

DDM-II

API, API REST, API RESTFUL

Prof. Anderson Vanin

O QUE É UMA API?

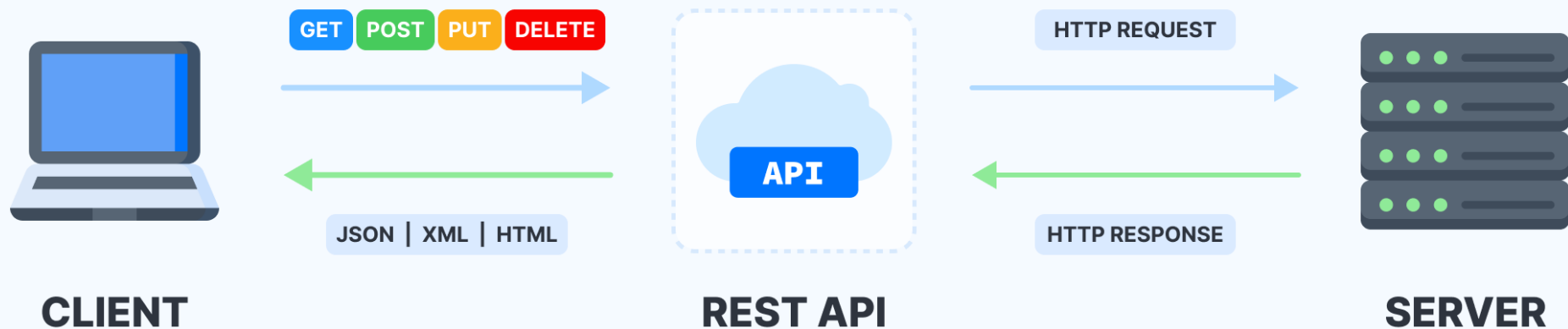
Na programação e desenvolvimento de softwares, existe uma série de conceitos fundamentais para garantir uma boa experiência ao usuário – que sequer sonha com isso.

Um deles é a **API**. Um código programável que consegue garantir que **dois softwares diferentes possam se comunicar entre si**.

E ainda a **REST API**, outro conceito que sua empresa pode utilizar para garantir um bom desenvolvimento de **serviços de web**.

O QUE É UMA API?

REST API Model



O QUE É UMA API?

Basicamente, ao consumir dados através de uma API, uma aplicação precisa fornecer unicamente detalhes sobre as informações que deseja imputar ou consultar através de uma determinada **URL (rota)**, e toda a mágica será feita pela aplicação de forma transparente.

O QUE É UMA REST API?

O nome completo de nosso personagem é **RESTFul API**.

Enquanto **API** é a sigla em inglês para Interface de Programação de Aplicação (*Application Programing Interface*), o **REST** é acrônimo para *Representation State Transfer*.

Bom, mas decifrar os nomes não ajuda muito. Na prática, que é o que vale mesmo, o **REST é uma API que usa restrições HTTP para extrair, inserir, postar e deletar dados.**

Com a REST API, o usuário *se conecta e interage com aplicações em nuvens*, o que é muito usado em sites como Google, Amazon, LinkedIn e Twitter.

Como toda API, ela conecta dados armazenados em uma plataforma e “transfere” para outra.

QUAIS SÃO OS TIPOS DE APIS?

A REST não é a única API que existe.

As APIs têm três formas de atuar: **localmente, baseada na web e em programas**. Veja:

- As de **atuação local são chamadas de privadas**, utilizadas internamente entre as aplicações de uma empresa.
- Atuando **baseada na web são as públicas**, usadas livremente e disponibilizadas por empresas para que os desenvolvedores façam integrações entre as aplicações.
- Já as **parceiras são aquelas usadas entre negócios** que comungam a integração dos dados, mesmo que sejam softwares diferentes.

QUAIS SÃO AS FUNCIONALIDADES DE UMA REST API?

A principal funcionalidade de uma REST API é ser uma “ponte” entre dois softwares diferentes em que um cede dados do usuário a outro software.

Um exemplo bem prático dessa funcionalidade é fazer login em um sistema utilizando dados já cadastrados em outro.

Sabe quando você acessa um sistema de pagamento online e faz login pelas redes sociais, como o Facebook?

Um REST API é a interface dessa transferência de dados. Isso ajuda a separar as aplicações back-end e front-end.

QUAIS SÃO AS FUNCIONALIDADES DE UMA REST API?

Isso significa dizer que os dados do usuário são preservados e a troca dessas informações são realizadas apenas para recuperar dados, inserir ou deletá-los.

Ter uma REST API em sua empresa ainda permite utilizar diversas funcionalidades em seu site.

A principal delas, como não poderia deixar de ser em um sistema de troca de informações, é a integração com diferentes plataformas, como as redes sociais e os sistemas de pagamento.

Isso é capaz de te dar, ainda, mais escalabilidade e ganhar confiança de seus usuários.

QUAIS SÃO OS MÉTODOS USADOS EM UMA REST API?

Para fazer tudo isso, uma REST API se utiliza de alguns meios. Esses métodos de recursos são verbos, em inglês, que dizem quais as ações que o API deve realizar, de acordo com o pedido do usuário.

