Web Services

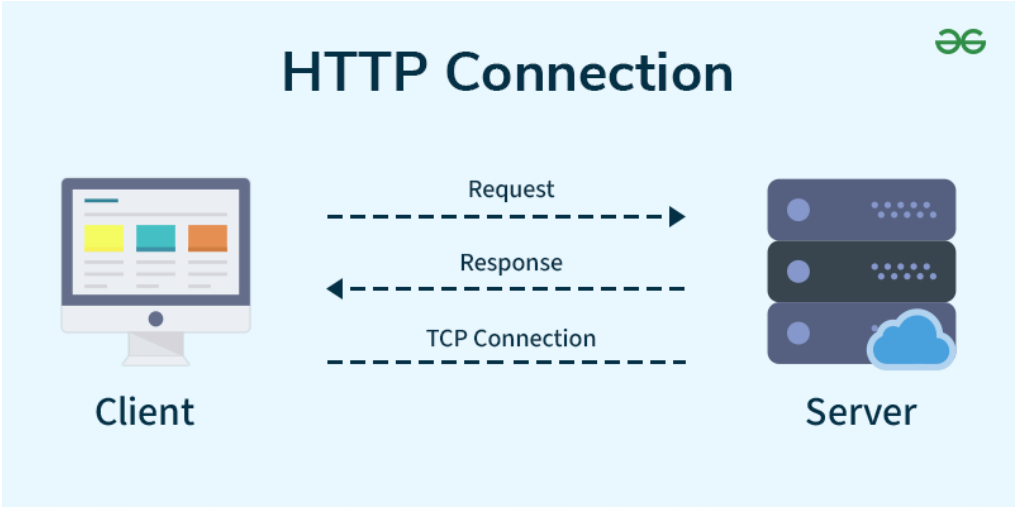
Conceitos Básicos

HTTP - Hypertext Transfer Protocol

- **HTTP - a espinha dorsal da comunicação na web.**
- O HTTP é um protocolo de comunicação que permite a transferência de dados entre clientes (como navegadores ou apps) e servidores na internet.
- Ele opera na camada de aplicação e é baseado em um modelo cliente-servidor, onde o cliente envia uma requisição e o servidor responde com os dados solicitados.

HTTP - Hypertext Transfer Protocol

- **Conceitos-chave do HTTP**
  - **Cliente-Servidor:** A comunicação HTTP se dá entre um cliente (um navegador ou um app) que envia uma requisição e um servidor que processa essa requisição e envia uma resposta.
  - **Requisição e Resposta:** O cliente envia uma requisição para o servidor, que responde com os dados solicitados, como um arquivo HTML, imagem ou vídeo.



HTTP - Hypertext Transfer Protocol

- **Conceitos-chave do HTTP**
  - **Requisição (Request):** enviada pelo cliente, contém informações como o método HTTP, URL, cabeçalhos e, às vezes, corpo de dados.
  - **Resposta (Response):** enviada pelo servidor, inclui um código de status (como 200 OK ou 404 Not Found), cabeçalhos e o corpo com os dados solicitados.
  - **URL (Uniform Resource Locator):** identifica o recurso que está sendo requisitado.
  - **Cabeçalhos (Headers):** metadados que descrevem a requisição ou resposta (como tipo de conteúdo, autenticação, etc.).
  - **Status Code:** indica o resultado da requisição (ex: 200, 404, 500).

HTTP - Hypertext Transfer Protocol

- **Métodos HTTP -** São mensagens que indicam a ação que o cliente deseja executar em um recurso no servidor.
- **GET:** Solicita / recupera dados de um recurso. Não altera o estado do servidor.
- **POST:** Envia dados para o servidor, geralmente para criar um novo recurso. Por exemplo, enviar um formulário com informações para cadastrar um novo usuário.
- **PUT:** Atualiza ou substitui um recurso já existente com novos dados.

