



Tópicos > APIs > O que é uma API?

O que é API?

Updated January 19, 2023 • 12-minute read

[Copy URL](#)

O que significa API?

API é a sigla em inglês para *Application Programming Interface*, ou interface de programação de aplicações. As interfaces de programação de aplicativos (APIs) são conjuntos de ferramentas, definições e protocolos para a criação de aplicações de software. APIs conectam soluções e serviços, sem a necessidade de saber como esses elementos foram implementados.

[Manual do proprietário de API \(em inglês\)](#)

Como as APIs funcionam?

Com as APIs, sua solução ou serviço podem se comunicar com outros produtos e serviços sem precisar de conhecimento prévio sobre como eles foram implementados. Isso simplifica o desenvolvimento de aplicações,

Olá! Como posso te ajudar?

1

gerando economia de tempo e dinheiro. Na hora de desenvolver suas novas ferramentas e soluções (ou ao gerenciar aquelas já existentes), as APIs oferecem a flexibilidade necessária para simplificar o design, a administração e o uso, além de trazer oportunidades de inovação.

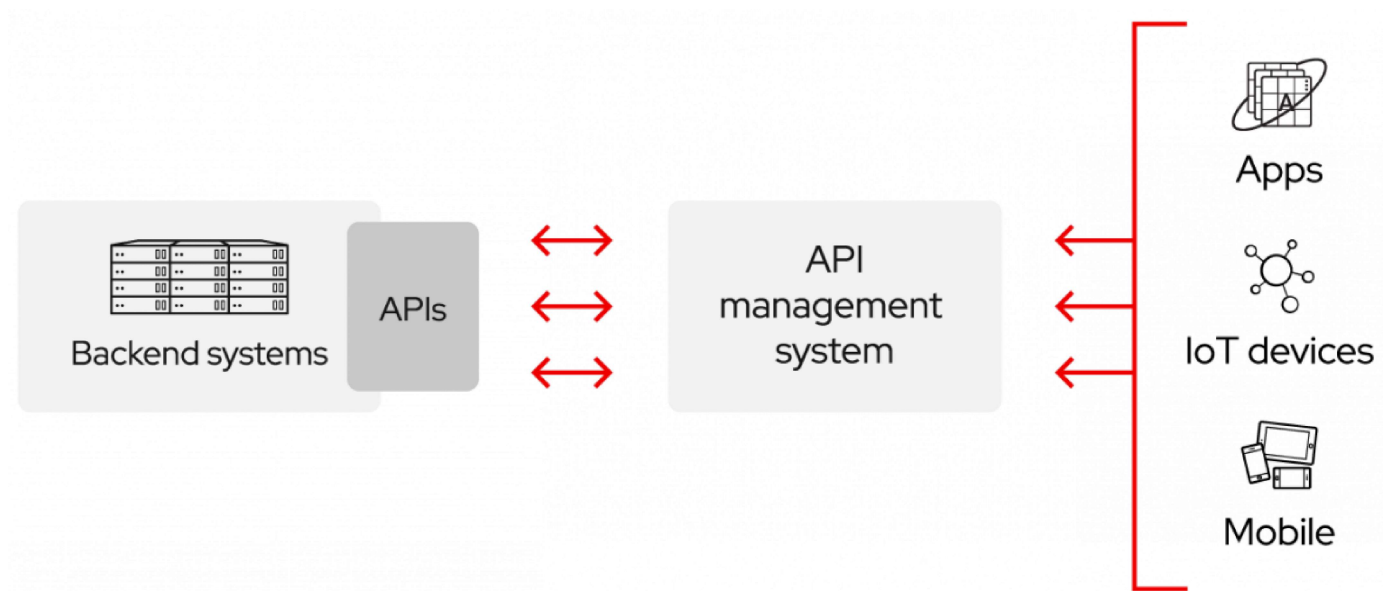
As APIs funcionam como se fossem contratos, com documentações que representam um acordo entre as partes interessadas. Se uma dessas partes enviar uma solicitação remota estruturada de uma forma específica, isso determinará como a aplicação da outra parte responderá.

Vantagens de usar APIs

As APIs simplificam a forma como os desenvolvedores integram novos componentes de aplicações a uma arquitetura preexistente. Por isso, elas ajudam na colaboração entre as empresas e as equipes de TI. Muitas vezes, as necessidades empresariais mudam rapidamente para responder aos mercados digitais em transformação. Nesse ambiente, novos concorrentes podem redefinir o setor inteiro com uma nova aplicação. Para manter a competitividade, é importante oferecer suporte à implantação e desenvolvimento rápidos de serviços inovadores. O desenvolvimento de aplicações nativas em nuvem é uma forma conhecida de aumentar a velocidade de criação. Ele depende de uma arquitetura de aplicações de microsserviços conectada por meio de APIs.

