CS 646 (Operating System)
Topic : API(Application Programming Interface)

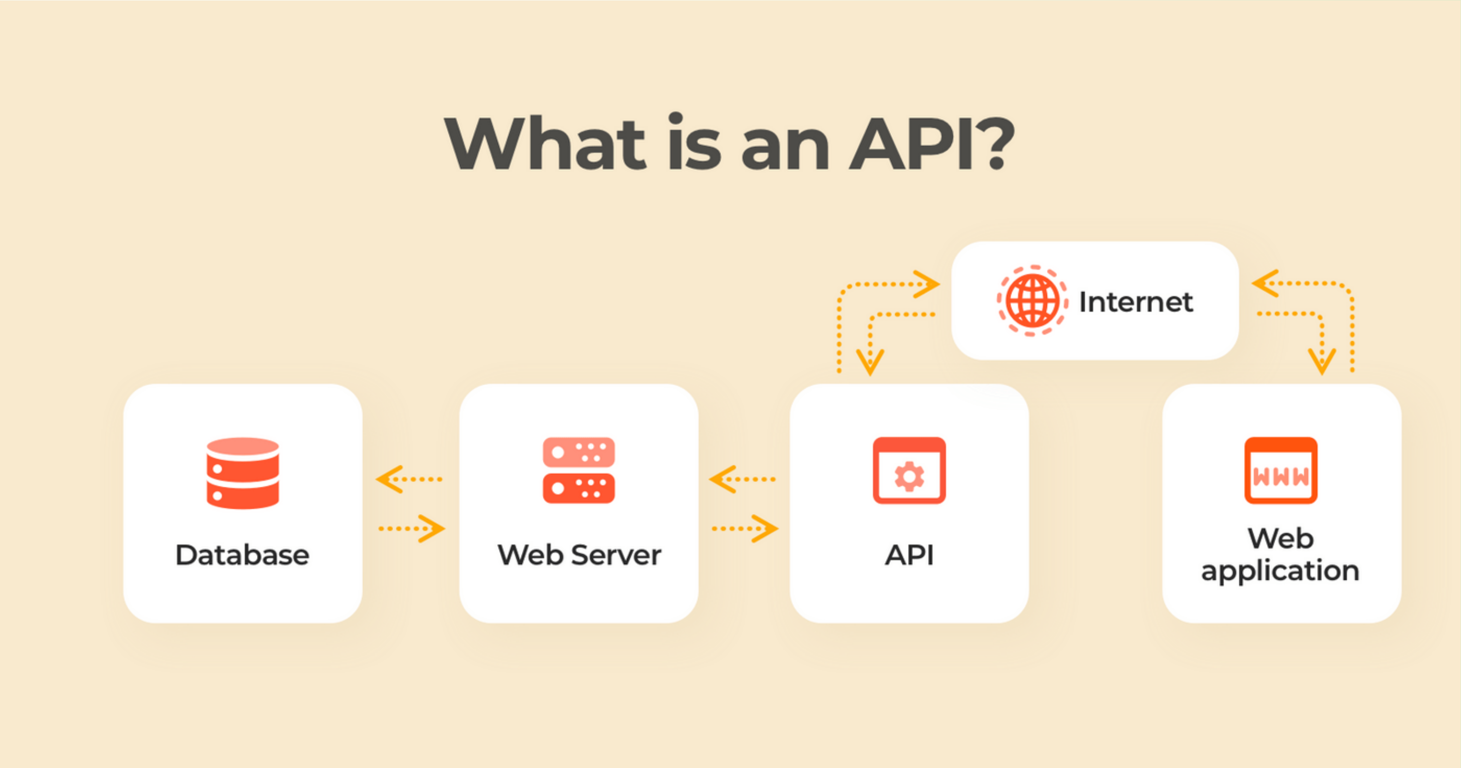
Presented by: Sushmita Sarker

Learning Outcomes:

- What is an API
- How an API works
- Types of API's
- Types of API protocols
- Popular examples of API's you may encounter.
- Why we need API

What is an API?

- API is the acronym **for Application Programming Interface**, which is a connection between computers or between computer programs. It is a type of software interface, offering a service to other pieces of software.



What Is an Example of an API?

When we use an application on our mobile phone, the application connects to the Internet and sends data to a server. The server then retrieves that data, interprets it, performs the necessary actions and sends it back to your phone. The application then interprets that data and presents you with the information you wanted in a readable way. This is what an API is .

