Introduction to APIs, Microservices, & cURL

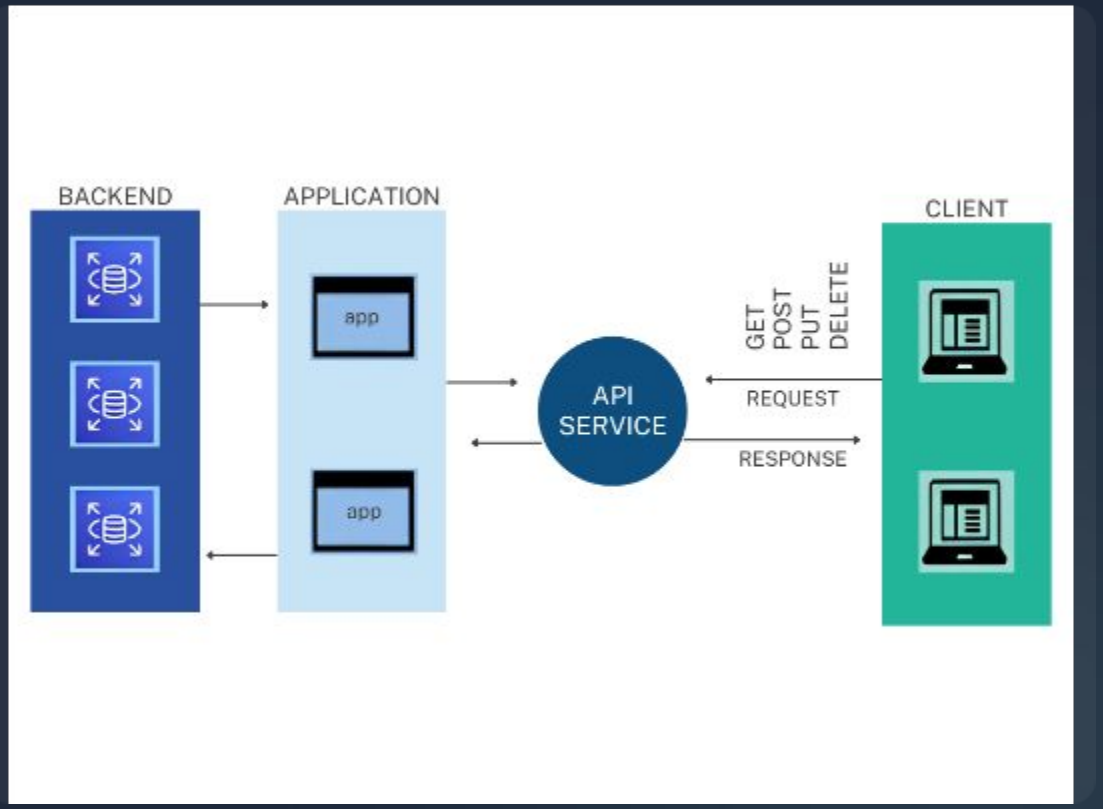
A Practical Lecture for Developers

What is an API?

The "Contract" for Software Communication

Stands for **Application Programming Interface**.

- ✓ It's a set of rules, definitions, and protocols that allow two software components to communicate with each other.
- ✓ Think of it as a **menu in a restaurant**: The menu (API) provides a list of functions you can order.
- ✓ You (the client) don't need to know how the kitchen (the server) works; you just need to know how to place your order.



How APIs Work: The Mailbox Analogy

An API is like a Mailbox

An API is like a mailbox (the **endpoint**) that has specific rules (**protocols**) for how you interact with it.

Your request is the **"verb"** or action you want to perform on that mailbox. Let's see what those actions are.

Common API Verbs (HTTP Methods)

POST (Create)

"Drop a new package in the mailbox." Used to send new data to the server.

GET (Read)

"Check the mailbox contents." Used to retrieve data from the server.

PUT (Update)

"Replace all packages with a new one." Used to update an existing resource.

DELETE (Delete)

"Remove all packages from the mailbox." Used to delete a resource.

Microservices

Building Big Apps from Small Pieces

What is a Microservice?

An architectural style that structures a single application as a collection of small, independent services.

Each service:

- ✓ Is **loosely coupled** from other services.
- ✓ Is **independently deployable**.
- ✓ Is organized around a specific **business capability**.
- ✓ Communicates with other services, often using... **APIs**!

Monolith vs. Microservices

Monolith (The "All-in-One")

Pros: Simple to start, single codebase.

Cons: Hard to scale, difficult to update, a single bug can crash the entire application.

Microservices (The "Building Blocks")

Pros: Scalable (scale only what's needed), resilient (one failure doesn't stop others), flexible (use different tech for different services).

Cons: More complex to manage, potential network latency between services.

cURL

Your Tool for "Talking" to APIs

What is cURL?

Stands for "Client URL"

A command-line tool used to transfer data to or from a server.

It supports many protocols, including HTTP and HTTPS, making it perfect for testing and interacting with APIs directly from your terminal.

It's the developer's "Swiss Army knife" for web requests.

cURL in Action (GET & POST)

GET (Read Data)

The default method. Simply provide the URL to fetch data.

```
curl https://api.example.com/users
```

POST (Send Data)

Use -X POST to specify the method and -d to send data.

```
curl -X POST -d "name=John&age=30"
https://api.example.com/users
```

cURL in Action (PUT & DELETE)

PUT (Update Data)

Use -X PUT to update a specific resource.

```
curl -X PUT -d "name=JohnDoe"
https://api.example.com/users/1
```

DELETE (Remove Data)

Use -X DELETE to remove a specific resource.

```
curl -X DELETE
https://api.example.com/users/1
```