API Design and Management

Mohamed Sweelam

Software Engineer

Outline

Course Objectives

1. Provide good Arabic content for the topic

Course Objectives

1. Provide good Arabic content for the topic
2. Overview of API Design and Management

Course Objectives

1. Provide good Arabic content for the topic
2. Overview of API Design and Management
3. Role and Importance of APIs in Distributed Systems

Course Objectives

1. Provide good Arabic content for the topic
2. Overview of API Design and Management
3. Role and Importance of APIs in Distributed Systems
4. The best practices you should follow today

Understanding APIs

Definition *wikipedia*

Application programming interface (API) is a way for two or more computer programs or components to communicate with each other. It is a type of software interface, offering a service to other pieces of software.

Definition *ChatGPT*

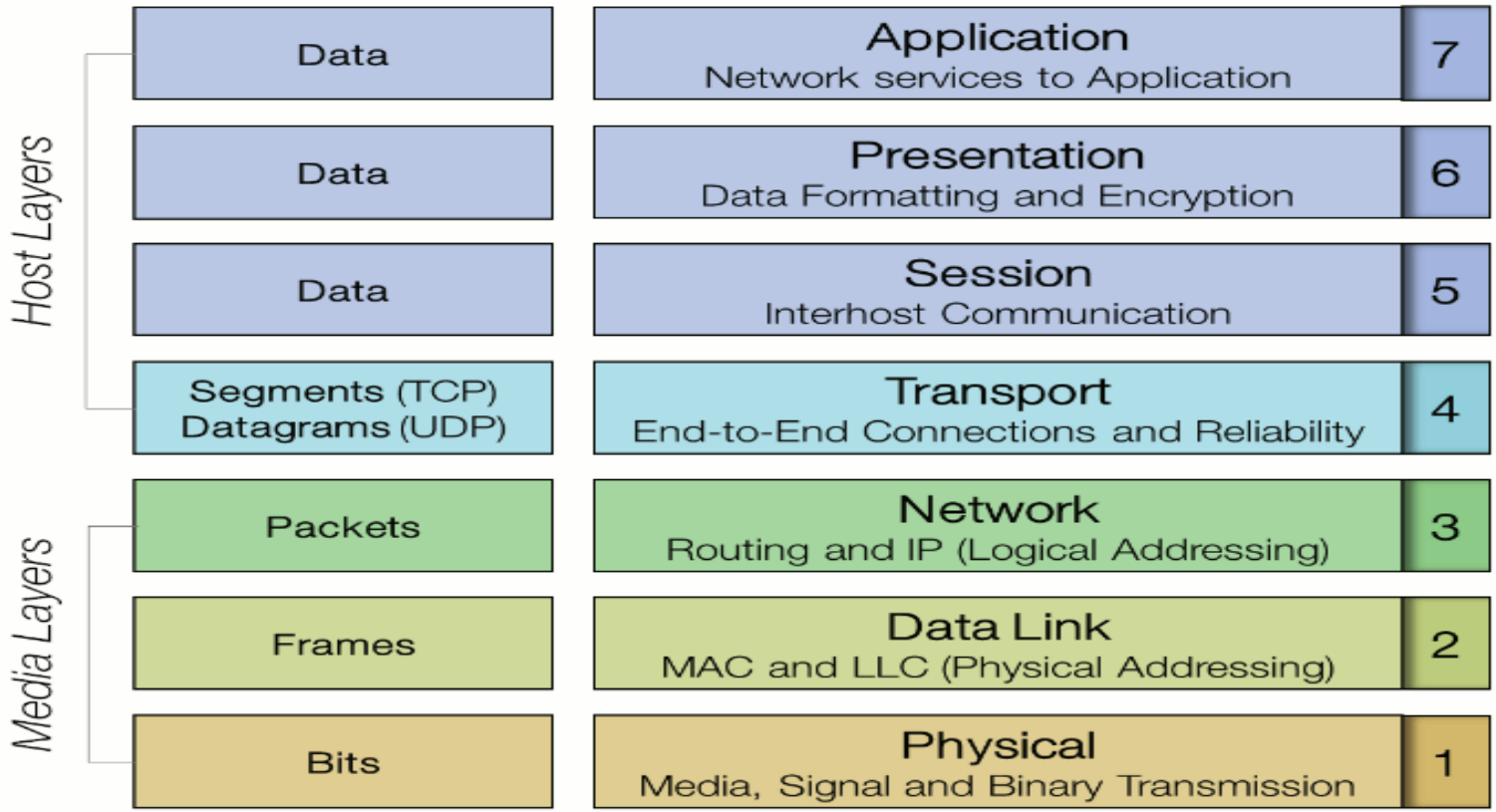
API (Application Programming Interface) is a set of rules, protocols, and tools for building software applications. It specifies how software components should interact and is used to enable the integration between different software systems.

