



www.devmedia.com.br

[versão para impressão]

Link original: <http://www.devmedia.com.br/articles/viewcomp.asp?comp=27658>

Populando selects dinamicamente com AJAX, JSON e PHP

Veja neste artigo como popular selects dinamicamente utilizando requisições AJAX no formato JSON consumindo consultas em PHP.

Olá pessoal, neste artigo será demonstrado como popular selects através de requisições AJAX a scripts em PHP, retornando os dados no formato JSON. Como exemplo, será construída uma página contendo listagens de Países, Estados e Cidades inseridas em selects, que serão filtrados conforme a opção selecionada.

Para construirmos esse exemplo, será necessário o seguinte ambiente de desenvolvimento, na seção links seguem endereço para downloads:

- Pacote WampServer (pode ser outro da preferência do leitor);
- Biblioteca jQuery (atualmente na versão 1.9.1);
- Editor de texto (Bloco de Notas, Notepad++, DREAMWEAVER, etc.);

Será utilizado CSS para melhorar o visual da única página "index.php", mas não serão detalhadas suas funcionalidades, pois o mesmo não faz parte do objetivo desse artigo, o nome do arquivo será "estilo.css".

Para executarmos as requisições AJAX no formato JSON, será utilizada a função "\$.getJSON()", lembrando que ela pertence à biblioteca jQuery. Então se torna indispensável linkar na página o caminho para essa biblioteca.

Visando a simplicidade, do artigo será criado apenas um script de consulta PHP, "consulta.php", onde através do método GET serão diferenciadas as funções de consultas (PAIS, ESTADO e CIDADE). Essas funções irão retornar os dados em JSON, para posteriormente serem consumidos pela função "\$.getJSON()".

Todos os dados estão gravados em uma base de dados "atlas" no MySQL, o que torna necessário configurar uma conexão com o banco para executarmos consultas. Será criado um arquivo conexao.php contendo o script de conexão em PDO.

Como exemplo, serão adicionados na página "index.php" 3 selects, onde, conforme for selecionado um País, serão carregados os estados daquele país no select "cmbEstado", quando selecionado um estado, serão carregadas as cidades daquele estado no select "cmbCidade", ou seja, um filtro bem dinâmico.

O nosso banco irá conter:

- Todos os Países do mundo;
- Todos Estados (somente do Brasil);
- Todas as Cidades (somente do Brasil);

Observação: Por motivos óbvios, não foram cadastrados todos os estados de todos os países do mundo, o mesmo vale para cidades. No banco existe um cadastro aproximado de todos os Países, todos os estados do Brasil e aproximadamente todas as cidades do Brasil, mas já serve como exemplo para esse artigo.

Antes de iniciar o desenvolvimento, é interessante organizar os arquivos dentro da pasta principal do projeto para melhor entendimento e organização do exemplo.

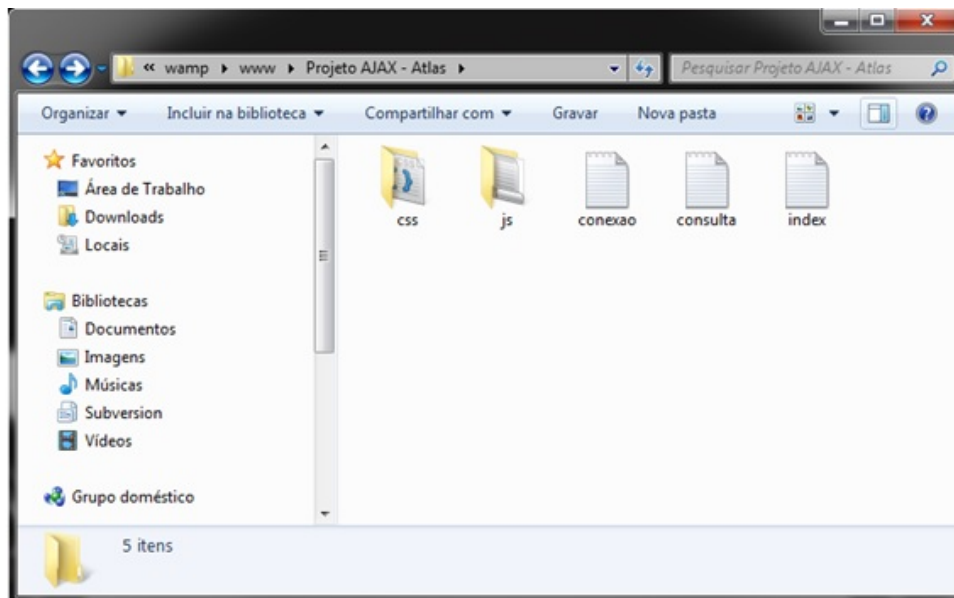


Figura 1: Organização das pastas e arquivos

Como mencionado acima, todas as informações estão gravadas em uma base de dados "atlas" no MySQL, abaixo segue o script para criação do banco e das tabelas.

