

MATERIAL EXTRA

Fundamentos de API

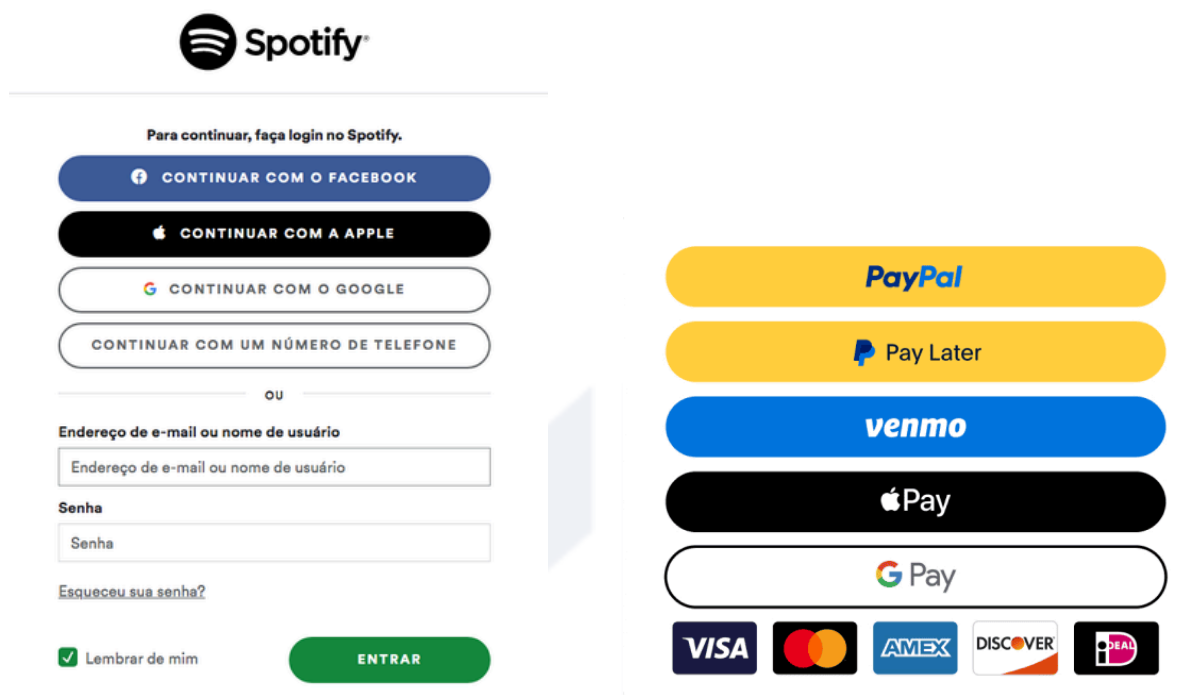


API

Uma **API** (*Application Programming Interface*, ou Interface de Programação de Aplicações) é um conjunto de definições e protocolos que permitem a comunicação entre diferentes sistemas, aplicativos ou plataformas. A API atua como uma ponte que conecta um software a outro, facilitando a troca de informações e o uso de funcionalidades sem que os desenvolvedores precisem compreender ou modificar o funcionamento interno de cada sistema.

Por exemplo, considere um aplicativo de previsão do tempo. Esse aplicativo não coleta os dados climáticos diretamente; em vez disso, ele usa uma API de um serviço meteorológico, como o [OpenWeather](#), para obter informações como temperatura, umidade e previsão para os próximos dias. Esse processo economiza tempo e esforço, pois o desenvolvedor do aplicativo não precisa construir uma infraestrutura para coletar dados climáticos — ele apenas faz um "pedido" à API, que retorna as informações necessárias.

APIs também são amplamente usadas em sites e aplicativos modernos. Um exemplo comum é a funcionalidade "Entrar com o Google" ou "Entrar com o Facebook", disponível em muitos serviços online. Nesse caso, a API dessas plataformas é usada para autenticar o usuário e compartilhar informações básicas de forma segura e padronizada. Outro exemplo é o uso de APIs de pagamento, como as oferecidas pelo PayPal ou Stripe, que permitem que sites e aplicativos processem pagamentos online sem precisar implementar sistemas financeiros complexos.