History *wikipedia*

The term "application program interface" is first recorded in a paper called Data structures and techniques for remote computer graphics in 1968. The authors use the term to describe the interaction of an application "Graphics Program" with the rest of the computer system.

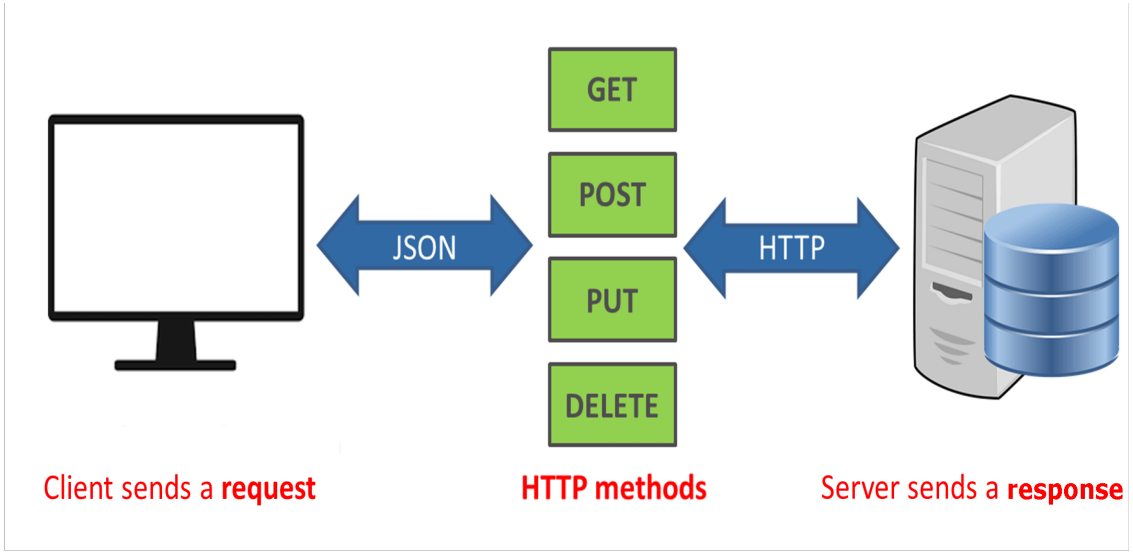
Understanding APIs *OSI Model*

The open systems interconnection (OSI) model is a conceptual model created for Standardization which enables diverse communication systems to communicate using standard protocols.



source: coengoedegebure.com/osi-model

Understanding APIs Example



source: phpenthusiast.com/blog/what-is-rest-api

Understanding APIs (API Types)

There are several types of API, each one serves specific use case.

- Public APIs (Open APIs)
  The APIs are publicly available and can be designed in various ways, taking security into account. However, the main priority is to ensure they are easily consumable by as many clients as possible.

Understanding APIs (API Types)

There are several types of API, each one serves specific use case.

- **Public APIs (Open APIs)**
  The APIs are publicly available and can be designed in various ways, taking security into account. However, the main priority is to ensure they are easily consumable by as many clients as possible.
- **Partner APIs**
  Specialized interfaces that enable organizations to access data and service offerings across businesses (B2B) in order to create unique features within their own applications or services by utilizing a partner's resources.

Understanding APIs (API Types)

There are several types of API, each one serves specific use case.

- **Public APIs (Open APIs)**
  The APIs are publicly available and can be designed in various ways, taking security into account. However, the main priority is to ensure they are easily consumable by as many clients as possible.
- **Partner APIs**
  Specialized interfaces that enable organizations to access data and service offerings across businesses (B2B) in order to create unique features within their own applications or services by utilizing a partner's resources.
- **Internal APIs**
  Intended for use internally by the organizations own developers. These APIs facilitate the transmission of data between different components of a system, enabling process automation.