HTTP - Hypertext Transfer Protocol

- **DELETE:** Remove um recurso específico do servidor.
- **PATCH:** Atualiza parcialmente um recurso existente. Diferente do PUT, que substitui o recurso por completo, o PATCH aplica apenas as alterações solicitadas.
- **HEAD:** É semelhante ao GET, mas solicita apenas os cabeçalhos da resposta, sem o corpo.
- **OPTIONS:** Retorna os métodos HTTP que o servidor suporta para um recurso específico.

HTTP - Hypertext Transfer Protocol

- **Códigos de status HTTP**
- Ao receber uma requisição, o servidor responde com um código de status numérico que indica o resultado da operação. Alguns exemplos comuns são:
  - **200 OK:** A requisição foi bem-sucedida.
  - **201 Created:** O recurso foi criado com sucesso.
  - **404 Not Found:** O recurso solicitado não foi encontrado.
  - **403 Forbidden:** O servidor entendeu a requisição, mas se recusa a autorizá-la.
  - **500 Internal Server Error:** O servidor encontrou uma situação inesperada que o impediu de completar a requisição.

Web Services - Conceitos Básicos

- **Web Services** ou serviços web, são tecnologias que viabilizam a comunicação entre sistemas e aplicações distintos por meio da internet.
- Permitem que softwares desenvolvidos em diferentes linguagens e plataformas — como Windows, Linux, Java, PHP, entre outros — possam interagir e compartilhar dados de forma estruturada e padronizada.
- Essa interoperabilidade é essencial para integrar soluções em ambientes heterogêneos, promovendo automação, escalabilidade e eficiência nos processos digitais.

Web Services - Conceitos Básicos

- **Como funciona**
- A comunicação via Web Service envolve três papéis principais:
  - **Provedor do serviço:** Oferece os dados ou funcionalidades que serão consumidos por outras aplicações.
  - **Consumidor do serviço:** A aplicação que solicita e utiliza os dados ou funcionalidades oferecidas pelo provedor.
  - **Registro de serviços (opcional):** Um local onde o provedor publica a descrição do seu serviço para que o consumidor possa encontrá-lo (como um catálogo).

Web Services - Conceitos Básicos

- **Funciona como uma ponte digital:**
  - Um sistema envia uma requisição (geralmente via HTTP ou HTTPS) para um servidor.
  - O servidor processa essa requisição e envia uma resposta, normalmente em formatos como XML ou JSON.
  - Tudo isso acontece de forma padronizada, garantindo que os sistemas consigam entender e processar os dados trocados.

Web Services - Conceitos Básicos

- **Principais tipos de Web Services**
- **SOAP (Simple Object Access Protocol)**
  - **Descrição**: Protocolo mais antigo e com regras rígidas. Ele usa XML para formatar as mensagens trocadas entre o cliente e o servidor.
  - **Estrutura**: Baseado em contratos formais, geralmente descritos por um arquivo WSDL (Web Services Description Language), que define todas as operações e parâmetros do serviço.
  - **Características**: Fortemente tipado, com baixa flexibilidade e maior complexidade, mas oferece maior segurança e confiabilidade para transações complexas.

Web Services - Conceitos Básicos

- **Principais tipos de Web Services**
- **REST (Representational State Transfer)**
  - **Descrição:** Trata-se de um estilo arquitetural moderno, conhecido por sua leveza e flexibilidade. Ele utiliza os métodos do protocolo HTTP — como GET, POST, PUT e DELETE — para realizar operações sobre dados.
  - **Formato de dados:** Suporta diversos formatos para troca de informações, como JSON, XML e outros. Sendo JSON, amplamente adotado.
  - **Características:** REST é especialmente indicado para aplicações web e mobile que exigem rapidez, escalabilidade e simplicidade na comunicação entre sistemas.