The image shows a Spotify login and payment interface. On the left, the Spotify logo is at the top. Below it, a message says "Para continuar, faça login no Spotify." (To continue, log in to Spotify). There are four buttons for social login: "CONTINUAR COM O FACEBOOK", "CONTINUAR COM A APPLE", "CONTINUAR COM O GOOGLE", and "CONTINUAR COM UM NÚMERO DE TELEFONE". Below these is a section for email or phone login, with fields for "Endereço de e-mail ou nome de usuário" and "Senha", and a link for "Esqueceu sua senha?". There is also a "Lembrar de mim" checkbox and an "ENTRAR" button. On the right, there are five payment options: "PayPal", "Pay Later", "venmo", "Apple Pay", and "G Pay". At the bottom right, there are logos for "VISA", "MasterCard", "AMEX", "DISCOVER", and "DEAL".

Em resumo, as APIs são ferramentas essenciais que tornam o desenvolvimento de software mais eficiente, promovendo a integração de sistemas e o reaproveitamento de funcionalidades existentes. Isso permite que os desenvolvedores se concentrem na criação de soluções inovadoras, enquanto delegam tarefas específicas a APIs confiáveis e bem estruturadas.

Como funcionam as APIs?

As APIs funcionam compartilhando dados entre aplicativos, sistemas e dispositivos. Isso acontece por meio de um ciclo de solicitação e resposta. A solicitação é enviada para a API, que recupera os dados e os retorna ao usuário. Aqui está uma visão geral de alto nível de como esse processo funciona.

1. Cliente de API

O cliente da API é responsável por iniciar a conversa enviando a solicitação ao servidor da API. A solicitação pode ser acionada de várias maneiras. Por exemplo, um usuário pode iniciar uma solicitação de API inserindo um termo de pesquisa ou clicando em um botão. As solicitações de API também podem ser acionadas por eventos externos, como uma notificação de outro aplicativo.

2. Solicitação de API

Uma solicitação de API terá uma aparência e um comportamento diferentes dependendo do tipo de API, mas normalmente incluirá os seguintes componentes:

- ◆ **Extremidade:** Um endpoint de API é um URL dedicado que fornece acesso a um recurso específico. Por exemplo, o `/articles` endpoint em um aplicativo de blog incluiria a lógica para processar todas as solicitações relacionadas a artigos.
- ◆ **Método:** O método da solicitação indica o tipo de operação que o cliente gostaria de executar em um determinado recurso. As APIs REST podem ser acessadas por meio de métodos HTTP padrão, que executam ações comuns, como recuperar, criar, atualizar e excluir dados.
- ◆ **Cabeçalhos de solicitação:** Os cabeçalhos de solicitação são pares de chave-valor que fornecem detalhes extras sobre a solicitação, como seu tipo de conteúdo ou credenciais de autenticação.

◆ **Parâmetros:** Parâmetros são as variáveis que são passadas para um endpoint de API para fornecer instruções específicas para a API processar. Esses parâmetros podem ser incluídos na solicitação de API como parte da URL, na string de consulta ou no corpo da solicitação. Por exemplo, o ponto de extremidade `/articles` de uma API de blog pode aceitar um parâmetro "topic", que seria usado para acessar e retornar artigos sobre um tópico específico.

◆ **Corpo da solicitação:** O corpo é a parte principal da solicitação e inclui os dados reais necessários para criar ou atualizar um recurso. Por exemplo, se você estivesse criando um novo artigo em um aplicativo de blog, o corpo da solicitação provavelmente incluiria o conteúdo, o título e o autor do artigo.

3. Servidor de API

O cliente da API envia a solicitação para o servidor da API, que é responsável por lidar com a autenticação, validar dados de entrada e recuperar ou manipular dados.

