### Aula 25: Postando o fomulário

Nesta última lição, vamos adicionar um botão “submit” e usar uma chamada AJAX post para enviar informações para o servidor.

**Preparando o Formulário para Envio**

Para enviar nossas informações, precisamos fazer algumas alterações no arquivo **index.html**. Abaixo do nosso menu suspenso, queremos adicionar um botão “submit” que enviará as informações quando clicado. Para que as informações sejam enviadas corretamente, também iremos envolver todos os elementos existentes em **tags de formulário** com um **ID** de **cart-form**. Além disso, iremos declarar um método **post** e enviá-lo especificamente para**cartSubmit.json**.

<div id="user-data">

    <form id="cart-form" method="post" action="data/cartSubmit.json">

        <br/>

        <label>Name</label>

        <input type="text"></input>

        <br/>

        <label>Email</label>

        <input type="text"></input>

        <br/>

        <label>Newsletter?</label>

        <input id="newsletter-checkbox" checked type="checkbox"></input>

        <br/>

        <select id="newsletter-frequency">

            <option>-</option>

            <option selected>Weekly</option>

            <option>Monthly</option>

        </select>

        <br/>

        <button id="submit-cart">Submit</button>

    </form>

</div>

Vamos criar um **novo arquivo** no Atom chamado **cartSubmit.json**. Neste arquivo, vamos simplesmente ter uma mensagem de sucesso como conteúdo.

{

    "message": "success"

}

**Chamada AJAX para Envio do Formulário**

Agora podemos criar uma função no arquivo **main.js**. No final da nossa **função document ready**, iremos criar um **ouvinte de** **eventos**no **formulário**(**cart-form**) usando um novo tipo de evento específico para formulários: **submit**. O primeiro elemento que queremos adicionar à função é uma chamada **preventDefault**, caso contrário, seríamos redirecionados diretamente para o arquivo **JSON**.

$('#cart-form').on('submit', function(event) {

    event.preventDefault();

});

Em seguida, precisamos configurar nossa **chamada AJAX**. A partir da configuração que fizemos no**index.html**, sabemos que nossas informações estão sendo enviadas para **cartSubmit.json**. Portanto, nossa **chamada AJAX** precisa identificar para onde está sendo enviada, lendo o **atributo action** do objeto de formulário (**cart-form**). Como antes, também precisamos definir parâmetros que declaram que estamos usando um **tipo de chamada post** e os **dados**. Para nossos **dados**, vamos enviar um tipo de formulário com os **dados serializados** do formulário e o preço da nossa **variável de carrinho** (**cart**). Vamos salvar nossos dados em uma variável local separada para uso posterior. No final das contas, nossa função ficará assim:

$('#cart-form').on('submit',function(event){

    event.preventDefault();

    let data = { form: $(this).serialize(), price: cart };

    $.ajax($(this).attr('action'), {

        type: 'post',

        data: data

    })

});

Usando nossa **chamada done**, queremos ser capazes de imprimir nossa resposta como uma string. No entanto, para fazer isso, precisamos fazer algumas alterações novamente em nosso arquivo **index.html**. **Para cada tag de input**, precisamos adicionar um **atributo name** que possa ser convertido em uma string.

<div id="user-data">

    <form id="cart-form" method="post" action="data/cartSubmit.json">

        <br/>

        <label>Name</label>

        <input name="name" type="text"></input>

        <br/>

        <label>Email</label>

        <input name="email" type="text"></input>

        <br/>

        <label>Newsletter?</label>

        <input name="wantNewsletter" id="newslettercheckbox" checked type="checkbox"></input>

        <br/>

        <select name="newsletterFrequency" id="newsletter-frequency">

            <option>-</option>

            <option selected>Weekly</option>

            <option>Monthly</option>

        </select>

        <br/>

        <button id="submit-cart">Submit</button>

    </form>

</div>

Agora podemos adicionar nossa **chamada done** à nossa **chamada AJAX**. Usando um **alert**, podemos imprimir os dados recebidos do formulário, bem como nossa mensagem de sucesso.

$('#cart-form').on('submit',function(event){

    event.preventDefault();

    let data = { form: $(this).serialize(), price: cart };

    $.ajax($(this).attr('action'), {

        type: 'post',

        data: data

    })

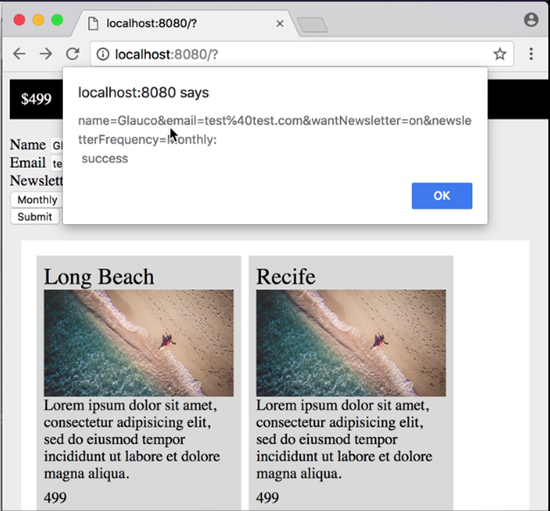
    .done(function(response){

        alet(data.form + ': ' + response.message);

    });

});

No navegador, podemos ver que os dados do formulário estão sendo enviados com sucesso por meio desse método!



**Última Adição**

Vamos fazer uma última adição. Abaixo da **tag de fechamento** do nosso formulário, mas antes da **tag de fechamento** do**container do formulário** (cart-form), vamos adicionar uma **div** para exibir uma mensagem de feedback.

<div id="feedback-message"></div>

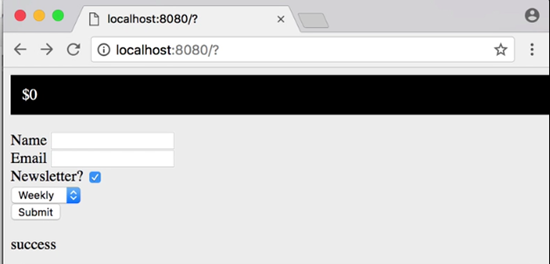
Agora, no **main.js**, podemos alterar nossa **chamada AJAX** .done configurando uma função que imprime a **mensagem** de resposta na nossa nova **div** **de feedback**.

.done(function(response) {

    $('#feedback-message').text(response.message);

});

Agora, quando você enviar o formulário, receberá a mensagem de sucesso.



Parabéns! Você concluiu o curso de **jQuery** e aprendeu como adicionar funcionalidades extras aos seus sites.