



Universidade Federal de Uberlândia

Faculdade de Computação

10º Trabalho de Programação para Internet – Prof. Daniel A. Furtado

Trabalho Individual – Requisições Assíncronas com a API Fetch sem async/await

Instruções Gerais

- Esta atividade deve ser realizada individualmente;
- Tecnologias permitidas: HTML5, CSS, JavaScript, Bootstrap, PHP, MySQL e API Fetch **sem** async/await (o emprego de async/await anulará o trabalho). Não é permitido usar o objeto XMLHttpRequest.
- Sintaxe da XHTML como ou
 não é permitida (anulará o trabalho);
- O website deve ser hospedado e disponibilizado online, conforme orientações disponíveis no final deste documento;
- Ao construir o website, utilize dados fictícios (**jamais utilize** dados pessoais como seu nome, CPF, endereço, e-mail etc.);
- Esteja atento às **observações sobre plágio** apresentadas no final deste documento;
- Trabalhos com implementações utilizando trechos de códigos retirados de sites da Internet ou de trabalhos de semestres anteriores serão anulados;
- As páginas web não devem conter qualquer conteúdo de caráter imoral, desrespeitoso, pornográfico, discurso de ódio, desacato etc.;
- O website deve ser validado utilizando as ferramentas disponíveis nos endereços **validator.w3.org** e **jigsaw.w3.org/css-validator** (não deve conter nenhum erro ou *warning*);
- O trabalho deve ser entregue até a data/hora definida pelo professor. Não deixe para enviar o trabalho nos últimos instantes, pois eventuais problemas relacionados à eventos adversos como instabilidade de conexão, congestionamento de rede, etc., não serão aceitos como motivos para entrega da atividade por outras formas ou em outras datas;
- Este trabalho deve ser feito **mantendo os trabalhos anteriores intactos**, ou seja, os trabalhos anteriores devem permanecer online conforme foram entregues;
- Trabalhos enviados por e-mail ou pelo MS Teams **não serão considerados**.

Leia os slides de aula 1-41 disponibilizados no endereço a seguir e resolva os exercícios seguintes.

<http://www.furtado.prof.ufu.br/site/teaching/PPI/PPI-Modulo7-AJAX-Parte2.pdf>

Exercício 1

Abra o arquivo <http://www.furtado.prof.ufu.br/site/teaching/PPI/Exemplos-Ajax-Fetch.zip> e coloque o exemplo **Ex1-cep-fetch** online, na pasta trab10/ex1, utilizando seu espaço de hospedagem no infinityfree. Analise o código JavaScript comparando-o com o código corresponde apresentado no trabalho anterior (que utilizou o XHR).

Exercício 2

Coloque o exemplo **Ex2-encadeamento-fetch** online, na pasta trab10/ex2. Teste o exemplo e em seguida observe o código JavaScript no Visual Studio Code. Explique o código JavaScript em detalhes e disponibilize a explicação no final da página **index.html** do exemplo.

Exercício 3

Faça uma cópia do exemplo **Ex6-login** apresentado no trabalho anterior e altere a cópia para que a função **sendForm** passe a enviar o formulário por meio de requisição Ajax com a API Fetch (ao invés do XHR). Disponibilize o exemplo modificado na pasta trab10/ex3.

Exercício 4

Este exercício utiliza uma versão gratuita da API **Alpha Vantage** para obtenção de cotações de ações na bolsa de valores. A versão gratuita da Alpha Vantage tem um limite de 5 requisições por minuto e 500 requisições por dia. Portanto, pode ser necessário aguardar alguns instantes antes de recarregar a página ou repetir as requisições.

- Faça o registro para obtenção da chave de API gratuita utilizando o link <https://www.alphavantage.co/support/#api-key>;
- Faça um teste de acesso ao serviço utilizando o endereço a seguir. Troque o texto **SuaChaveDeApi** pela chave obtida no passo anterior. Observe que a cotação da ação deve ser informada na própria URL utilizando o símbolo da ação na B3 seguido por ".SA".
https://www.alphavantage.co/query?function=GLOBAL_QUOTE&symbol=PETR4.SA&apikey=SuaChaveDeApi
- Utilizando a API Fetch, crie uma função JavaScript de nome **GetQuote** que receba como parâmetro o código da ação e retorne uma **promise** que seja cumprida produzindo como resultado um objeto JavaScript contendo a cotação da ação e o respectivo código da ação. Por exemplo, ao chamar `GetQuote("PETR4")`, a função deve retornar uma *promise* que se resolva em um objeto similar a: `{ acao: "PETR4", cotacao: "26,30" }`. A função `GetQuote` **não** deve criar **explicitamente** uma nova *promise*;
- Crie uma função JavaScript de nome **RenderStock** que receba como parâmetro um objeto contendo o código da ação e a respectiva cotação e mostre o mesmo na página HTML;
- Crie um formulário contendo 5 campos textuais (para que o usuário possa informar 5 códigos de ações) e um botão **Cotar Sequencialmente**. Crie uma função JavaScript que resgate os códigos das ações fornecidos pelo usuário e faça 5 requisições Ajax **encadeadas** para obtenção das cotações e apresentação na página à medida em que cada requisição seja finalizada. A função não deve efetuar uma nova requisição antes que a requisição anterior tenha sido finalizada. A página deverá mostrar o resultado como na figura a seguir:

Cotações das Ações Indicadas

ABEV3: 14,6

ASAI3: 11,15

AZUL4: 12,36

B3SA3: 12,81

BIDI11: 10,06

- f) Acrescente um novo botão **Cotar em Paralelo** e crie uma função JavaScript para efetuar requisições em paralelo para obter as cotações das ações. A função deve utilizar o método [Promise.all](#) e só deve apresentar as cotações na tela quando todas as requisições finalizarem. Ao testar o exemplo, observe os tempos das requisições utilizando o ambiente de desenvolvimento do navegador (F12). Compare com as requisições sequenciais do item anterior.

Disponibilização Online

As páginas dos exercícios devem ser disponibilizadas online utilizando o subdomínio gratuito registrado anteriormente, porém em pasta própria (isto é, seusubdominio.com/trabalhoX/ex1, seusubdominio.com/trabalhoX/ex2, etc.). Não altere ou exclua as pastas dos trabalhos anteriores.

Acrescente um arquivo de nome **index.html** na pasta raiz do trabalho contendo links para as páginas dos exercícios.

Entrega

Além da disponibilização online, a pasta raiz contendo as subpastas dos exercícios deve ser compactada no formato zip e enviada pelo Sistema Acadêmico de Aplicação de Testes (SAAT) até a data limite indicada pelo professor em sala de aula.

Adicione também um arquivo de nome **link.txt**, na pasta raiz, contendo a URL do trabalho online (para a pasta raiz do trabalho).

Sobre Eventuais Plágios

Este é um trabalho individual. Os alunos envolvidos em qualquer tipo de plágio, total ou parcial, seja entre equipes ou de trabalhos de semestres anteriores ou de materiais disponíveis na Internet (exceto os materiais de aula disponibilizados pelo professor), serão duramente penalizados (art. 196 do Regimento Geral da UFU). Todos os alunos envolvidos terão seus **trabalhos anulados** e receberão **nota zero**.