Listagem 1: Instruções SQL para criar a base de dados

```
CREATE DATABASE IF NOT EXISTS;

CREATE TABLE IF NOT EXISTS `pais` (
  `sigla` char(2) NOT NULL,
  `iso3` char(3) NOT NULL,
  `numcode` smallint(6) NOT NULL,
  `nome` varchar(255) NOT NULL,
  PRIMARY KEY (`sigla`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `estado` (
  `id` int(2) unsigned zerofill NOT NULL AUTO_INCREMENT,
  `sigla` varchar(2) NOT NULL DEFAULT '',
  `nome` varchar(20) NOT NULL DEFAULT '',
  `pais` char(2) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=31 DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `cidade` (
  `id` int(4) unsigned zerofill NOT NULL AUTO_INCREMENT,
  `uf` varchar(4) NOT NULL DEFAULT '',
  `nome` varchar(50) NOT NULL DEFAULT '',
  UNIQUE KEY `id` (`id`),
  KEY `id_2` (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=9715 DEFAULT CHARSET=utf8;
```

Abaixo seguem os scripts em PHP.

conexao.php

Esse script possui uma função "Conectar()", que retorna uma conexão PDO devidamente configurada.

Listagem 2: Script para conexão com o banco de dados

```
<?php
function Conectar(){
    try{
        $opcoes = array(PDO::MYSQL_ATTR_INIT_COMMAND => 'SET NAMES UTF8');
        $con = new PDO("mysql:host=localhost; dbname=atlas;", "root", "011224", $opcoes);
        return $con;
    } catch (Exception $e){
        echo 'Erro: '.$e->getMessage();
        return null;
    }
}
```

consulta.php

Nesse script está toda a lógica das consultas que serão consumidas na página principal através das requisições AJAX. Poderíamos criar um script para cada consulta (PAÍS, ESTADO e CIDADE), mas usando um "SWITCH" podemos selecionar qual consulta será disparada dentro do mesmo script. Conforme a requisição, também será enviado um parâmetro para a instrução SQL que também é tratado nesse "SWITCH".

Uma vez selecionada uma das funções de consulta, a mesma irá retornar os dados no formato JSON, atendendo assim o tipo de requisição feita.

Listagem 3: Script para consultas no banco de dados

```
<?php
require_once('conexao.php');
$opcao = isset($_GET['opcao']) ? $_GET['opcao'] : '';
$valor = isset($_GET['valor']) ? $_GET['valor'] : '';
if (! empty($opcao)){
    switch ($opcao)
    {
        case 'pais':
            {
                echo getAllPais();
                break;
            }
        case 'estado':
            {
                echo getFilterEstado($valor);
                break;
            }
        case 'cidade':
            {
                echo getFilterCidade($valor);
                break;
            }
    }
}

function getAllPais(){
```

```

        $pdo = Conectar();
        $sql = 'SELECT sigla, nome FROM pais';
        $stm = $pdo->prepare($sql);
        $stm->execute();
        sleep(1);
        echo json_encode($stm->fetchAll(PDO::FETCH_ASSOC));
    $pdo = null;
}

function getFilterEstado($pais){
    $pdo = Conectar();
    $sql = 'SELECT sigla, nome FROM estado WHERE pais = ?';
    $stm = $pdo->prepare($sql);
    $stm->bindValue(1, $pais);
    $stm->execute();
    sleep(1);
    echo json_encode($stm->fetchAll(PDO::FETCH_ASSOC));
    $pdo = null;
}

function getFilterCidade($estado){
    $pdo = Conectar();
    $sql = 'SELECT nome FROM cidade WHERE uf = ?';
    $stm = $pdo->prepare($sql);
    $stm->bindValue(1, $estado);
    $stm->execute();
    sleep(1);
    echo json_encode($stm->fetchAll(PDO::FETCH_ASSOC));
    $pdo = null;
}
?>

```

Observações: Foi utilizada a função "sleep()" somente para atrasar o retorno da função em 1 segundo, assim será exibido na tela uma mensagem de aviso para usuário. Não é aconselhável usar esta prática para o desenvolvimento, uma vez que o próprio processamento na WEB já acarreta certo atraso no retorno das informações.

estilo.css

Para deixar o visual da página "index.php" mais agradável, será utilizado CSS. Para o leitor que se interessar, no portal DevMedia existem vários cursos sobre o assunto, pois o mesmo não será aprofundado neste artigo.

