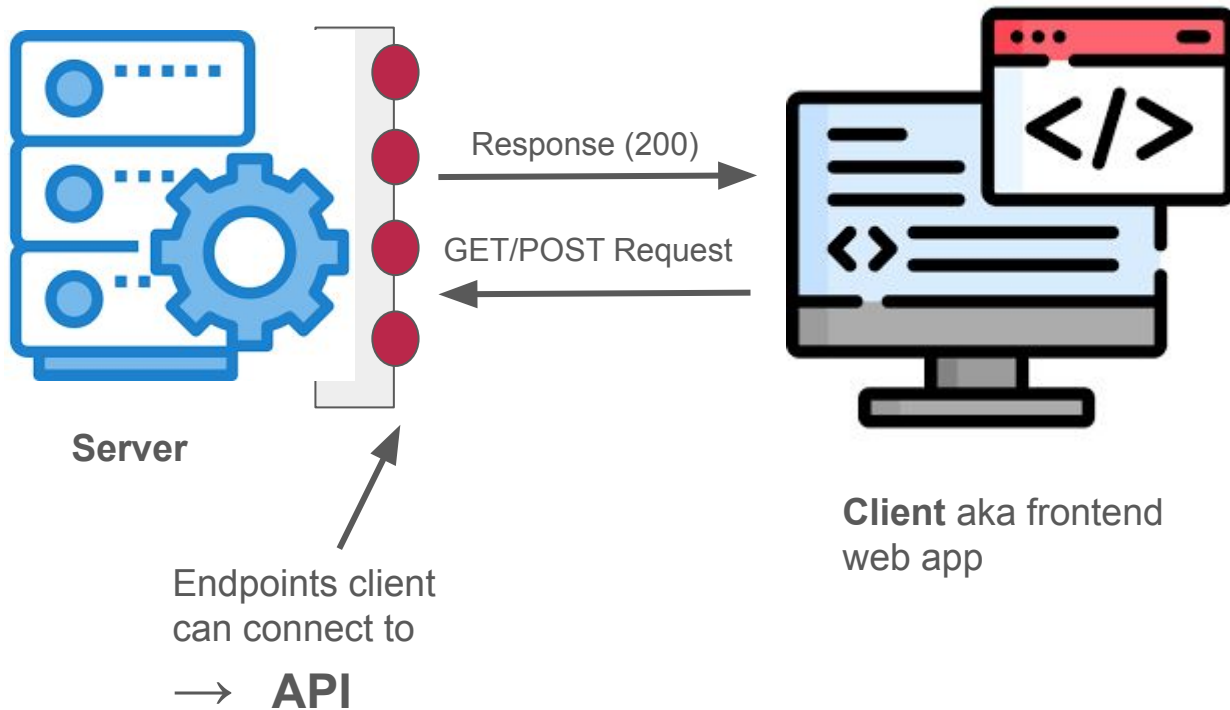
- Standardized format for describing RESTful APIs
  - REST = (Re)presentational (S)tate (T)ransfer
- Defines endpoints, methods, parameters, and responses
- Makes APIs machine readable and human understandable

Uses HTTP methods:  
GET/POST that we  
discussed!



*Citation: OpenAPI Specification - Wikipedia*

# OpenAPI & REST API



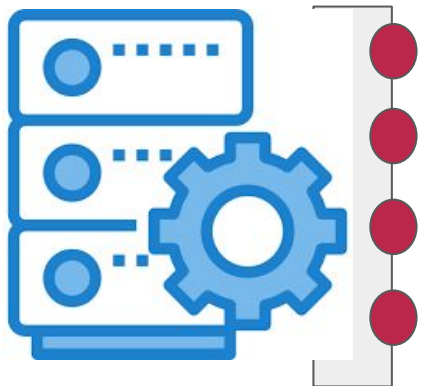
New Developer



Doesn't know what's going on in your API



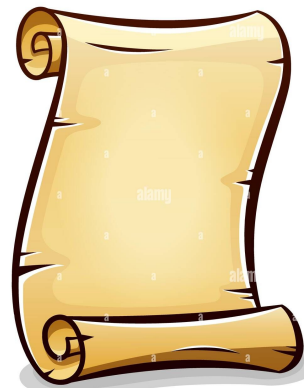
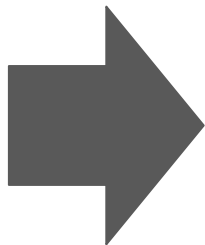
# OpenAPI & REST API



Server

Endpoints client  
can connect to

→ **API**



**New Developer**



Instead of digging through  
your code, new dev can  
just read your **OpenAPI**  
**definition**

# OpenAPI Documentation Cont.

- Written in YAML or JSON
- **Format**
  - Starts with version # and general info.
  - “paths” section lists each endpoint
  - HTTP methods and info. defined under each path

```
1  {
2    "openapi": "3.0.0",
3    "info": {
4      "title": "Task API",
5      "version": "1.0.0"
6    },
7    "paths": {
8      "/tasks": {
9        "get": {
10          "summary": "Get all tasks",
11          "responses": {
12            "200": {
13              "description": "Successful response"
14            }
15          }
16        }
17      }
18    }
19  }
```