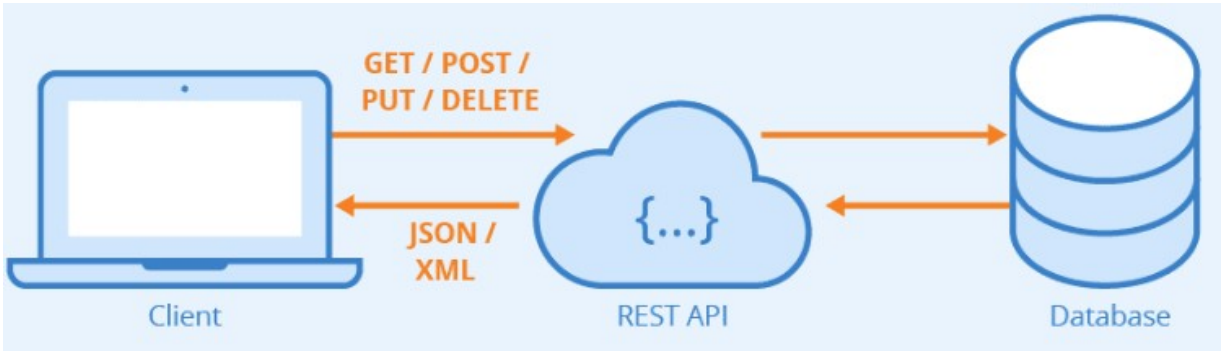
Web Services - Conceitos Básicos

- Web Services são fundamentais para a integração de sistemas, permitindo que funcionalidades sejam compartilhadas entre diferentes aplicações. Por exemplo:
  - Um app de clima acessa um Web Service para obter a previsão do tempo.
  - Um site de e-commerce consulta um Web Service para verificar o estoque ou processar pagamentos.

API - Conceitos Básicos

Interface de Programação de Aplicações

- **API Application Programming Interface**
  - É um conjunto de regras e protocolos que permite a comunicação e interação entre diferentes softwares.
  - Ela atua como um intermediário, recebendo solicitações de uma aplicação e retornando respostas de outra, sem que o usuário precise conhecer os detalhes internos de como o sistema opera nos bastidores.
  - Um conjunto de regras e definições que permite que diferentes softwares se comuniquem entre si.



API - Conceitos Básicos

Interface de Programação de Aplicações

- **Funcionamento básico:**
  - Um aplicativo (o cliente) envia uma requisição para a API.
  - A API leva essa requisição a outro sistema (o servidor) que possui os dados ou a funcionalidade necessária.
  - O servidor processa a requisição e devolve uma resposta à API.
  - A API entrega a resposta de volta ao aplicativo.
- **Usos:**
  - Permitir que um app de clima acesse dados meteorológicos.
  - Integrar sistemas de pagamento em sites de e-commerce.
  - Conectar redes sociais a aplicativos externos.

API - Conceitos Básicos

Interface de Programação de Aplicações

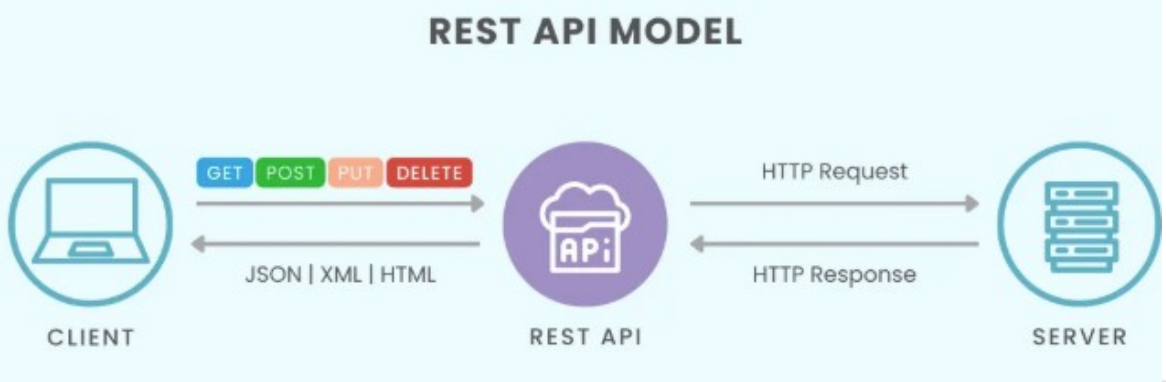
- **Exemplos práticos:**
- **Aplicativos de clima:** Quando você verifica a previsão do tempo no seu smartphone, o aplicativo usa uma API para solicitar os dados mais recentes de um serviço de meteorologia.
- **Pagamentos online:** Um e-commerce utiliza a API de um serviço de pagamento para processar transações financeiras de forma segura, sem precisar lidar com os detalhes bancários do cliente.
- **Login com redes sociais:** Muitos sites e apps permitem que você faça login usando sua conta do Google ou Facebook. Isso acontece através de uma API que permite a comunicação e a autenticação segura entre os sistemas.

