Nesta lição, vamos trabalhar na adição de funcionalidade ao nosso botão “Adicionar ao Carrinho”.

**Configuração do Contêiner do Carrinho**

Para começar, vamos adicionar um contador de preço para o nosso carrinho. No **index.html**, acima do nosso **contêiner**, vamos adicionar um novo **div** para conter nosso **carrinho**. Também daremos a ele um preço padrão.

<div id="cart-container">

    $0

</div>

Em seguida, vamos melhorar a aparência do nosso contêiner no **style.css**, alterando a largura e as cores.

#cart-container {

width: 100%;

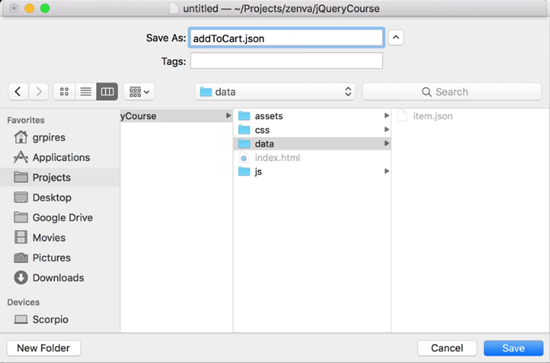
background-color: #000;

color: #FFF;

padding: 12px;

}

Agora podemos passar para o **main.js** para trabalhar na funcionalidade real do nosso botão “Adicionar ao Carrinho”. Para fazer isso, precisamos fazer uma **chamada AJAX do tipo POST**. Com uma chamada **POST**, enviamos informações para o servidor e o servidor nos envia uma resposta. Como não estamos usando um servidor real, precisamos criar um **novo arquivo** **JSON** para simular isso. Vamos chamá-lo de **addToCart.json** e salvá-lo em nossa **pasta data**.



Neste arquivo **JSON**, simplesmente queremos configurar uma mensagem confirmando que o item foi adicionado ao carrinho e um preço de amostra.

{

    "message": "sucesso",

    "price": 499

}

**Configuração da Chamada AJAX**

Abaixo da chamada **AJAX atual**, vamos criar uma função de **evento “clique”** que é configurada em nosso **contêiner** e usa o botão “Adicionar ao Carrinho” como delegado. É dentro da função deste evento que faremos nossa **chamada AJAX**.

$('#container').on('click','.item-add',function(){

});

(**Observação**: Verifique se seu botão “Adicionar ao Carrinho” tem a classe correta na função addItem no início do main.js).

Da mesma forma que antes, precisamos especificar que queremos acessar **addToCart.json** na chamada. Também precisamos adicionar alguns parâmetros adicionais. Como queremos**“enviar”** informações, precisamos especificar o **tipo** como **post** (ao contrário da chamada anterior que usava o tipo padrão get). Também declararemos os tipos de dados e de conteúdo como antes. No entanto, como estamos enviando e recebendo dados, precisamos especificar o que estamos enviando e o que queremos como resposta.

$('#container').on('click','.item-add',function(){

    $.ajax('data/addToCart.json', {

        type: 'post',

        data: { },

        dataType: 'json',

        contentType: 'application/json'

        })

    });

});

Para simplificar as coisas, vamos enviar e receber **IDs**. Para isso funcionar, precisamos voltar para o arquivo **item.json** e adicionar **IDs** a cada um dos nossos itens.

{

 "items": [{

        "id": 1000,

        "name": "Santa Monica",

        "description": "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do

        eiusmod tempor incididunt ut labore et dolore magna aliqua.",

        "price": 499,

        "moreInfo": "Lorem ipsum dolor sit amet"

    }, {

        "id": 1001,

        "name": "Recife",

        "description": "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do

        eiusmod tempor incididunt ut labore et dolore magna aliqua.",

        "price": 499,

        "moreInfo": "Lorem ipsum dolor sit amet"

    }, {

        "id": 1002,

        "name": "Long Beach",

        "description": "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do

        eiusmod tempor incididunt ut labore et dolore magna aliqua.",

        "price": 499,

        "moreInfo": "Lorem ipsum dolor sit amet"

    }]

}

No **main.js**, agora podemos usar esse **ID** como a informação sendo enviada e recuperada. Para salvar corretamente esse **ID** em nosso **HTML**, também adicionaremos isso à nossa função **addItem**. Na parte do HTML, usaremos **data-id** para especificar para que serve.

let html = '';

html += '<div class="item" data-id="' + id + '">';

html += '<div class="name">' + name + '</div>'

Por fim, precisamos refletir nosso parâmetro adicional em nossa **chamada AJAX** original para o **item.json**.

.done(function(response){

    let items = response.items;

    items.forEach(function(item){

    addItem(item.id, item.name, item.description, item.price, item.moreInfo);

    });

})

**Concluindo a chamada AJAX de post**

Agora que fizemos essas adições e podemos especificar nossos dados corretamente, podemos concluir nossa chamada. Acima de nossa nova **chamada** **AJAX** em nosso **evento de clique**, precisamos salvar o **ID** como uma variável, usando alguns métodos de **travessia** para chegar lá a partir do nosso botão. Até agora, nosso código ficará assim:

$('#container').on('click','.item-add',function(){

    let id = $(this).parent().data('id');

    $.ajax('data/addToCart.json', {

        type: 'post',

        data: { id: id },

        dataType: 'json',

        contentType: 'application/json'

        })

    });

});

Precisamos encadear isso agora com uma **chamada done** para que possamos obter a resposta apropriada. Usando nossa mensagem de sucesso, podemos criar uma **declaração if** que adicionará adequadamente o preço da resposta (ou seja, o objeto com o ID correspondente) a uma **variável** chamada **“cart”** (carrinho). Em seguida, podemos alterar o texto do nosso **cart-container** para corresponder ao preço armazenado na **variável “cart”**.

$('#container').on('click','.item-add',function(){

    let id = $(this).parent().data('id');

    $.ajax('data/addToCart.json', {

        type: 'post',

        data: { id: id },

        dataType: 'json',

        contentType: 'application/json'

    })

    .done(function(response) {

        if (response.message === 'success') {

        let price = response.price;

        cart += price;

        $('#cart-container').text('$' + cart);

    }

    });

    });

});

Claro, também precisamos adicionar essa **variável cart** ao nosso arquivo **main.js.** Podemos fazer isso facilmente adicionando a seguinte linha no topo do arquivo.

var cart = 0;

Agora, ao clicar no botão “Add to Cart”, deve ser exibida a alteração no preço no topo!