As APIs são uma maneira simplificada de conectar a própria infraestrutura por meio do desenvolvimento de aplicações nativas em nuvem. No entanto, elas também possibilitam o

compartilhamento de dados com clientes e outros usuários externos. As APIs públicas agregam valor de negócios porque simplificam e ampliam como você se conecta aos parceiros, além de, possivelmente, monetizar seus dados. Um exemplo famoso é a API do Google Maps.



Por exemplo, imagine uma empresa distribuidora de livros. Essa distribuidora de livros poderia oferecer aos clientes uma aplicação em nuvem onde os atendentes de uma livraria verificassem a disponibilidade de um título diretamente com a distribuidora. Essa aplicação poderia ser projeto caro, limitado pela plataforma e que exigiria longos períodos de desenvolvimento e manutenção contínua.

Como alternativa, a distribuidora, poderia fornecer uma API para verificar a disponibilidade no estoque. Essa abordagem proporciona vários benefícios, incluindo:

- O acesso aos dados por meio de uma API ajuda os clientes a consolidarem informações sobre seu inventário em um único local.

- A distribuidora de livros pode fazer alterações nos sistemas internos sem causar impacto nos clientes, contanto que o comportamento da API não mude.
- Com uma API disponibilizada publicamente, os desenvolvedores que trabalham para a distribuidora de livros, os vendedores ou terceiros poderiam desenvolver uma aplicação para ajudar os clientes a encontrar os livros que procuram. Isso poderia resultar no aumento das vendas ou outras oportunidades de negócios.

Para resumir, com as APIs, você libera o acesso aos seus recursos sem abdicar da segurança e do controle. É você quem determina como isso será feito e quem terá acesso. A segurança das APIs depende de um bom gerenciamento, que inclui o uso de um gateway de APIs. É possível conectar APIs e criar aplicações que usam os dados ou funcionalidades disponibilizadas por elas usando uma plataforma de integração distribuída que ligue todos os elementos, incluindo sistemas legados e dispositivos de Internet das Coisas (IoT).

E-book: Integração ágil - referência para a arquitetura empresarial →

Tipos de API

Privada

A API é usada apenas internamente. Isso oferece às empresas um maior controle.

Parceiros

A API é compartilhada com parceiros de negócios específicos. Isso pode fornecer fluxos de receita adicionais sem comprometer a qualidade.

Pública

A API é disponibilizada para todos. Terceiros podem desenvolver aplicações que interajam com a sua API e isso pode se tornar uma fonte de inovação.

Global Tech Outlook 2023: um relatório da Red Hat

APIs e inovação

A exposição das suas APIs aos parceiros ou ao público pode:

- Criar novos canais de receita ou ampliar os existentes.
- Expandir o alcance da sua marca.
- Facilitar a inovação aberta ou aumentar a eficiência por meio da colaboração e de desenvolvimento externos.

Parece bom, não é? Mas como as APIs podem fazer tudo isso?

Vamos voltar para o exemplo da empresa distribuidora livros.

Suponhamos que um dos parceiros da empresa desenvolva uma aplicação para ajudar as pessoas a encontrar livros nas prateleiras de livrarias. Essa experiência aprimorada atrai mais consumidores para livraria (que é cliente da distribuidora) e amplia o canal de receita existente.

Talvez um terceiro use uma API pública para desenvolver uma aplicação que permita que as pessoas comprem livros diretamente da distribuidora em vez de em uma livraria. Isso abre um novo canal de receita para a distribuidora de livros.

O compartilhamento de APIs (com parceiros selecionados ou aberta a todos) pode ter efeitos positivos. Cada parceria amplia o reconhecimento da sua marca para além dos esforços de marketing da sua empresa. Abrir a tecnologia para todos, como fazem as APIs públicas, estimula os desenvolvedores a criar um ecossistema de aplicações com base na sua API. Quanto mais pessoas usam sua tecnologia, maior é a sua chance de gerar novos negócios.

Tornar a tecnologia pública pode gerar resultados novos e inesperados. Às vezes, esses resultados revolucionam setores inteiros. No caso da nossa empresa distribuidora de livros, novas organizações (um serviço de empréstimos de livros, por exemplo) podem mudar fundamentalmente a maneira dela de fazer negócios. Com as APIs públicas e em parceria, você pode se beneficiar dos esforços criativos de uma comunidade muito maior do que a sua equipe interna de desenvolvedores. Novas ideias podem surgir de qualquer lugar. Por disso, as empresas precisam estar cientes das mudanças no mercado e prontas para agir. As APIs podem ajudar.

