**ABOUT APIs**

**By**

**Oghenetejiri Amrasa**

* 1. **DEFINITION OF APIs**

**An API (Application Programming Interface), is a set of defined rules that enable different applications to communicate with each other. It acts as an intermediary layer that processes data transfers between systems, letting companies open their application data and functionality to extend third-party developers, business partners and internal departments within their companies.**

**2.1 Types OF APIs**

1. **Private**
2. **Public/Partner**
3. **External**

**2.1.1 Private APIs**

**Private APIs, also known as Internal APIs or Enterprise APIs, are application programming interfaces that are designed and developed for exclusive use within a specific organization or company. Unlike public APIs, which are made available to external developers or third-party applications, private APIs are intended for internal use only and are not exposed to the public.**

**Private APIs serve as the communication bridge between different systems, applications, or services within an organization. They enable seamless data exchange and integration between internal software components, helping to streamline business processes, improve efficiency, and enhance collaboration within the organization.**

**Here are some key aspects and characteristics of private APIs:**

**1. Limited Access: Private APIs are accessible only within the organization that develops and maintains them. They are typically protected by authentication mechanisms such as API keys, tokens, or other access control methods to ensure that only authorized internal applications or systems can interact with them.**

**2. Tailored for Internal Needs: Private APIs are designed specifically to meet the unique requirements and workflows of the organization. They are developed with a deep understanding of the internal systems, data structures, and business processes, allowing for tighter integration and optimized interactions.**

**3. Security and Compliance: Since private APIs deal with internal data and systems, security and compliance measures play a crucial role. Organizations can implement additional security measures like encryption, access controls, and audit logs to protect sensitive data and ensure compliance with industry regulations.**

**4. Integration of Legacy Systems: Private APIs often serve as a means to connect modern applications and services with legacy systems that may not have built-in integration capabilities. By encapsulating the functionality of legacy systems behind APIs, organizations can modernize their software infrastructure while maintaining compatibility with existing systems.**

**5. Service-Oriented Architecture (SOA): Private APIs are often a key component of a service-oriented architecture, where different internal services expose APIs to enable communication and collaboration. These APIs allow services to interact with each other, share data, and invoke functionalities, facilitating a modular and scalable architecture.**

**6. Enhanced Efficiency and Productivity: Private APIs enable different internal applications or systems to share data and functionality seamlessly. This promotes automation, reduces manual effort, and improves overall efficiency and productivity within the organization. For example, private APIs can be used to integrate a customer relationship management (CRM) system with an enterprise resource planning (ERP) system, allowing for automatic synchronization of customer data and order information.**

**7. Developer Empowerment: Private APIs empower internal developers by providing them with reusable components and services that can be leveraged across different projects. Developers can build new applications, services, or integration more efficiently by utilizing the capabilities exposed through private APIs.**

**It's important to note that while private APIs are not publicly accessible, they still require proper documentation and developer support within the organization. Clear documentation helps internal developers understand the API's endpoints, request/response formats, authentication methods, and any usage guidelines or best practices.**

**Overall, private APIs enable organizations to unlock the potential of their internal systems, improve connectivity, and streamline operations. They offer a secure and controlled environment for integrating internal software components and services, fostering collaboration, and enhancing productivity within the organization.**

**2.1.2 Public**

**Public APIs, also known as Application Programming Interfaces, are sets of rules and protocols that allow different software applications to communicate and interact with each other. These APIs are made available by companies, organizations, or developers to provide access to specific functionalities or data of their software or platforms.**

**Public APIs serve as a bridge between different software systems, enabling them to exchange information and perform actions seamlessly. They define the methods and data formats that developers need to follow in order to make requests and receive responses from the API provider. These APIs typically use standard protocols such as HTTP (Hypertext Transfer Protocol) to facilitate communication over the internet.**

**Public APIs can offer a wide range of functionalities and services. Some common examples include:**

**1. Social media APIs: Platforms like Facebook, Twitter, or Instagram provide APIs that allow developers to integrate their applications with social media networks. These APIs enable developers to retrieve user data, post updates, or interact with social media features programmatically.**

**2. Mapping and geolocation APIs: Services like Google Maps offer APIs that provide access to mapping data, geocoding, routing, and other location-related services. Developers can use these APIs to embed maps, display directions, or perform location-based searches within their applications.**

