**UNIVERSIDADE FEDERAL DE OURO PRETO – UFOP**

**CIÊNCIA DA COMPUTAÇÃO**

****

****

**SISTEMAS OPERACIONAIS**

**TRABALHO PRÁTICO 5**

Marcus Vinícius Souza Fernandes

19.1.4046

**Ouro Preto**

**2021**

**Métodos HTTP**

O HTTP (Hypertext Transfer Protocol) é um protocolo de comunicação. Através dele o cliente e o servidor conseguem se comunicar, seguindo um conjunto de regras bem definidas (por isso chamamos de protocolo). A Request ou requisição traduzindo diretamente para português, é o pedido que um cliente realiza ao servidor. Esse pedido contém uma série de dados que são usados para descrever exatamente o que o cliente precisa. No navegador toda vez que trocamos de página ou apertamos enter na barra de endereço uma nova request é feita. Independente se estamos apenas pedindo a exibição de uma página, cadastrando um novo recurso, atualizando ou excluindo.

[GET](https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods/GET): O método GET solicita a representação de um recurso específico. Requisições utilizando o método GET devem retornar apenas dados.

[HEAD](https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods/HEAD): O método HEAD solicita uma resposta de forma idêntica ao método GET, porém sem conter o corpo da resposta.

[POST](https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods/POST): O método POST é utilizado para submeter uma entidade a um recurso específico, frequentemente causando uma mudança no estado do recurso ou efeitos colaterais no servidor.

[PUT](https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods/PUT): O método PUT substitui todas as atuais representações do recurso de destino pela carga de dados da requisição.

[DELETE](https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods/DELETE): O método DELETE remove um recurso específico.

[CONNECT](https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods/CONNECT): O método CONNECT estabelece um túnel para o servidor identificado pelo recurso de destino.

**OAS3.0 (Open API)**

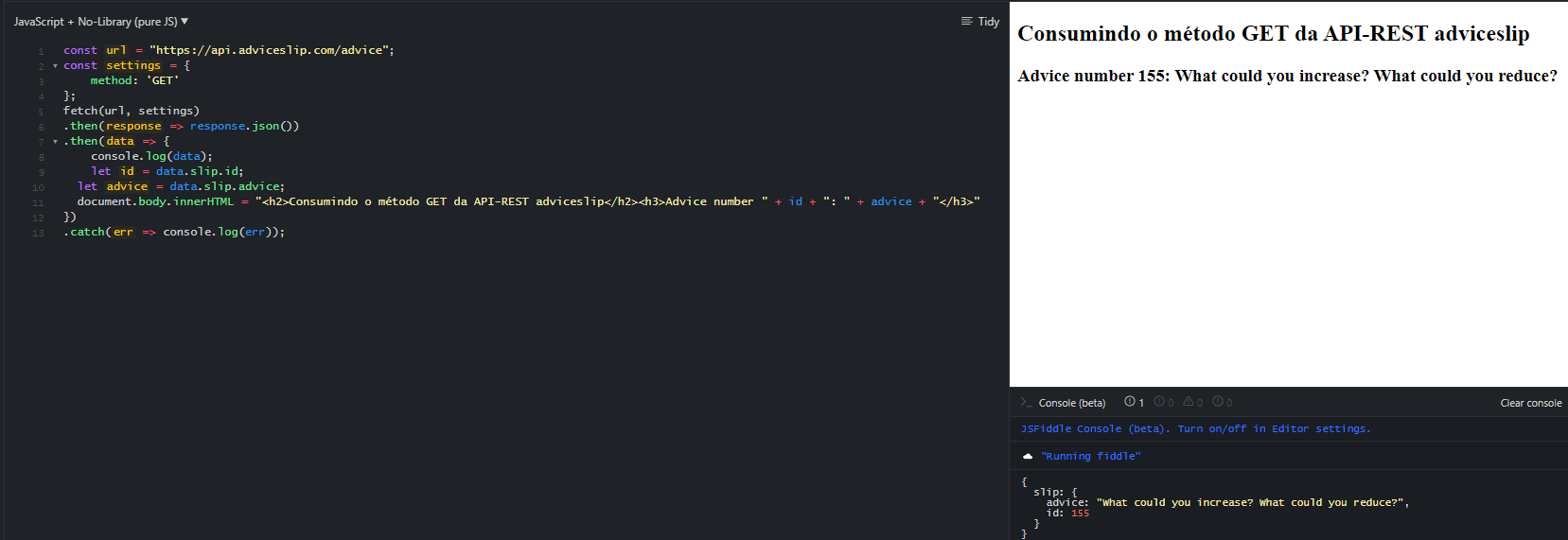
A Especificação OpenAPI é uma especificação aberta dirigida pela comunidade dentro da [Iniciativa OpenAPI](https://www.openapis.org/), um Projeto Colaborativo da Linux Foundation. Ela define uma descrição de interface agnóstica de linguagem de programação padrão para APIs HTTP, que permite que humanos e computadores descubram e entendam os recursos de um serviço sem exigir acesso ao código-fonte, documentação adicional ou inspeção do tráfego de rede. Quando definido corretamente por meio do OpenAPI, um consumidor pode entender e interagir com o serviço remoto com uma quantidade mínima de lógica de implementação.

**Consumindo uma API pública (Método GET)**

Como solicitado, a api-rest escolhida foi a “Advice Slip” (<https://api.adviceslip.com>), se trata de um serviço simples que basicamente em um de seus métodos GET, nos retorna um conselho aleatório. A mesma possuí uma documentação muito interessante, organizada e acessível. Como a demanda foi relativamente simples, não foi necessário a criação de uma estrutura/projeto complexo para consumir o método desta mesma api.

Foi utilizado uma funcionalidade presente na linguagem javascript denominada “fetch”, ela fornece uma interface para acessar e manipular partes do pipeline HTTP, tais como requests e responses. Hoje, esta funcionalidade substituí o “XMLHttpRequest”.

O script criado e evidenciado abaixo se resume a atribuição do endereço equivalente ao método get random da api escolhida, construímos as configurações para a requisição (GET) e em seguida utilizamos a funcionalidade oferecida pelo fetch, no qual modelamos a resposta para o formato JSON e filtramos a data obtida para que possamos mostra-la em tela via construção HTML, comprovando assim, a eficácia do código. Também tratamos a ocorrência de possíveis erros com a funcionalidade catch.



Contextualizando a imagem, à esquerda se encontra o código criado na linguagem javascript, à direita superior o nosso HTML gerado com as informações obtidas e no canto inferior direito temos o JSON obtido na requisição, sem nenhum tratamento.

**Referências**

<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>

<https://github.com/OAI/OpenAPI-Specification>

https://developer.mozilla.org/pt-BR/docs/Web/API/Fetch\_API/Using\_Fetch