Infográfico: Oito etapas para desenvolver apps nativas em nuvem →

História das APIs

As APIs surgiram nos primeiros dias da computação, muito antes do computador pessoal. Naquela época, elas eram normalmente usadas como bibliotecas para sistemas operacionais. Embora a API enviasse mensagens entre mainframes em alguns momentos, ela era quase sempre local para os sistemas em que operava. Depois de quase 30 anos, as APIs se expandiram para além dos ambientes locais. No início dos anos 2000, elas estavam se tornando uma tecnologia importante para a integração remota de dados.

APIs remotas ou APIs Web

As APIs remotas foram projetadas para interagir por meio de uma rede de comunicações. Quando falamos *remota*, queremos dizer que os recursos utilizados pela API estão em algum lugar fora do computador que realiza a solicitação. Como a rede de comunicações mais usada é a Internet, a maioria das APIs são projetadas com base em padrões da web. Nem todas as APIs remotas são web, mas é justo afirmar que, em geral, as APIs web são remotas.

As APIs web normalmente usam o protocolo HTTP para mensagens de solicitação e fornecem uma definição da estrutura das mensagens de resposta. Essas mensagens de resposta geralmente têm o formato de arquivo XML ou JSON. Tanto XML quanto JSON são formatos de preferência porque apresentam os dados de forma simplificada, facilitando a manipulação por outras aplicações.

API SOAP e API REST

Com a proliferação das APIs web, uma especificação de protocolo foi desenvolvida para ajudar a padronizar a troca de informações: o Simple Object Access Protocol, mais conhecido como SOAP. As APIs projetadas com SOAP usam o XML como formato de mensagem e recebem solicitações por HTTP ou SMTP. O SOAP facilita o compartilhamento de informações por aplicações executadas em ambientes diferentes ou escritos em linguagens diferentes.

Outra especificação é a Representational State Transfer (REST). APIs web que adotam as restrições de arquitetura da REST são chamadas de APIs RESTful. A REST é fundamentalmente diferente

do SOAP: o SOAP é um protocolo e a REST é um estilo de arquitetura. Isso significa que não há um padrão oficial para APIs RESTful web. Conforme definido na dissertação de Roy Fielding "Architectural Styles and the Design of Network-based Software Architectures", as APIs serão consideradas RESTful se estiverem em conformidade com seis restrições de arquitetura:

- **Arquitetura cliente-servidor:** a arquitetura REST é composta por clientes, servidores e recursos. Ela lida com as solicitações via HTTP.
- **Sem monitoração de estado:** nenhum conteúdo do cliente é armazenado no servidor entre as solicitações. Em vez disso, as informações sobre o estado da sessão são mantidas com o cliente.
- **Capacidade de cache:** o armazenamento em cache pode eliminar a necessidade de algumas interações entre o cliente e o servidor.
- **Sistema em camadas:** as interações entre cliente e servidor podem ser mediadas por camadas adicionais. Essas camadas podem oferecer recursos extras, como balanceamento de carga, caches compartilhados ou segurança.
- **Código sob demanda (opcional):** os servidores podem ampliar a funcionalidade de um cliente por meio da transferência de códigos executáveis.
- **Interface uniforme:** essa restrição é essencial para o design de APIs RESTful e inclui quatro vertentes:
 - **Identificação de recursos nas solicitações:** os recursos são identificados nas solicitações e separados das representações retornadas para o cliente.

- **Manipulação de recursos por meio de representações:** os clientes recebem arquivos que representam recursos. Essas representações precisam ter informações suficientes para permitir a modificação ou exclusão.
- **Mensagens autodescritivas:** cada mensagem retornada para um cliente contém informações suficientes para descrever como ele deve processá-las.
- **Hipermídia como plataforma do estado das aplicações:** depois de acessar um recurso, o cliente REST pode descobrir todas as outras ações disponíveis no momento por meio de hiperlinks.

Essas restrições podem parecer excessivas, mas são muito mais simples do que um protocolo prescrito. Por isso, as APIs RESTful estão se tornando mais comuns do que as APIs SOAP.

Nos últimos anos, as especificações da OpenAPI se tornaram o padrão na hora de definir APIs REST. A OpenAPI permite que desenvolvedores de todas as linguagens criem interfaces de API REST compreensíveis com o mínimo de suposições.

GraphQL

Outro padrão de API emergente é o GraphQL, uma linguagem de consulta e ambiente de execução voltado a servidores alternativa ao REST. A prioridade do GraphQL é fornecer exatamente os dados que os clientes solicitam e nada além. Como alternativa à arquitetura REST, o GraphQL permite aos desenvolvedores construir solicitações que extraem os dados de várias fontes em uma única chamada de API.

O que é arquitetura orientada a serviços (SOA)?

