

API

- APIs são uma ponte. Elas conectam 2 softwares. Por exemplo, o departamento de meteorologia e o que mostra o clima no seu telefone. Entre eles existe uma ponte, isso são as APIs.

Dep. de Meteorologia <----- API -----> Celular

- API significa Application Programming Interface, e essa interface é construída a partir de comunicações entre os aplicativos por meio de normas e diretrizes. Ou seja, você manda para ele o que você quer, a comunicação acontece através da API, que define as normas e regras para a troca de mensagens. O cliente monta uma requisição seguindo essas normas, e o servidor a entende porque foi programado para seguir as mesmas normas. O aplicativo traduz e manda de volta para você o que você requisitou. A API faz essa comunicação a partir de normas. Como você pode pensar, alguns sites utilizam dos correios para calcular fretes para você. Eles fazem isso por meio de APIs.

- Como elas funcionam? . Baseamos ela em Cliente e Servidor, sendo o cliente quem solicita e quem envia a resposta o servidor. . Logo, o cliente faz uma requisição, o servidor processa ela, checando se está tudo dentro dos conformes, e, depois, manda de volta a resposta com a requisição aceita ou não. Assim que a API opera.

- Existem 4 tipos de API: . API SOAP: Simple Object Access Protocol, é uma API que usa XML e o servidor e o cliente trocam mensagens por HTTP. O XML serve como ponte.

- API RPC: Remote Procedure Calls, chama um procedimento, uma função, que executa no servidor e checa os parâmetros. Depois disso, ele retorna a resposta da função.

- API WebSocket: É uma API que invoca um canal, como objetos JSON, para transmitir dados. Comunicação bidirecional com o servidor e o cliente. Diferente das duas anteriores, que funcionam como "cartas", esse funciona em tempo real. Ele vai no backend, invoca a mensagem do cliente e manda ela para o servidor. Funciona com o Handshake, processo que o cliente fala para o servidor que ele quer sair do HTTP e ir para o WS. Depois disso, a conexão é estabelecida, mudando de HTTP para ws se o servidor concordar. Agora, uma conexão TCP (Transmission Control Protocol) é mantida com baixa latência.

- API REST: Representational State Transfer, usa os métodos de HTTP (GET, POST, PUT, DELETE). Através do CRUD (Create, Read, Update, Delete), ele executa esses protocolos HTTP por meio de JSON. Nesse sentido, os dados tem endereços específicos no site, o usuário e seus dados também, e para interagir com eles, você usa os métodos, os verbos do HTTP. Os pilares dessa API são Arquitetura Cliente-Servidor, onde o cliente e o servidor são independentes; o Stateless, que o servidor não salva informações sobre o cliente, toda vez que o cliente precisa interagir, ele precisa apresentar todas as informações necessárias; a Interface Uniforme, composta pela Identificação de Recursos Uniformes (URIs), Manipulação Através de Representações, ou seja, acessar o banco de dados por meio de uma cópia dele, como o JSON, e não o banco de fato e Mensagens Autodescritivas que basicamente é a característica de todos os dados estarem organizados e serem, por si só, suficientes para entender a comunicação.

REST API Design

Best Practices



Core Principles of API Design

URL Versioning Example

URL	Version
	Version 1
	Version 2

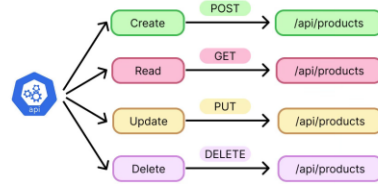
Query Parameter Versioning Example

URL	Version
	Version 1
	Version 2

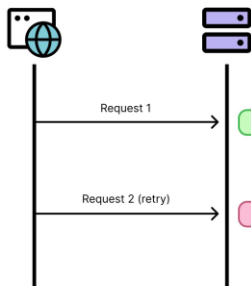
LEVEL 200
200: OK
201: Created
202: Accepted
204: No Content

LEVEL 400
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
409: Conflict

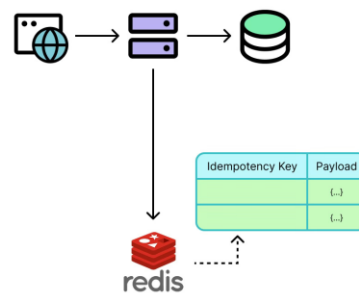
LEVEL 500
500: Server Error
502: Bad Gateway
503: Service Unavailable
504: Gateway Timeout



Idempotency in API Design

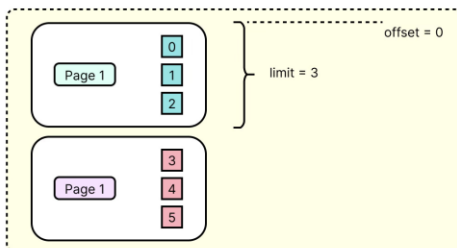


HTTP Method	Idempotence
GET	YES
HEAD	YES
DELETE	YES
POST	NO
PATCH	NO

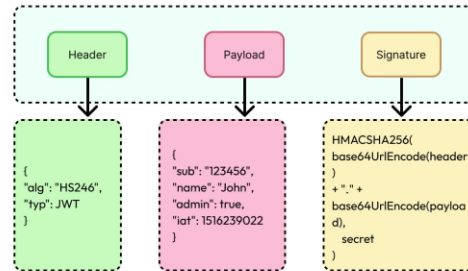


Pagination

GET /orders?limit=3&offset=0



API Security



JWT Structure