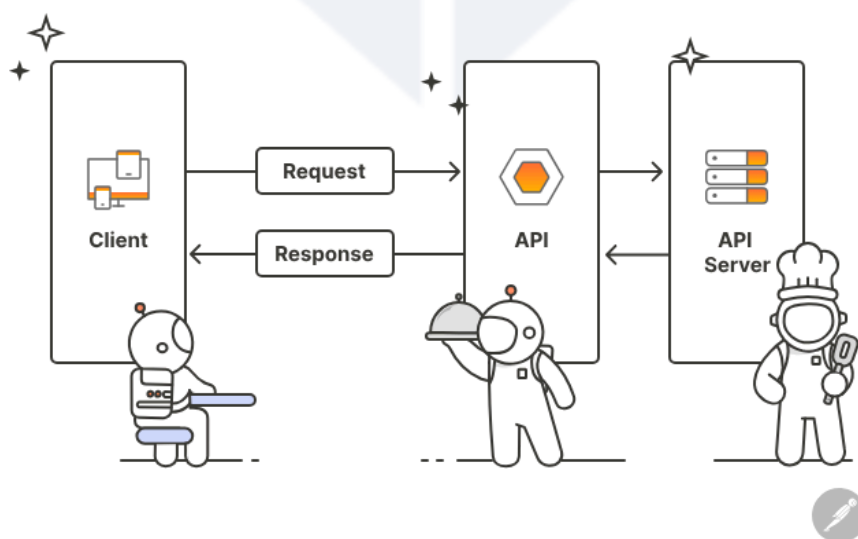
4. Resposta da API

Por fim, o servidor de API envia uma resposta ao cliente. A resposta da API normalmente inclui os seguintes componentes:

◆ **Cabeçalhos de resposta:** os cabeçalhos de resposta HTTP são muito semelhantes aos cabeçalhos de solicitação, exceto que são usados para fornecer informações adicionais sobre a resposta do servidor.

- ◆ **Código de status:** Os códigos de status HTTP são códigos de três dígitos que indicam o resultado de uma solicitação de API. Alguns dos códigos de status mais comuns incluem 200 OK, que indica que o servidor retornou com êxito os dados solicitados, 201 Criado, que indica que o servidor criou com êxito um novo recurso, e 404 Não Encontrado, que indica que o servidor não conseguiu encontrar o recurso solicitado.
- ◆ **Corpo da resposta:** O corpo da resposta inclui os dados ou o conteúdo real que o cliente solicitou ou uma mensagem de erro se algo der errado.

Para entender melhor esse processo, pode ser útil pensar em APIs como restaurantes. Nessa metáfora, o cliente é como o usuário, que diz ao garçom o que ele quer. O garçom é como um cliente de API, recebendo o pedido do cliente e traduzindo-o em instruções fáceis de seguir para a cozinha, às vezes usando códigos ou abreviações específicas que a equipe da cozinha reconhecerá. A equipe da cozinha é como o servidor API porque cria o pedido de acordo com as especificações do cliente e o entrega ao garçom, que o entrega ao cliente.



Disponível em [What is an API? A Beginner's Guide to APIs | Postman](https://www.postman.com/resources/guides/what-is-an-api/)

Quais são os benefícios das APIs?

As APIs conectam vários sistemas de software, aplicativos e dispositivos, permitindo que eles se comuniquem uns com os outros. Isso desbloqueia muitos benefícios, desde experiências de usuário aprimoradas até maior eficiência nos negócios. As vantagens mais comuns das APIs incluem:

- **Automação:** As APIs podem ser usadas para automatizar o trabalho repetitivo e demorado para que os humanos possam se concentrar em tarefas mais complexas. Isso melhora a produtividade, especialmente para desenvolvedores e testadores.
- **Inovação:** As APIs públicas podem ser usadas por equipes de engenharia externas, o que estimula a inovação e acelera o desenvolvimento, permitindo que os desenvolvedores reaproveitem a funcionalidade existente para criar novas experiências digitais.
- **Segurança:** As APIs podem fornecer uma camada adicional de proteção contra violações não autorizadas, exigindo autenticação e autorização para qualquer solicitação de acesso a dados confidenciais.
- **Eficiência de custos:** As APIs fornecem acesso a ferramentas e infraestrutura úteis de terceiros, o que ajuda as empresas a evitar as despesas de criação de sistemas internos complexos.