As duas abordagens de arquitetura que mais usam APIs remotas são a arquitetura orientada a serviços (SOA) e a arquitetura de microsserviços. A SOA, a mais antiga das duas abordagens, começou como um aprimoramento das aplicações monolíticas. Considerando que um único aplicativo monolítico faz tudo, algumas funções podem ser fornecidas por aplicações diferentes levemente acopladas por meio de um padrão de integração, como um barramento de serviços corporativos (ESB).

Embora a SOA seja, de forma geral, mais simples que uma arquitetura monolítica, ela traz o risco de mudanças em cascata por todo o ambiente caso as interações entre os componentes não sejam compreendidas claramente. Essa complexidade adicional traz de volta alguns dos problemas que a SOA foi criada para solucionar.

As arquiteturas de microsserviços são parecidas com os padrões SOA no que diz respeito à utilização de serviços levemente acoplados. No entanto, elas vão além no rompimento com as arquiteturas tradicionais. Na arquitetura de microsserviços, os serviços usam um framework de mensageria comum, como as APIs RESTful. As APIs RESTful são usadas na comunicação entre serviços sem transações de conversão de dados complicadas ou camadas de integração adicionais. O uso de APIs RESTful permite – e até estimula – a entrega mais rápida de novos recursos e atualizações. Cada serviço é independente. Um serviço pode ser substituído, aprimorado ou descartado sem afetar nenhum outro na arquitetura.

Essa arquitetura leve ajuda a otimizar os serviços distribuídos ou de nuvem e oferece suporte à escalabilidade dinâmica para serviços individuais.

O que é arquitetura orientada a serviços (SOA)? →

APIs e webhooks

Um webhook é uma função de retorno de chamada baseada em HTTP que viabiliza uma comunicação lightweight e orientada por eventos entre duas APIs. Os webhooks são usados por várias aplicações web para receber pequenos volumes de dados de outras aplicações, mas também podem ser utilizados para acionar fluxos de trabalho de automação nos ambientes do GitOps.

Em geral, os webhooks são descritos como APIs reversas ou APIs de push, porque colocam a responsabilidade da comunicação no servidor, e não no cliente. Em vez de o cliente enviar as solicitações HTTP pedindo os dados até receber uma resposta, o servidor envia para o cliente uma solicitação HTTP POST exclusiva quando os dados estiverem disponíveis. Apesar dos apelidos, os webhooks não são APIs. Eles trabalham juntos. Uma aplicação precisa ter uma API para usar um webhook.

O que é um webhook? →

Leitura recomendada



ARTIGO

O que é uma API?

API significa interface de programação de aplicações, um conjunto de definições e protocolos para criar e integrar softwares de aplicações.

Leia mais →

ARTIGO

Qual é a função de um gateway de API?

O gateway de API é uma ferramenta de gerenciamento de interfaces de programação de aplicações (APIs) que fica entre o cliente e uma coleção de serviços de back-end.

Leia mais →

ARTIGO

Por que escolher a Red Hat para o uso de APIs?

Nossas soluções de API se concentram na capacidade de reutilização, agilidade da TI e interface de gerenciamento para você avaliar, monitorar e escalar.

Leia mais →

Leia mais sobre APIs

Soluções

Artigos relacionados

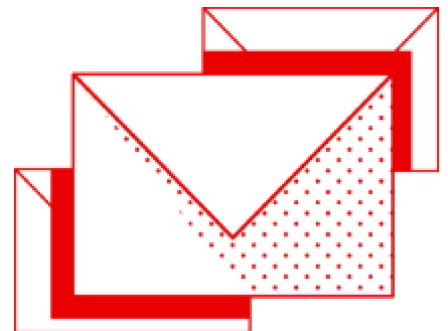
Recursos



Red Hat
3scale API
Management

Uma plataforma de infraestrutura que permite compartilhar, distribuir, controlar e monetizar APIs (interfaces de programação de aplicações).

Mais informações [→](#)



Quer receber mais conteúdo deste tipo?

Cadastre-se para receber a nossa newsletter Red Hat Shares.

Continuar



Produtos

Ferramentas

Experimente, compre, venda

Comunicação

Sobre a Red Hat

A Red Hat é a líder mundial em soluções empresariais open source como Linux, nuvem, containers e Kubernetes. Fornecemos soluções robustas que facilitam o trabalho em diversas plataformas e ambientes, do datacenter principal até a borda da rede.

Assine nossa newsletter Red Hat Shares

Cadastre-se agora →

Selecione um idioma

 Português ▼



Sobre a Red Hat

Oportunidades de emprego

Eventos

Escritórios

Fale com a Red Hat

Blog da Red Hat

[Diversidade, equidade e inclusão](#)

[Cool Stuff Store](#)

[Red Hat Summit](#)

© 2023 Red Hat, Inc.

[Declaração de privacidade](#)

[Termos de uso](#)

[Todas as políticas e diretrizes](#)

[Acessibilidade digital](#)

[Preferências de cookies](#)