EDUCAÇÃO, TREINAMENTOS, INOVAÇÃO E CURSOS

O HUB School é tudo isso!

Porque não existe inovação e empreendedorismo sem pessoas capacitadas.

Aristóteles Esteves Marçal da Silva

Applications Program Interfaces

- Apis servem como interface para comunicar diferentes partes de software
- Como assim?
  - Você escolhe algo no cardápio sem entender como efetivamente será entregue sua comida.

Como era antes?

- Monolíticos x Aplicações Distribuídas
  - Você tinha que importar um conjunto de bibliotecas para a o seu projeto
  - Muito código replicado
  - Muitos pontos de manutenção

Micro Serviços

- Várias aplicações menores e independentes
- Melhora a escalabilidade
- Facilidade na integração e execução de testes
- Propõe que sejam completamente independentes
- Exemplos: gtalk, gmail e contatos do gmail

SOAP

- SOAP - Simple Object Access Process
- Era a recomendação da W3C em 2003
- Deriva da Xlm-RPC
- Utiliza xml para enviar dados

API Rest

- Representational State Transfer
- Foi criado junto com o HTTP 1.1 com o objetivo de prover acesso a qualquer sistema
- REST, não é uma tecnologia, nem uma biblioteca mas sim um modelo a ser utilizado para se projetar arquiteturas de software distribuído, baseadas em comunicação via rede.
- Um servidor pode se comportar como um cliente
- Padronização da comunicação

API RestFul

- Ter uma arquitetura cliente/servidor formada por clientes, servidores e recursos, com solicitações gerenciadas por meio de HTTP.
- Realizar comunicação cliente/servidor stateless. Isso significa que cada solicitação é separada e não conectada com outras, e que nenhuma informação do cliente é armazenada entre elas.
- Armazenar dados em cache para otimizar as interações entre cliente e servidor.

Métodos HTTP

- GET - Lê a a representação de um recurso
- POST - Cria um recurso
- PUT - Atualiza um recurso de forma completa
- PATCH - Modifica um recurso parcialmente, não precisa enviar Json inteiro
- DELETE - Deleta um recurso

HTTP Status Code

- 200 OK - Requisição bem sucedida
- 201 Created - Foi criado um novo recurso
- 2xx - Sucesso
- 3xx - Redirection
- 4xx - Client Error
- 5xx - Server error

EndPoints

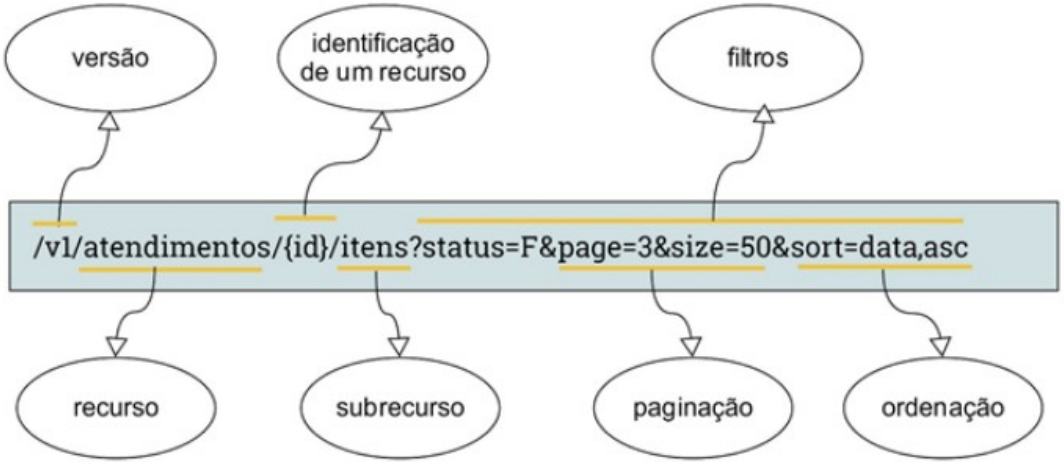
- Link que a partir de uma entrada específica retorna valores para sua consulta
- Lembrem de Interface = contrato

EndPoints

- A grande vantagem em REST é que se está aproveitando métodos HTTP existentes para implementar funcionalidades significativas em apenas um único endpoint /tickets.
  - GET /tickets Retorna a lista de bilhetes
  - GET /tickets/12 Retorna um bilhete específico
  - POST /tickets Cria um novo bilhete
  - PUT /tickets/12 Atualiza o bilhete #12
  - PATCH /tickets/12 Atualiza parcialmente o bilhete #12
  - DELETE /tickets/12 Apaga o bilhete #12

Como é esta url?

