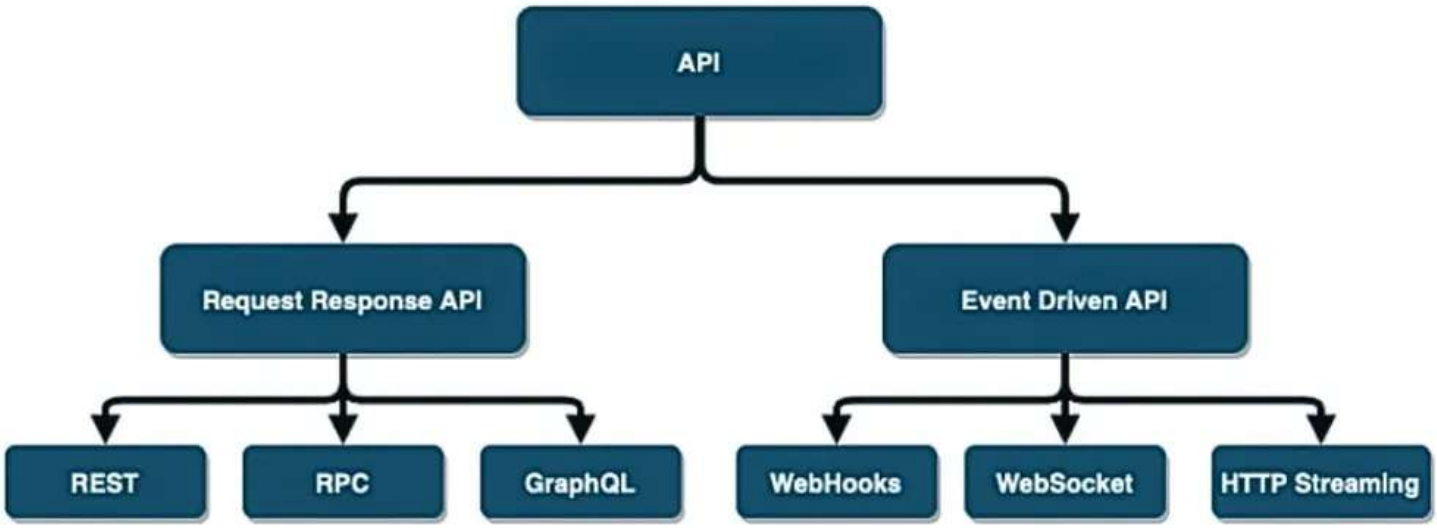
Types of API

(With Example)



Swipe>>

1) REST API

- Stands for **Representational State Transfer**
- Most common, uses **HTTP methods like GET, POST, PUT, DELETE.**
- Simple, scalable, and widely used for web and mobile applications.
- Example: **Fetching user data from a website.**

```
app.get('/user/:id', (req, res) => {
    res.json({ id: req.params.id, name: "John Doe" });
});
```

2) SOAP API

- Stands for **Simple Object Access Protocol**
- Uses **XML-based messaging** for communication.
- More **secure and reliable**, used in banking and enterprise applications.
- Example: Secure transactions between banking servers.

```
<soapenv:Body>
    <GetUserInfo>
        <UserId>123</UserId>
    </GetUserInfo>
</soapenv:Body>
```

3) GraphQL API

- Allows clients to request specific data, **avoiding over-fetching.**
- More efficient for modern applications like social media platforms.
- **Example: Fetching a user's name and profile picture without extra data.**

```
user(id: "123") {
  name
  email
}
```

4) WebSocket API

- Enables real-time, **two-way communication** between client and server.
- Used in **chat apps, gaming, and live stock** updates.
- **Example**: Receiving instant messages in a chat app.

```
const socket = new WebSocket("ws://example.com/socke

socket.onmessage = (event) => {
    console.log("Message from server:", event.data);
};
```

5) gRPC API

- Stands for **Google Remote Procedure Call**
- Uses **Protocol Buffers** (not JSON/XML) for faster communication.
- Ideal for **microservices, streaming,** and large-scale systems.
- Example: **High-speed communication** between backend services.

```
service UserService {
  rpc GetUser (UserRequest) returns (UserResponse);
}
message UserRequest { string id = 1; }
message UserResponse { string name = 1; string email
```