Nesta lição, vamos trabalhar em nosso arquivo **item.json** e carregá-lo.

**Configuração da Chamada AJAX**

Na última lição, configuramos nossa chamada **AJAX** como faríamos com um evento. No entanto, na verdade, não queremos chamar nosso arquivo dessa forma. Em vez disso, queremos simplificar as coisas e encadear três chamadas de função com nossa chamada **AJAX**.

$.ajax('data/item.json')

    .done()

    .fail()

    .always();

Nessas chamadas, adicionaremos nossas funções reais. Nossa **chamada de função “done”** é aquela que será chamada sempre que a chamada **AJAX** for concluída, para que possamos mover nossa **função de resposta** original para esta chamada. Como o nome sugere, **“fail”** é chamado quando a chamada **AJAX** falha. Como isso ajudará na depuração, vamos criar uma função que imprime uma mensagem de erro para nós. Vamos deixar a função **“always”** vazia, pois não precisamos que nenhuma parte de nossa página seja chamada o tempo todo, independentemente do sucesso ou falha.

$.ajax('data/item.json')

    .done(function(response) {

        console.log(response);

    })

    .fail(function(request, errorType, errorMessage) {

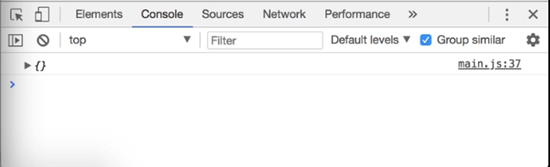
        console.log(errorMessage);

    })

    .always(function() {

    });

Adicionando chaves ao **item.json**, você pode testar o código e ver que ele imprime as chaves no console. Se você não vir isso, o console exibirá uma mensagem de erro.





**Adicionando itens ao arquivo JSON**

Agora que verificamos nossa chamada **AJAX**, vamos criar nossos itens em**item.json**. Para fazer isso, precisamos criar um **vetor** de **itens** e fazer com que cada elemento do **vetor** seja uma **lista** de todos os aspectos de que precisamos. No código abaixo, criamos três itens, incluindo um que usa as informações de “Santa Monica” que tínhamos antes.

{

     "item":[{

         "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"

     }, {

         "name": "Recife",

         "description": "Lorem ipsum dolor sit amet, consectetur adipisicing elit, seddo eiusmod tempor incididunt ut labore et dolore magna aliqua.",

         "price": 499,

         "moreInfo": "Lorem ipsum dolor sit amet"

     }, {

         "name": "Long Beach",

         "description": "Lorem ipsum dolor sit amet, consectetur adipisicing elit, seddo eiusmod tempor incididunt ut labore et dolore magna aliqua.",

         "price": 499,

         "moreInfo": "Lorem ipsum dolor sit amet"

     }

     ]

}

De volta ao **main.js**, agora podemos alterar nossa **chamada “done”** para responder com nossos itens. No console, você poderá ver que conseguimos visualizar com sucesso esses três elementos.

**$.ajax(‘data/item.json’)**

**.done(function(response) {**

**let items = response.item;**

**console.log(items);**

**})**

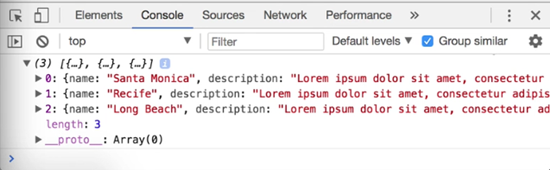
**.fail(function(request, errorType, errorMessage) {**

**console.log(errorMessage);**

**})**

**.always(function() {**

**});**



 Agora queremos criar uma função que realmente adiciona o conteúdo à nossa página. No topo do **main.js**, acima da **função document ready**, vamos adicionar uma nova função chamada **addItem**. Esta função receberá como parâmetros o nome, descrição, preço e mais informações do item. Para simplificar, podemos **copiar** e **colar** grande parte do nosso código anterior do botão “Adicionar item” do Admin. No entanto, onde for relevante, vamos substituir algumas linhas pelos nossos parâmetros em strings concatenadas.

 function addItem(name, description, price, moreInfo) {

     let html = '';

     html += '<div class="item">';

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

     html += '<div class="description">' + description + '</div>';

     html += '<div class="price">' + price + '</div>';

     html += '<button class="item-add">Add to cart</button>';

     html += '<br />';

     html += '<a href="#">More info</a>';

     html += '<div class="more-info">' + moreInfo + '</div>';

     html += '</div>';

     $('#container').prepend(html);

}

Enquanto estamos nisso, também vamos **remover** três coisas:

* O admin-container de index.html
* O conteúdo dentro do nosso div container, exceto a quebra de linha
* A função button-create-item de main.js

Com isso removido, podemos prosseguir para adicionar nossos itens dinamicamente. Para fazer isso, precisamos **iterar** sobre cada item em nosso **array** e chamar nossa nova **função addItem**. Podemos fazer isso usando um **loop foreach** dentro da nossa **chamada done**. Lembre-se de incluir os quatro parâmetros necessários para que nossa função funcione corretamente.

$.ajax('data/item.json')

    .done(function(response) {

        let items = response.item;

        items.forEach(function(item){

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

        });

    })

    .fail(function(request, errorType, errorMessage) {

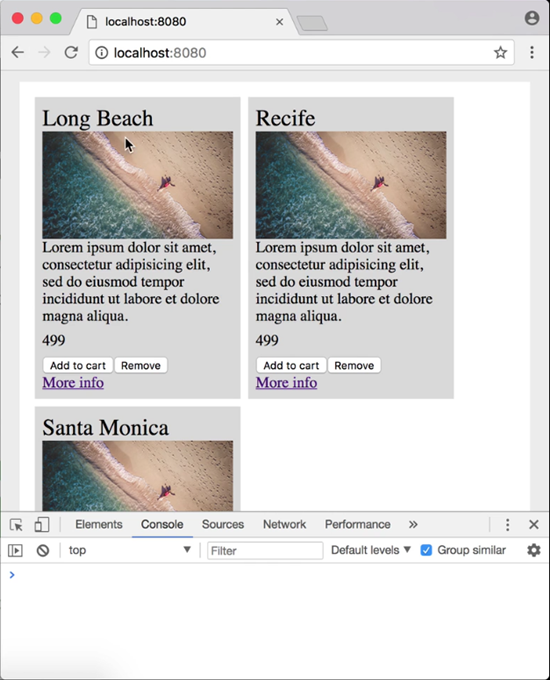
        console.log(errorMessage);

    })

    .always(function() {

    });

Se você testar a página, seus itens agora serão adicionados a partir do **arquivo JSON**.



**Evitando problemas de cache**

Antes de terminar, precisamos lembrar que os navegadores fazem **cache** das páginas para que não seja necessário baixar a mesma página várias vezes. No entanto, com o **AJAX**, não queremos que o navegador faça **cache** dos itens que estamos adicionando dinamicamente. Portanto, para nossa **chamada AJAX**, precisamos adicionar um segundo parâmetro que informe ao nosso código o tipo de dado que está sendo usado, o tipo de conteúdo e que não faça cache das informações do **JSON**.

$.ajax('data/item.json', {

    dataType: 'json',

    contentType: 'application/json',

    cache: false

})