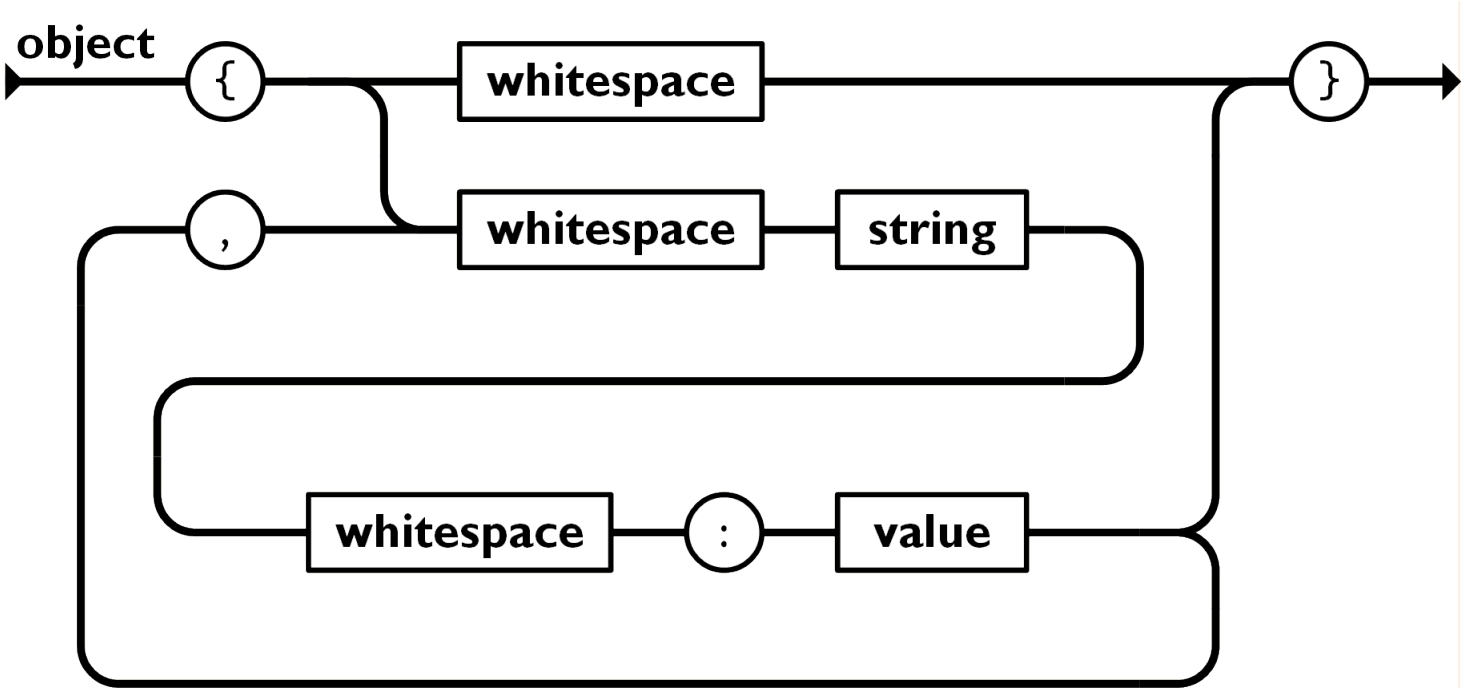
- Resource (recurso): é um objeto ou a representação de alguma coisa, que tem alguns dados associados a ele e podendo ter um conjunto de operações (criar, apagar, alterar).
  - Exemplo: alunos, funcionários, produtos são recursos;



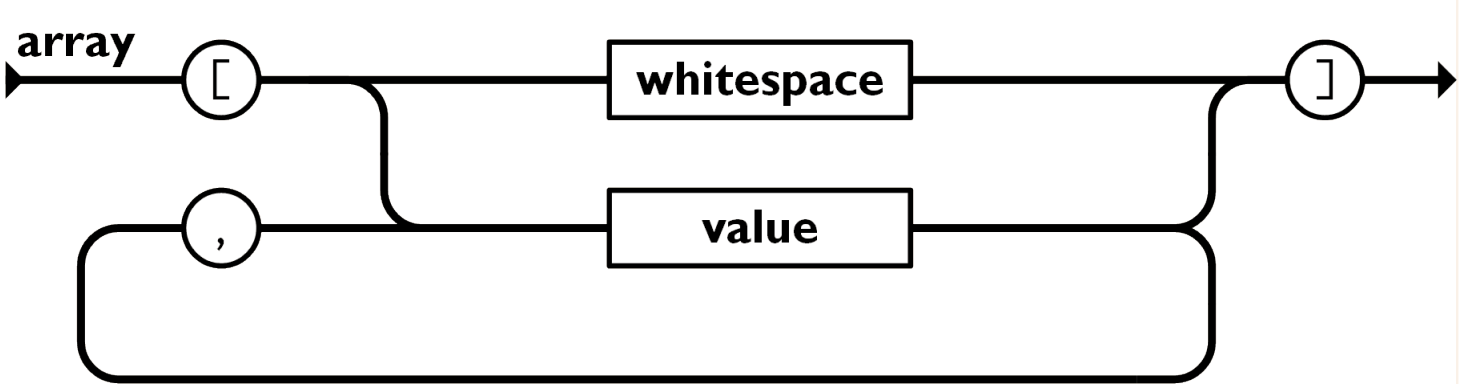
JSON

- JSON (JavaScript Object Notation) é um formato de intercâmbio de dados leve.
- É fácil para humanos ler e escrever.
- É fácil para as máquinas analisar e gerar.
- JSON é baseado em uma coleção de pares de chave / valor