Web Service vs. API

- É comum confundir os dois termos, mas a relação é hierárquica:
- **API (Interface de Programação de Aplicações):** É um conceito mais amplo, sendo um conjunto de regras e protocolos que define como duas aplicações interagem.
- **Web Service:** É um tipo de API, mas que funciona especificamente através da rede, usando protocolos web como o HTTP.
- Em resumo, todo Web Service é uma API, mas nem toda API é um Web Service.

Web Service REST

- **Web Service REST (Representational State Transfer)**
  - É um tipo de serviço web que segue princípios arquiteturais simples e eficientes para permitir a comunicação entre sistemas via internet.
  - Ele é amplamente usado em APIs modernas por ser leve, escalável e fácil de implementar.
  - REST não é um protocolo, mas sim um conjunto de restrições arquiteturais para projetar APIs que se baseiam em recursos.

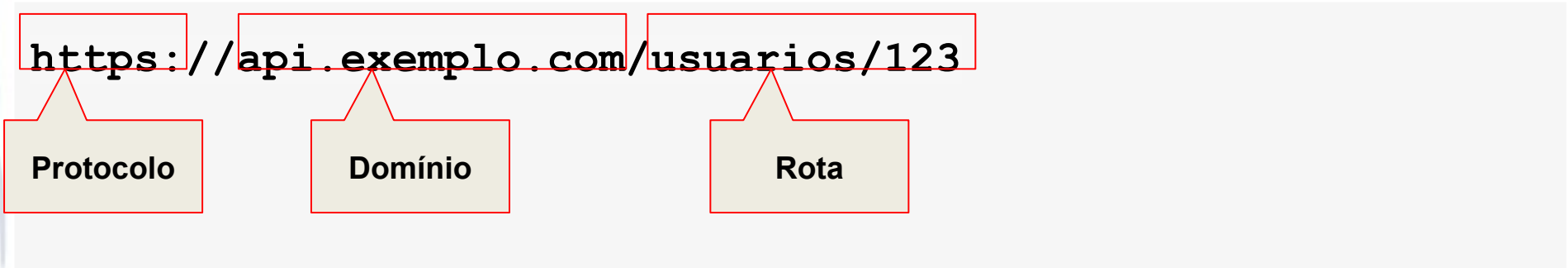


Web Service REST

- **Características principais**
  - **Baseado em HTTP:** Utiliza os métodos padrão do protocolo HTTP (GET, POST, PUT, DELETE, etc.).
  - **Recursos como URIs:** Cada recurso (como usuários, produtos, pedidos) é representado por uma URI única.
  - **Formato leve de dados:** A troca de informações geralmente ocorre em JSON, embora também possa usar XML.
  - **Sem estado (stateless):** Cada requisição é independente; o servidor não guarda informações da requisição anterior.
  - **Cacheável:** Pode usar mecanismos de cache para melhorar a performance.

Web Service REST

- **Características principais**
  - **URI (Uniform Resource Identifier)** É um identificador universal de um recurso na web.
  - Geralmente formada por:
    - protocolo - protocolo de acesso ao serviço.
    - domínio - localizador da máquina que hospeda o serviço.
    - rota - endpoint - caminho, localização do recurso no servidor.



Web Service REST

- **Verbos HTTP e operações CRUD**
  - As APIs REST utilizam os verbos HTTP para definir as ações que podem ser realizadas sobre um recurso.
  - Isso mapeia diretamente para as operações de **CRUD (Create, Read, Update, Delete):**
    - **GET:** Obtém a representação de um recurso (Read).
    - **POST:** Cria um novo recurso (Create).
    - **PUT:** Atualiza um recurso existente (Update).
    - **PATCH:** Atualiza parcialmente um recurso existente.
    - **DELETE:** Remove um recurso (Delete).