Example OpenAPI JSON Document

# Practice with OpenAPI

Write your own JSON OpenAPI document for a Bookstore API that:

- Is version 2.1.3
- Has path `/books/:bookId` with:
  - Get request to retrieve a book with correct `bookId`

```
1  {
2    "openapi": "3.0.0",
3    "info": {
4      "title": "Task API",
5      "version": "1.0.0"
6    },
7    "paths": {
8      "/tasks": {
9        "get": {
10          "summary": "Get all tasks",
11          "responses": {
12            "200": {
13              "description": "Successful response"
14            }
15          }
16        }
17      }
18    }
19  }
```

Example OpenAPI JSON Document

The background is a solid dark red color. On the left side, there is a stylized, light red illustration of a smartphone. The screen of the phone shows a dark red oval shape with two white, triangular eyes, representing the Postman character. Below the screen, there are several small white dots and a white outline of a lightbulb, suggesting a 'bright idea' or a notification. The word 'Postman' is written in a bold, white, sans-serif font across the middle of the image, partially overlapping the smartphone illustration.

# Postman

# What is Postman?

- Software tool for building, testing, and managing APIs
- Provides a user interface for sending HTTP requests and receiving responses
- Eliminates the need for a functioning frontend to test APIs
- Works with OpenAPI specs.



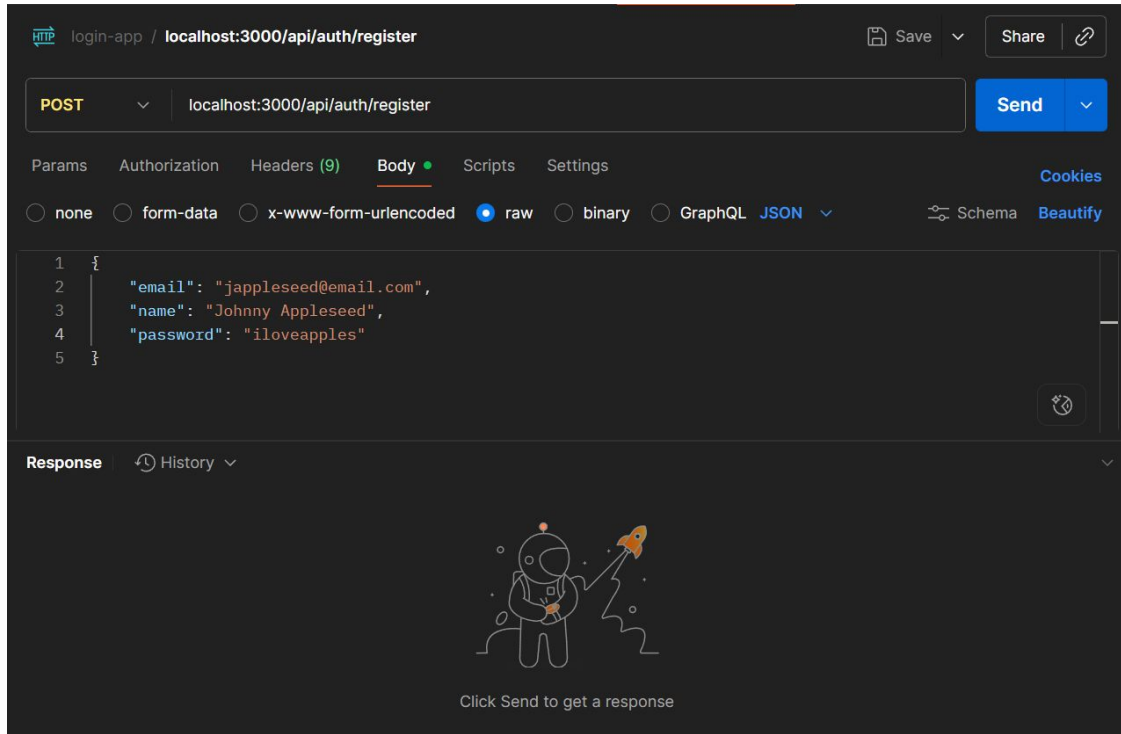
POSTMAN

*Citation: Postman Reviews 2025: Details, Pricing, & Features - G2*

# Some Core Features

- **Send custom HTTP requests and view responses**
  - Status, headers, body, etc.
- **Collections**
  - Organize related requests into groups
  - Ex. a full API's endpoints
- **Tests**
  - Write small JavaScript scripts to verify responses
- **Mock Servers**
  - Simulate real API behavior before backend is ready

# Example: POST Request



The background is a solid dark red color. On the left side, there is a faint, stylized illustration of a smartphone. The screen of the phone shows a cartoon character of a Postman, which is a red envelope with a face, arms, and legs. The phone's interface includes a home button at the bottom and various icons on the screen. The text 'Postman Demo' is centered over the phone illustration.

# Postman Demo



**Questions?**

# Next Bonus Workshop...

## Database Security

- Wednesday 11/05/2025, 6:00 PM–7:00 PM
- Row Level Security (RLS) & Access Control
- Anon & Service Role Keys
- Handling Sensitive Data

# Attendance



Please let us know where we can improve the  
format of the lessons!