- **GET**: esse recurso é quando um usuário requisita dados para a API, que, por sua vez, busca essas informações na nuvem, disponibilizadas por outras plataformas, como as redes sociais.
- **POST**: essa ação acontece quando o usuário permite o envio de dados para serem processados em outros recursos. É como preencher um formulário web que, posteriormente, poderá servir para abastecer um requerimento.
- **PUT**: já esse comando é utilizado para atualizar dados.
- **DELETE**: como não poderia deixar de ser, um recurso que é usado para apagar dados.

PARA QUE USAR RESTS APIS?

Veja algumas vantagens que sua empresa terá ao usar esse recurso.:

- Dados de clientes e servidor separados

Ainda há pouco, você leu sobre essa separação de dados entre cliente e servidor.

Mas, é importante ressaltar que, dessa forma, você protege todo o armazenamento dessas informações, já que toda troca de comunicação entre a **REST API é para troca de informações, pois não há tratamento de regras de negócio.**

- Rapidez

De modo geral, toda **API facilita a comunicação entre dois softwares.**

Isso faz com que o cadastro de um usuário faça essa ação de maneira muito mais rápida. Basta pegar o exemplo do login em uma plataforma de pagamentos clicando em um botão com “entrar com Facebook”.

Melhorando a experiência do usuário em seu app ou site, ele certamente te dará mais autoridade, gerando escalabilidade em seu negócio.

A API é uma maneira de automatizar processos em seu site, fazendo com que o usuário ganhe tempo e tenha uma experiência melhor em sua plataforma.

RACHEL

BARISTA

KITCHEN

METHOD

PARAMETERS

REQUEST

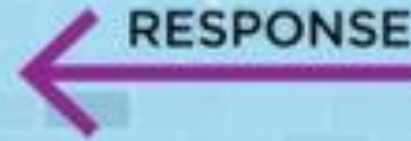
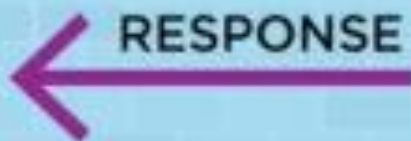
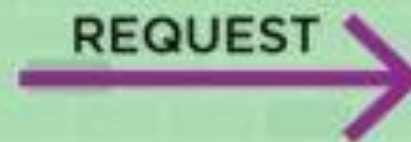
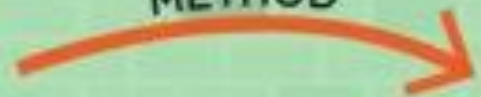
RESPONSE

RESPONSE

USER

API

APP



EXEMPLO REPRESENTACIONAL - REQUEST

Um cliente vai à um restaurante para comprar uma sopa. Ao chegar no balcão de atendimento, ele vai escolher seu pedido e fazer o pagamento por ele. Assim, na nossa API temos o POST, imagine que a rota é o balcão de atendimento e as regras de transação é cardápio onde nosso cliente vai escolher que tipo de sopa ele quer, por fim pagamento é o estado representacional que ele vai fornecer (nesse caso dinheiro). Em programação o pagamento seria um estado representacional de algo que queremos consumir no caso da nossa API o estado que vamos usar serão informações em formato JSON. Na estrutura de um restaurante por exemplo, o dinheiro vai se transformar em sopa quando o dono do restaurante compra com o dinheiro os ingredientes da sopa e com o mesmo dinheiro paga o salário do cozinheiro da sopa e do garçom que irá servi-la ao cliente.

EXEMPLO REPRESENTACIONAL - REQUEST

Abaixo um estado representacional de request (cliente -> servidor):

POST: `http://restaurante/cardapio/sopadecebola`

Estado representacional do pedido (JSON, mas poderia ser XML ou outro)

```
{  
  produto: sopa,  
  pagamento: valor_da_sopa  
}
```

EXEMPLO REPRESENTACIONAL - RESPONSE

A segunda fase da negociação é o response, que nada mais é do que a resposta à nossa requisição. Ainda no nosso restaurante, podemos imaginar os tipos de resposta.

Imagine que o cliente fez um pagamento no cartão e não tinha saldo suficiente. Logo receberemos um erro 4XX onde o cliente não forneceu os dados corretamente e assim não pode comprar sua sopa pois as informações fornecidas no caso o pagamento, não foram condizentes com o esperado pelo nosso restaurante (API). No caso da nossa API teremos um pagina não encontrada 404 ou do restaurante a sopa não poderá ser vendida pois o cliente não pode pagar por ela.

Ainda podemos ter os casos onde o cliente conseguiu pagar, mas quando chegou enviou o pedido para a cozinha o garçom foi informado de que não tinham os ingredientes para a sopa e então tivemos um 500 (erro interno) e o restaurante não pode fazer a venda da sopa.

EXEMPLO REPRESENTACIONAL - RESPONSE

Por fim, temos o caso de sucesso onde o cliente escolheu a sopa o garçon informou que estava a caminho (200 OK) e então a sopa será entregue ao cliente que receberá assim o estado representacional que poderá ser consumido a SOPA. No caso do nosso sistema.

```
Estado representacional do produto sendo servido (JSON mas poderia ser XML ou outro)
{
  produto: "sopa de cebola quentinha"
}
```