Web Service REST

- **Formatos de dados**
  - A representação dos recursos é geralmente enviada em formatos leves e de fácil leitura, como **JSON (JavaScript Object Notation)** ou **XML (eXtensible Markup Language)**.
  - Sendo que o **JSON** se tornou o padrão mais comum em APIs REST.

Web Service REST

- **JSON (JavaScript Object Notation)**
  - É um formato leve e de fácil leitura para representar dados estruturados.
  - Ele é baseado em texto e usa uma sintaxe simples, parecida com objetos do JavaScript, mas é compatível com praticamente todas as linguagens de programação.
  - **Chave-valor:** Cada dado é representado por uma chave (string) e um valor.

Web Service REST

- **Exemplo de JSON:**

```
{
  "Curso": "Programação Python POO",
  "estudantes": 30,
  "ementa": [
    "Estrutura de dados",
    "Python Básico",
    "Python POO",
    "Web Services",
    "Banco de Dados"
  ],
  "ativo": true
}
```

```
{
  "id": 123,
  "nome": "Fulano",
  "email": "Fulano@email.com"
}
```

Web Service REST

- **Exemplo:**
  - Considere um sistema de gerenciamento de livros.
  - A API REST poderia funcionar da seguinte forma:
    - **GET /livros:** Retorna uma lista de todos os livros.
    - **POST /livros:** Cria um novo livro com base nos dados enviados.
    - **GET /livros/123:** Retorna os detalhes do livro com o ID 123.
    - **PUT /livros/123:** Atualiza os dados do livro com o ID 123.
    - **DELETE /livros/123:** Exclui o livro com o ID 123.

Web Service REST

- **Exemplo prático de consulta a um API:**

  API de consulta de endereços via CEP

  https://viacep.com.br/

```
https://viacep.com.br/ws/72320-328/json/
{
  "cep": "72320-328",
  "logradouro": "Quadra QI 416 Área Especial",
  "complemento": "s/n CBM",
  "unidade": "",
  "bairro": "Samambaia Norte (Samambaia)",
  "localidade": "Brasília",
  "uf": "DF",
  "estado": "Distrito Federal",
  "regiao": "Centro-Oeste",
  "ibge": "5300108",
  "gia": "",
  "ddd": "61",
  "siafi": "9701"
}
```

Web Service REST

- **Exemplo prático de consulta a um API:**

  API de consulta de filmes

  https://www.omdbapi.com/

Examples

By Title

Title: avatar Year: Plot: Short Response: JSON Search Reset

Request:

http://www.omdbapi.com/?t=avatar

Response:

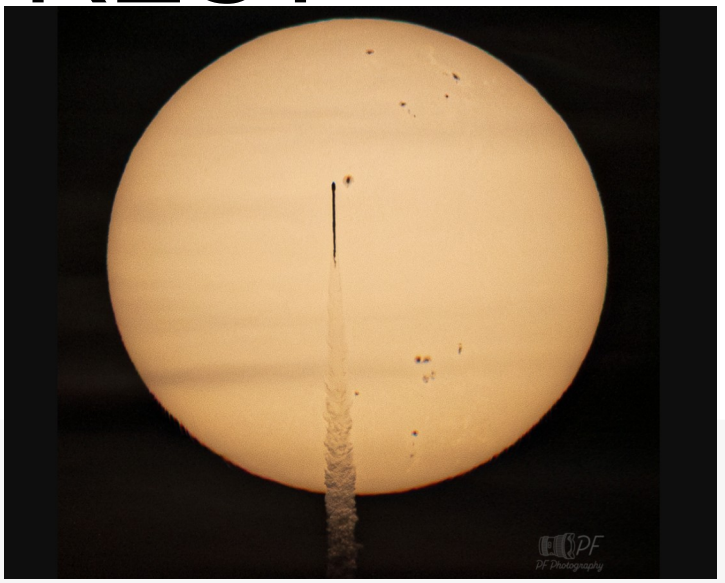
{"Title":"Avatar","Year":"2009","Rated":"PG-13","Released":"18 Dec 2009","Runtime":"162 min","Genre":"Action, Adventure, Fantasy","Director":"James Cameron","Writer":"James Cameron","Actors":"Sam Worthington, Zoe Saldaña, Sigourney Weaver","Plot":"A paraplegic Marine dispatched to the moon Pandora on a unique mission becomes torn between following his orders and protecting the world he feels is his home.","Language":"English, Spanish","Country":"United States, United Kingdom","Awards":"Won 3 Oscars. 91 wins & 131 nominations total","Poster":"https://m.media-amazon.com/images/M/MV5BMDEzMmQwZjctZWU2My00MWNlLWE0NjItMDJlYTRlNGJiZjcyXkEyXkFqcGc@._V1_SX300.jpg","Ratings":[{"Source":"Internet Movie Database","Value":"7.9/10"},{"Source":"Rotten Tomatoes","Value":"81%"},{"Source":"Metacritic","Value":"83/100"}],"Metascore":"83","imdbRating":"7.9","imdbVotes":"1,444,157","imdbID":"tt0499549","Type":"movie","DVD":"N/A","BoxOffice":"$785,221,649","Production":"N/A","Website":"N/A","Response":"True"}

Web Service REST

- **Exemplo prático de consulta:**

  API de consulta da NASA

  https://api.nasa.gov/



https://api.nasa.gov/planetary/apod?api_key=DEMO_KEY


```
{
  "copyright": "Pascal Fouquet",
  "date": "2025-09-27",
  "explanation": "On the morning of September 24 a rocket crosses the bright solar disk in this long range telescopic snapshot captured from Orlando, Florida. That's about 50 miles north of its Kennedy Space Center launch site. This rocket carried three new space weather missions to space. Signals have now been successfully acquired from all three - NASA's Interstellar Mapping and Acceleration Probe, NASA’s Carruthers Geocorona Observatory, and the National Oceanic and Atmospheric Administration (NOAA) Space Weather Follow-On Lagrange 1 (SWFO-L1) - as they begin their journey to L1, an Earth-Sun lagrange point. L1 is about 1.5 million kilometers in the sunward direction from planet Earth. Appropriately, major space weather influencers, aka dark sunspots in active regions across the Sun, are posing with the transiting rocket. In fact, large active region AR4225 is just right of the rocket's nose.",
  "hdurl": "https://apod.nasa.gov/apod/image/2509/IMAP-IG2-001.JPG",
  "media_type": "image",
  "service_version": "v1",
  "title": "A Rocket in the Sun",
  "url": "https://apod.nasa.gov/apod/image/2509/IMAP-IG2-001_1024.JPG"
}
```

Web Service REST

- **Exemplo prático de consulta:**

  API loja falsa para testes

  https://fakeapi.platzi.com/en/rest/products/

https://api.escuelajs.co/api/v1/products/14


```
{
  "id":14,
  "title":"Classic High-Waisted Athletic Shorts",
  "slug":"classic-high-waisted-athletic-shorts",
  "price":43,
  "description":"Stay com…….,
  "category":{
    "id":1,
    "name":"Clothes",
    "slug":"clothes",
    "image":"https://i.imgur.com/QkIa5tT.jpeg",
    "creationAt":"2025-09-28T11:18:06.000Z",
    "updatedAt":"2025-09-28T11:18:06.000Z"
  },
  "images":["https://i.imgur.com/eGOUveI.jpeg","https://i.imgur.com/UcsGO7E.jpeg","https://i.imgur.com/NLn4e7S.jpeg"],
  "creationAt":"2025-09-28T11:18:06.000Z",
  "updatedAt":"2025-09-28T11:18:06.000Z"
}
```

Atividade de Fixação

Procurar APIs gratuitas, se cadastrar se necessário para obter o token de acesso, e realizar consultas.

FIM