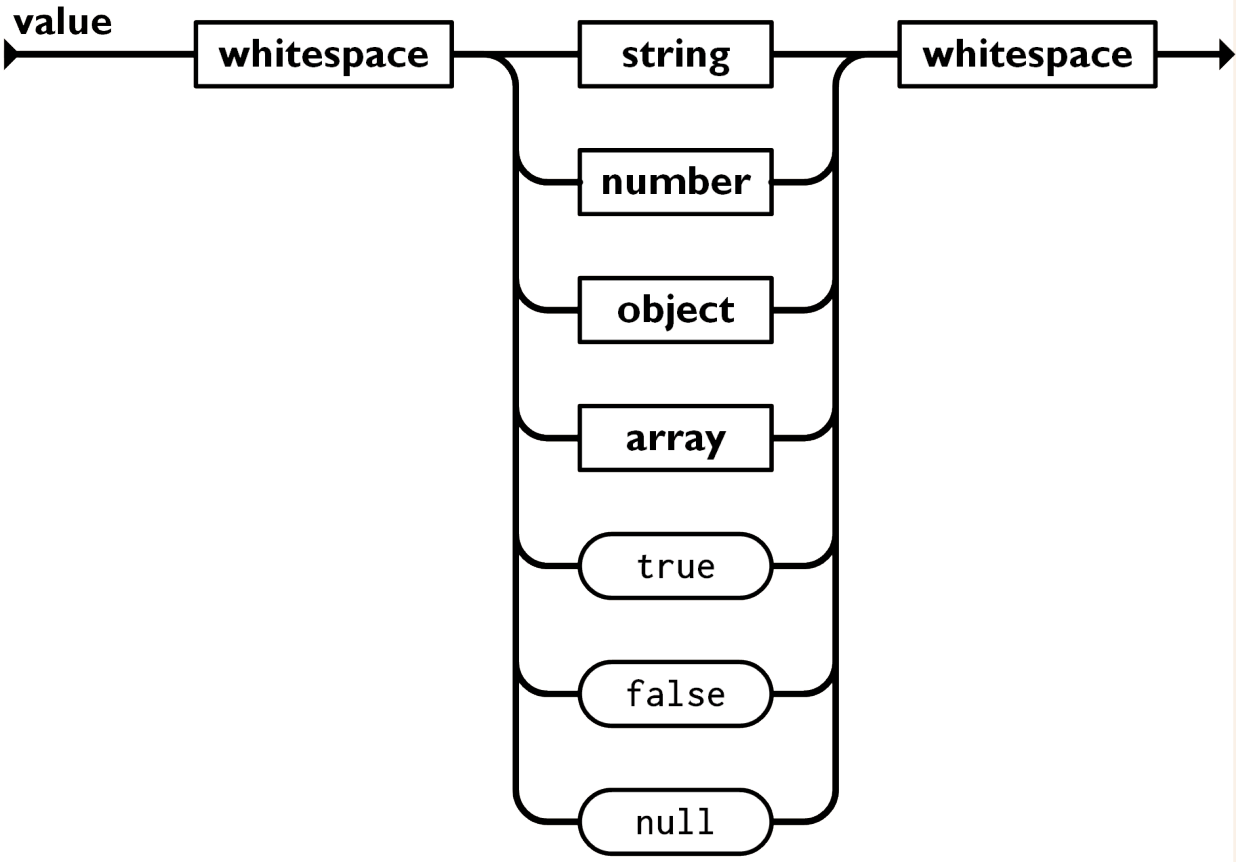
Objeto



Array



Valor


value
whitespace
string
number
object
array
true
false
null
whitespace

Exemplo

- https://randomuser.me/api/
- https://randomuser.me/api/?nat=br&results=20

EndPoints

- Link que a partir de uma entrada específica retorna valores para sua consulta
- Lembrem de Interface = contrato

JSON Server

- Ter uma API pronta para começar o desenvolvimento do front-end da sua aplicação as vezes pode ser um problema.
- JSON Server é uma biblioteca capaz de criar uma API Fake rápido e sem precisar escrever nenhuma linha de código.

Obrigado!