

Entendendo APIs

Por Escola Dnc

Introdução

APIs (Application Programming Interfaces) são conjuntos de ferramentas, definições e protocolos para a criação de aplicações de software. Elas conectam soluções e serviços sem a necessidade de saber como esses elementos foram implementados.

Uma API é essencialmente uma forma de relacionamento e comunicação entre diferentes sistemas e aplicações. Neste ebook, vamos explorar em detalhes o que são APIs, como funcionam na prática e os principais conceitos envolvidos.

O que é uma API?

Uma API é como uma "conversa" entre duas pontas:

- O **cliente**: que consome e utiliza a API
- O **servidor**: que disponibiliza e serve a API

O **cliente** faz solicitações e consome as respostas. Ele não precisa necessariamente saber todos os detalhes técnicos de como a API foi implementada no servidor.

Já o **servidor** disponibiliza endpoints e recursos, processa as solicitações do cliente e retorna as respostas adequadas.

Essa comunicação acontece através de **protocolos** e padrões bem definidos, que veremos mais adiante.

Exemplo Prático

Vamos imaginar um prédio comercial, com várias empresas em cada andar. Cada empresa oferece produtos e serviços diferentes:

- Andar 1: Empresa de hambúrgueres
- Andar 2: Empresa de sorvetes
- Andar 3: Empresa de salgados

Agora imagine que na portaria deste prédio tem um **porteiro**. Este porteiro seria como uma **API** que intermedia a comunicação entre clientes externos e as empresas internas.

Se um cliente chega na portaria e pede um hambúrguer, o que acontece?

1. O cliente faz uma **solicitação** (quer um hambúrguer)
2. O porteiro **processa** esta solicitação e sabe que deve encaminhar para a empresa de hambúrgueres
3. A empresa de hambúrgueres **atende** a solicitação e prepara o pedido
4. O porteiro **retorna** o hambúrguer pronto para o cliente
5. O cliente recebe o que pediu sem precisar saber os detalhes internos de cada empresa

Essa interação ilustra bem o conceito de uma API. O porteiro faz o "meio de campo" entre clientes e servidores, encaminhando solicitações e respostas entre as partes.

Principais Características

As APIs possuem algumas características importantes:

- **Protocolos bem definidos:** a comunicação acontece através de protocolos (como HTTP, WebSocket, etc) com estruturas e padrões claros e documentados.
- **Abstração:** o cliente consome a API sem precisar saber da complexidade interna do servidor. Ele apenas conhece os recursos disponibilizados.
- **Desacoplamento:** clientes e servidores não precisam saber muitos detalhes um do outro. Basta que a API seja bem documentada.
- **Escalabilidade:** APIs permitem integrar facilmente novos clientes e expandir a capacidade conforme a demanda cresce.
- **Reutilização:** uma API bem projetada permite o reuso entre vários produtos e clientes.

Principais Tipos de API

Existem diferentes tipos de APIs, cada uma com suas características e casos de uso:

API Web Service

São APIs acessíveis via protocolos web como HTTP e HTTPS. São amplamente utilizadas para integrar aplicações web, mobile e internet das coisas. Costumam retornar dados no formato JSON ou XML.

Exemplos: API do Twitter, Stripe, etc.

API de Banco de Dados

Permite acesso direto e integração com bancos de dados como MySQL, MongoDB, etc. Usa linguagens de consulta como SQL para buscar e manipular dados.

API de Hardware

Para se comunicar com hardwares e dispositivos físicos como impressoras, scanners, equipamentos médicos, etc. Geralmente usam padrões específicos.

API de Sistema Operacional

Interface para acessar recursos e serviços do sistema operacional: rede, arquivos, periféricos, execução de programas, etc.

Exemplos: API do Windows, Linux, drivers de dispositivo.

API Webhook

Permite que aplicações se registrem para receber atualizações automáticas de eventos/dados de outra aplicação. Muito usado para notificações em tempo real.

Arquitetura de uma API

A arquitetura básica de uma API envolve os seguintes componentes:

Endpoints

São as "portas de entrada" onde os clientes fazem requests e recebem responses. Cada endpoint tem uma URI (endereço) e geralmente retorna dados em JSON ou XML.

```
GET https://api.exemplo.com/users
```

Gateway da API

Responsável por receber todas as chamadas à API e roteá-las para os serviços certos internamente. Também trata segurança, monitoramento, caching, etc.

Business Logic

Onde estão as regras de negócio e lógica de aplicação. Processa as requests recebidas e retorna as informações solicitadas.

Data Layer

Camada que abstrai o armazenamento físico dos dados, geralmente em um banco de dados. Responsável por persistir e buscar dados quando solicitado pela business logic.

Developer Portal

Site com documentação onde os desenvolvedores podem aprender a usar a API: referência de endpoints, parâmetros, autenticação, exemplos, SDKs, etc.

Principais Protocolos de API

As APIs utilizam protocolos bem definidos para padronizar a comunicação entre clientes e servidores. Os mais comuns são:

HTTP

O protocolo HTTP (Hypertext Transfer Protocol) é a base da comunicação entre aplicações web. Usa verbos (GET, POST, PUT, DELETE, etc) para indicar ações.

É simples, onnipresente na web e permite comunicação stateless (sem estado entre requests). Por isso, é o protocolo mais utilizado para APIs web.

WebSocket

Permite comunicação bidirecional e em tempo real entre cliente e servidor através de uma conexão persistente. Muito usado em aplicações que exigem troca contínua de mensagens.

GraphQL

Protocolo criado pelo Facebook para APIs. Permite que o cliente especifique exatamente os dados que deseja buscar, com resposta pré-estruturada. Agiliza o

desenvolvimento.

gRPC

Framework de RPC (Remote Procedure Call) que usa HTTP/2 para comunicação eficiente, com foco em performance e produtividade. Indicado para APIs internas de microsserviços.

Boas Práticas para APIs

Para que uma API seja bem projetada e atenda bem seus clientes, algumas boas práticas devem ser seguidas:

- **Simplicidade:** evite complexidade desnecessária. Uma API bem feita é simples de entender e usar.
- **Consistência:** segue padrões uniformes para nomes, parâmetros, payloads de requests/responses.
- **Documentação:** tenha uma documentação clara e objetiva. É a "cara" da sua API para os desenvolvedores.
- **Segurança:** proteja o acesso com protocolos de autenticação sólidos como OAuth e JWT.
- **Performance:** assegure velocidade e escalabilidade com caching, compressão de dados, paginação, etc.
- **Monitoramento:** monitorar métricas, logs e alertas é essencial para manter a API funcionando e melhorá-la.
- **Versionamento:** versões permitem introduzir mudanças sem quebrar clientes existentes.

Seguindo essas práticas, você garante que sua API atenda bem às necessidades de negócio e técnicas.

Conclusão

Neste ebook, apresentamos uma visão geral sobre APIs: o que são, seus tipos, protocolos, arquitetura, boas práticas e como funcionam na prática através de exemplos.

APIs são a base para integração entre diferentes aplicações e ambientes computacionais modernos.

Entendendo bem os conceitos apresentados aqui, você estará preparado para começar a desenvolver, consumir ou projetar APIs para resolver problemas de integração e comunicação em sistemas.

Espero que as informações apresentadas sejam úteis! Fique à vontade para entrar em contato em caso de dúvidas.