**3. Payment gateway APIs: Companies like PayPal or Stripe offer APIs that facilitate online payment processing. These APIs allow developers to integrate payment functionality into their applications, enabling users to make transactions securely.**

**4. Weather APIs: Weather services such as OpenWeatherMap provide APIs that deliver weather forecasts, current conditions, and historical data. Developers can leverage these APIs to display weather information within their applications or build weather-related features.**

**5. E-commerce APIs: Online marketplaces like Amazon or Shopify offer APIs that enable developers to manage products, process orders, and retrieve data related to e-commerce activities. These APIs empower developers to create custom storefronts, synchronize inventory, or build integration with e-commerce platforms.**

**Public APIs provide numerous benefits to developers and businesses. They allow for the creation of new applications that can leverage existing functionalities and data, saving time and resources. APIs also promote interoperability, enabling different systems to work together seamlessly. They foster innovation, as developers can build upon the capabilities of existing platforms to create new services or improve user experiences.**

**To access a public API, developers typically need to obtain an API key or token from the provider, which authenticates and tracks their usage. The API documentation provided by the provider serves as a guide, explaining how to make requests, interpret responses, and handle errors.**

**2.1.3 External APIs**

**External APIs, also known as Third-Party APIs or Partner APIs, are application programming interfaces provided by external organizations or service providers to allow developers to access their functionalities, data, or services. These APIs enable developers to integrate external resources, services, or features into their own applications, extending their capabilities and providing additional value to users.**

**Here are some key aspects and characteristics of external APIs:**

**1. Accessing External Services: External APIs provide a standardized way for developers to interact with external services or platforms. These services can range from social media networks (e.g., Facebook, Twitter), payment gateways (e.g., PayPal, Stripe), cloud storage providers (e.g., Amazon S3, Google Drive), or various specialized services (e.g., weather data, image recognition). By integrating external APIs, developers can leverage the functionalities offered by these services without having to build them from scratch.**

**2. Enabling Data Exchange: External APIs facilitate the exchange of data between different systems or applications. They allow developers to retrieve data from external sources, submit data for processing, or update information in external systems. For example, an e-commerce application might use an external shipping API to calculate shipping costs or a weather API to display localized weather information to users.**

**3. Pre-Built Functionality: External APIs provide pre-built functionalities that developers can integrate into their applications, saving time and effort. Instead of reinventing the wheel, developers can rely on the capabilities of external APIs to provide features such as authentication, search, messaging, analytics, or real-time communication. This accelerates development cycles and enables developers to focus on the core functionality of their applications.**

**4. Standardized Interfaces: External APIs typically follow standardized protocols and data formats, such as REST (Representational State Transfer) or GraphQL, making it easier for developers to understand and utilize them. They provide clear documentation and specifications, including API endpoints, request/response structures, authentication mechanisms, and usage limits. This standardization promotes interoperability and allows developers to work with different APIs more efficiently.**

**5. Monetization Opportunities: Some external APIs are offered as part of a monetization strategy by the API provider. They may provide free access with limitations, tiered pricing models based on usage, or subscription plans. Developers can utilize these APIs to enhance their applications and monetize their services by offering additional features or accessing premium data or services.**

**6. Ecosystem and Collaboration: External APIs foster the growth of developer communities and ecosystems centered around specific platforms or services. These communities encourage collaboration, knowledge sharing, and the development of complementary tools and services. Developers can benefit from the collective expertise and resources of the ecosystem while building applications that integrate with external APIs.**

**7. Versioning and Stability: External APIs often undergo updates and improvements over time. API providers usually release new versions of their APIs to introduce new features, enhance performance, or address issues. Proper versioning and deprecation strategies are employed to ensure backward compatibility and give developers time to adapt their applications to new API versions.**

**Developers using external APIs need to consider factors such as API reliability, performance, support, and terms of use. They should also be aware of any usage limits, rate throttling, or data usage restrictions imposed by the API provider.**

**In summary, external APIs offer a way for developers to access and integrate external services, functionalities, and data into their own applications. They provide a means to extend application capabilities, leverage pre-built functionalities, and integrate with specialized services or platforms. Proper utilization of external APIs can enhance the functionality, user experience, and value of applications while reducing development time and effort.**

* 1. **Use cases**

1. **Pay with PayPal**
2. **Weather apps**
3. **Google maps**

**References**

**https://www.freecodecamp.org**

**<https://www.ibm.com/topics/api>**

**https://www.emizentech.com/blog/api-examples.html**