Tipos de APIs

Existem muitos tipos diferentes de APIs e maneiras de categorizá-las. Por exemplo, você pode categorizar APIs por quem tem acesso a elas. Essa estrutura organizacional inclui:

- **APIs privadas:** As APIs privadas, também conhecidas como APIs internas, são usadas para conectar diferentes componentes de software em uma única organização e não estão disponíveis para uso de terceiros. Por exemplo, um aplicativo de mídia social pode ter uma API privada que lida com o fluxo de trabalho de login, outra API privada que lida com o feed e ainda outra API privada que facilita a comunicação entre os usuários. Alguns aplicativos podem incluir dezenas ou até centenas de APIs privadas.
- **APIs públicas:** As APIs públicas fornecem acesso público aos dados, funcionalidades ou serviços de uma organização, que desenvolvedores de terceiros podem integrar em seus próprios aplicativos. Algumas APIs públicas estão disponíveis gratuitamente, enquanto outras são oferecidas como produtos faturáveis. Por exemplo, um aplicativo de comércio eletrônico pode incorporar uma API de pagamento pública, como o Stripe, para lidar com o processamento de pagamentos sem precisar criar essa funcionalidade do zero.
- **APIs de parceiros:** As APIs de parceiros permitem que duas ou mais empresas compartilhem dados ou funcionalidades para colaborar em um projeto. Eles não estão disponíveis para o público em geral e, portanto, aproveitam os mecanismos de autenticação para garantir que sejam usados apenas por parceiros autorizados.

Estilos de arquitetura de API mais comuns

Você também pode categorizar APIs de acordo com seu estilo de arquitetura, que são muitas. Os estilos arquitetônicos mais usados são:

- **REST:** Conforme discutido acima, o REST é a arquitetura de API mais popular para transferência de dados pela Internet. Em um contexto RESTful, os recursos podem ser acessados por meio de endpoints e as operações são executadas nesses recursos com métodos HTTP padrão, como GET, POST, PUT e DELETE.
- **SOAP:** Significa *Simple Object Access Protocol* e usa XML para transferir mensagens altamente estruturadas entre um cliente e um servidor. O SOAP é frequentemente usado em ambientes corporativos ou sistemas legados e, embora inclua recursos avançados de segurança, pode ser mais lento do que outras arquiteturas de API.
- **GraphQL:** O GraphQL é uma linguagem de consulta de código aberto que permite que os clientes interajam com um único endpoint de API para recuperar os dados exatos de que precisam, sem encadear várias solicitações. Essa abordagem reduz o número de viagens de ida e volta entre o cliente e o servidor, o que pode ser útil para aplicativos que podem ser executados em conexões de rede lentas ou não confiáveis.
- **gRPC:** RPC significa *Chamada de Procedimento Remoto*, e as APIs gRPC foram originadas pelo Google. Nas arquiteturas gRPC, um cliente pode chamar um servidor como se fosse um objeto local, o que facilita a comunicação entre aplicativos e sistemas distribuídos.

- **Webhooks:** Os webhooks são usados para implementar arquiteturas orientadas a eventos, nas quais as solicitações são enviadas automaticamente em resposta a gatilhos baseados em eventos. Por exemplo, quando um evento específico ocorre em um aplicativo, como um pagamento sendo feito, o aplicativo pode enviar uma solicitação HTTP para uma URL de webhook pré-configurada com os dados de evento relevantes na carga útil da solicitação. O sistema que recebe o webhook pode processar o evento e tomar a ação apropriada.

Casos de uso comuns de API

As APIs são extremamente versáteis e oferecem suporte a uma ampla variedade de casos de uso que incluem:

1. Integração com sistemas internos e externos

Um dos motivos mais comuns pelos quais os desenvolvedores recorrem às APIs é integrar um sistema a outro. Por exemplo, você pode usar uma API para integrar seu sistema de gerenciamento de relacionamento com o cliente (CRM) ao seu sistema de automação de marketing, o que permitiria enviar automaticamente um e-mail de marketing quando um representante de vendas adicionasse um novo cliente em potencial ao CRM.