Understanding APIs (API Types)

There are several types of API, each one serves specific use case.

- **Public APIs (Open APIs)**
  The APIs are publicly available and can be designed in various ways, taking security into account. However, the main priority is to ensure they are easily consumable by as many clients as possible.
- **Partner APIs**
  Specialized interfaces that enable organizations to access data and service offerings across businesses (B2B) in order to create unique features within their own applications or services by utilizing a partner's resources.
- **Internal APIs**
  Intended for use internally by the organizations own developers. These APIs facilitate the transmission of data between different components of a system, enabling process automation.
- **Composite APIs** Executes a series of API requests in a single call.

Understanding APIs (API Types)

There are several types of API, each one serves specific use case.

- **Public APIs (Open APIs)**
  The APIs are publicly available and can be designed in various ways, taking security into account. However, the main priority is to ensure they are easily consumable by as many clients as possible.
- **Partner APIs**
  Specialized interfaces that enable organizations to access data and service offerings across businesses (B2B) in order to create unique features within their own applications or services by utilizing a partner's resources.
- **Internal APIs**
  Intended for use internally by the organizations own developers. These APIs facilitate the transmission of data between different components of a system, enabling process automation.
- **Composite APIs** Executes a series of API requests in a single call.

The types can be designed and developed using two ways

1. API Architectural Style,
2. API Standard Protocol.

Understanding APIs (API Architecture vs API Protocols)

API Architecture Style

It refers to the high-level structural design of the API. It encompasses the standards, and best practices governing how the API is developed, how it interacts with other systems, and how it exposes its functionality and data.
Example: REST

No Restrictions

Architectural style is sensitive to change and enhancement; it relies more on human experience.

API Protocol

It refers to a set of rules and standards used for communication between various software components. The protocol dictates how requests and responses are formatted and transmitted, and what are the restrictions of the communication.
Example: SOAP

Understanding APIs (SOAP API)

- SOAP (Simple Object Access Protocol) is a protocol used for exchanging structured information in web services in computer networks.

Understanding APIs (SOAP API)

- SOAP (Simple Object Access Protocol) is a protocol used for exchanging structured information in web services in computer networks.
- It's a standards-based web services access protocol that has been around for a long time.

Understanding APIs (SOAP API)

- SOAP (Simple Object Access Protocol) is a protocol used for exchanging structured information in web services in computer networks.
- It's a standards-based web services access protocol that has been around for a long time.
- It relies on XML as its message format and usually relies on other application layer protocols, most notably Hypertext Transfer Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP), for message negotiation and transmission.

Understanding APIs (SOAP API)

- SOAP (Simple Object Access Protocol) is a protocol used for exchanging structured information in web services in computer networks.
- It's a standards-based web services access protocol that has been around for a long time.
- It relies on XML as its message format and usually relies on other application layer protocols, most notably Hypertext Transfer Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP), for message negotiation and transmission.

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <!-- header information here -->
  </soap:Header>
  <soap:Body>
    <m:GetPrice xmlns:m="http://www.example.org/stock">
      <m:StockName>IBM</m:StockName>
    </m:GetPrice>
  </soap:Body>
  <soap:Fault>
    <!-- fault information here -->
  </soap:Fault>
</soap:Envelope>
```

API Design Principles

- Fundamentals of Good API Design
- Designing for Scalability and Performance
- API Versioning Strategies

RESTful API Design

- RESTful Architecture Principles
- Designing RESTful Services (Endpoints, HTTP Methods, Status Codes)
- Best Practices in RESTful API

Advanced API Protocols

- Introduction to GraphQL and Its Advantages
- Implementing gRPC for Microservices
- Comparison of Different API Styles

API Documentation and Specification

- Importance of Comprehensive API Documentation
- Tools for API Documentation (Swagger, OpenAPI Specification)
- Maintaining and Versioning API Documentation

API Security

- Authentication and Authorization Mechanisms (OAuth, JWT)
- Securing API Endpoints
- Handling Sensitive Data and Privacy Concerns

API Testing and Quality Assurance

- Writing Effective API Tests
- Tools and Frameworks for API Testing
- Performance Testing and Load Testing for APIs

API Management and Lifecycle

- The Lifecycle of API Development
- API Deployment Strategies
- Monitoring and Analytics for APIs

Conclusion

- Recap of Key Learnings
- Emerging Trends in API Development