How an API works?

A client application initiates an API call to retrieve information—also known as a request. This request is processed from an application to the web server via the API’s Uniform Resource Identifier (URI) and includes a request verb, headers, and sometimes, a request body.

After receiving a valid request, the API makes a call to the external program or web server.

The server sends a response to the API with the requested information.

The API transfers the data to the initial requesting application

Types of APIs

- **Open APIs** are open-source application programming interfaces you can access with the HTTP protocol. Also known as public APIs.

- **Partner APIs** are application programming interfaces exposed to business partners. Typically, developers can access these APIs in self-service mode through a public API developer portal.

- **Internal APIs** are application programming interfaces that remain hidden from external users. These private APIs aren't available for users outside of the company.

- **Composite APIs** combine multiple data or service APIs. These services allow developers to access several endpoints in a single call.

Types of API protocols:

With the increased API usage, certain protocols have been developed to provide users with a set of defined rules that specifies the accepted data types and commands.

- **SOAP** (Simple Object Access Protocol) is an API protocol built with XML, enabling users to send and receive data through SMTP and HTTP.

- **XML-RPC** is a protocol that relies on a specific format of XML to transfer data, whereas SOAP uses a proprietary XML format.

- **JSON-RPC** is a protocol similar to XML-RPC, as they are both remote procedure calls (RPCs), but this one uses JSON instead of XML format to transfer data.

- **REST** (Representational State Transfer) is a set of web API architecture principles, which means there are no official standards. To be a REST API, the interface must adhere to certain architectural constraints.

Common API examples

Here are some popular examples of application programming interfaces

Universal logins: A popular API example is the function that enables people to log in to websites by using their Facebook, Twitter, or Google profile login details.

Third-party payment processing: For example, the now-ubiquitous "Pay with PayPal" function you see on ecommerce websites works through an API. This allows people to pay for products online without exposing any sensitive data or granting access to unauthorized individuals.

Travel booking comparisons: Travel booking sites aggregate thousands of flights, showcasing the cheapest options for every date and destination. This service is made possible through APIs that provide application users with access to the latest information about availability from hotels and airlines.

Google Maps: One of the most common examples of a good API is the Google Maps service. In addition to the core APIs that display static or interactive maps, the app utilizes other APIs and features to provide users with directions or points of interest.

Twitter: Each Tweet contains descriptive core attributes, including an author, a unique ID, a message, a timestamp when it was posted, and geolocation metadata.

Why we need APIs?

- **Improved collaboration:** The average enterprise uses almost 1,200 cloud applications. APIs enable integration so that these platforms and apps can seamlessly communicate with one another. Without APIs, many enterprises would lack connectivity and would suffer from informational silos that compromise productivity and performance.

- **Easier innovation:** APIs offer flexibility, allowing companies to make connections with new business partners, offer new services to their existing market, and, ultimately, access new markets that can generate massive returns and drive digital transformation.

- **Data monetization:** Many companies choose to offer APIs for free, at least initially, so that they can build an audience of developers around their brand and forge relationships with potential business partners. However, if the API grants access to valuable digital assets, you can monetize it by selling access

- **Added security:** APIs create an added layer of protection between your data and a server. Developers can further strengthen API security by using tokens, signatures, and Transport Layer Security (TLS) encryption; by implementing API gateways to manage and authenticate traffic; and by practicing effective API management.

Why API is so Important

- An **application program interface** (API) is code that allows two software programs to communicate with each other. An API defines the correct way for a developer to request services from an operating system (OS) or other application and expose data within different contexts and across multiple channels.
- APIs make life easier for developers
- APIs are also used to control access to hardware devices and software functions that an application may not necessarily have permission to use. That’s why APIs often play a big role in security. This same principle is used on modern mobile operating systems like iOS and Android, where mobile apps have permissions that can be enforced by controlling access to APIs
- File systems that use permissions—as they do on Windows, Mac, and Linux—have those permissions enforced by the file system API.
- APIs are used for communication between services There are APIs for requesting text translation from Google Translate or embedding Facebook comments or tweets from Twitter on a website.

References

1. https://www.mulesoft.com/resources/api/what-is-an-api#:~:text=API%20is%20the%20acronym%20for,you're%20using%20an%20API.
2. https://www.ibm.com/cloud/learn/api#toc-apisweb-se-ByCUYXau
3. https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces
4. https://en.wikipedia.org/wiki/API

Thank you.