2. Adicionando ou aprimorando a funcionalidade

As APIs permitem que você incorpore funcionalidades adicionais ao seu aplicativo, o que pode melhorar a experiência dos clientes. Por exemplo, se você estiver trabalhando em um aplicativo de entrega de comida, poderá incorporar uma API de mapeamento de terceiros para permitir que os usuários rastreiem seus pedidos enquanto eles estão a caminho.

3. Conectando dispositivos IoT

As APIs são essenciais para o ecossistema da *Internet das Coisas* (IoT), que inclui dispositivos como relógios inteligentes, rastreadores de fitness, campainhas e eletrodomésticos. Sem APIs, esses dispositivos não seriam capazes de se conectar à nuvem – ou uns aos outros – o que os tornaria inúteis.

4. Criação de sistemas mais escaláveis

As APIs são usadas para implementar arquiteturas baseadas em microsserviços, nas quais os aplicativos são criados como uma coleção de pequenos serviços que se comunicam entre si por meio de APIs privadas. Os microsserviços são gerenciados, implantados e provisionados independentemente uns dos outros, o que permite que as equipes dimensionem seus sistemas de maneira confiável e econômica.

5. Redução de custos

As APIs ajudam as organizações a reduzir os custos operacionais automatizando tarefas demoradas, como envio de e-mails, extração de relatórios e compartilhamento de dados entre sistemas. Eles também podem reduzir os custos de desenvolvimento, permitindo que as equipes reutilizem a funcionalidade existente, em vez de reinventar a roda.

6. Melhorar a segurança e a governança organizacional

As APIs alimentam muitos fluxos de trabalho que são essenciais para a segurança organizacional. Por exemplo, o login único (SSO), que permite que os usuários usem um nome de usuário e senha para vários sistemas, é possibilitado por APIs. As APIs também são usadas para aplicar e automatizar regras e políticas de governança corporativa, como a exigência de que as despesas sejam aprovadas antes que os funcionários sejam reembolsados.

Esta lista está longe de ser exaustiva e continuará crescendo à medida que os desenvolvedores continuarem a criar soluções inovadoras que mudam a maneira como vivemos, trabalhamos e interagimos uns com os outros.

Exemplos de APIs populares

Se você está procurando exemplos reais de APIs, um bom lugar para começar é um catálogo de APIs públicas, como a [Rede de APIs Públicas do Postman](#). A Rede de APIs Públicas, que oferece suporte a uma comunidade de mais de 35 milhões de desenvolvedores, é uma biblioteca de APIs pesquisável e altamente organizada que facilita a localização da API certa para você. Você pode usar a funcionalidade de pesquisa da rede se estiver procurando por algo específico ou navegar por categoria se quiser se inspirar. Algumas empresas que publicaram ótimos exemplos de APIs na Rede de APIs Públicas incluem:

- **Salesforce**: O Salesforce é um software de CRM baseado em nuvem que permite que os clientes encontrem e se conectem com clientes, fechem negócios e forneçam serviços em escala. O espaço de trabalho público do Salesforce inclui muitos recursos de API para desenvolvedores em nuvens e produtos do Salesforce.
- **Notion**: O Notion é um aplicativo de anotações que fornece um único espaço para pensar, escrever e planejar. A equipe do Notion publicou uma coleção simples para ajudar os desenvolvedores a começar a interagir com a API do Notion. Eles fornecem soluções atualizadas e práticas, facilitando a conexão e a automação de suas tarefas no Notion.

- **Discord**: O Discord é uma plataforma social que facilita a comunicação baseada em voz, vídeo e mensagens, bem como o compartilhamento de mídia e arquivos. A equipe do Discord publicou coleções que ajudam os desenvolvedores a criar aplicativos do Discord para personalizar seus servidores do Discord. Eles também incluíram uma introdução útil para ajudá-lo a começar.
- **Pinterest**: O Pinterest permite que os usuários compartilhem imagens, se conectem uns com os outros e se inspirem. A API do Pinterest aumenta significativamente o escopo do que você pode fazer como desenvolvedor de sites ou aplicativos, e eles publicaram coleções para ajudá-lo a começar.

BONS ESTUDOS