Listagem 4: Folha de estilo CSS

```

*{ margin: 0; padding: 0; font-family: Verdana, Geneva, sans-serif;}

p{ text-align: center;}

select{ cursor: pointer; }

label{ color: #000000; font-weight: bold; font-size: 15px; line-height: 40px; margin: 20px;}

.destaque{ color: #900; font-weight: bold; font-size: 20px; line-height: 50px; }

.botao{ width: 100px; height: 25px; border-radius: 10px; border: 2px solid #900; cursor: pointer; }

.mensagem{ color: #333; font-size: 15px; font-weight: bold; margin: 10px;}

#conteudo{ width: 900px; height: 300px; margin: 10px auto; border-radius: 10px; box-shadow: 5px 5px 5px

```

index.php

Para finalizar, vamos agregar todos os scripts nesse arquivo, pois será ele a interface final com o usuário. Pelo fato do arquivo ser um pouco extenso, serão abordadas somente as partes de interesse para o funcionamento do exemplo.

Listagem 5: Linkando CSS e jQuery

```
<link rel="stylesheet" type="text/css" href="css/estilo.css">
<script type="text/javascript" src="js/jquery-1.8.2.js"></script>
```

Listagem 6: Códigos HTML do corpo da página

```
<body>
  <div id="conteudo">
    <p><span class="destaque">Populando selects usando AJAX + JSON + PHP</span></p>
    <hr />
    <br/>

    <div id="pais">
      <label>Selecione o Pais:</label>
      <select id="cmbPais">
        <option>Carregar Países</option>
      </select>
      <input type="button" value="Carregar Pais" id="btnPais" class="botao"/>
    </div>

    <div id="estado">
      <label>Selecione o Estado:</label>
      <select id="cmbEstado">
        <option>Carregar Estados</option>
      </select>
    </div>

    <div id="cidade">
      <label>Selecione a Cidade:</label>
      <select id="cmbCidade">
        <option>Carregar Cidades</option>
      </select>
    </div>

    <hr />
    <p><span class="destaque">Mensagens:</span></p>
    <div id="mensagem">

    </div>
  </div>
</body>
```

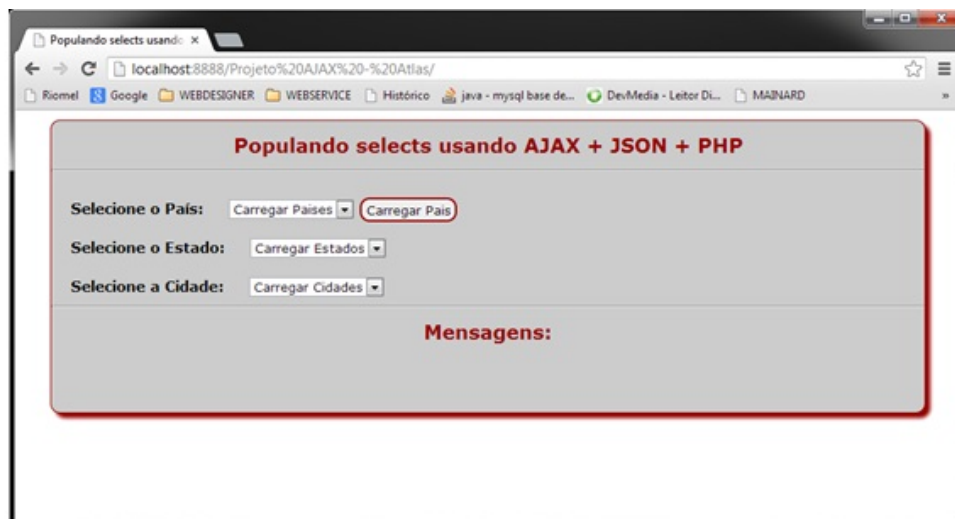


Figura 2: Layout da página index.php

Listagem 7: Códigos JavaScript para requisições

```
<script type="text/javascript">
$(document).ready(function() {

    <!-- Carrega os Países -->
    $('#btnPais').click(function(e) {
        $('#btnPais').hide();
        $('#mensagem').html('<span class="mensagem">Aguarde, carregando ...</span>');

        $.getJSON('consulta.php?opcao=pais', function (dados) {

            if (dados.length > 0) {
                var option = '<option>Selecione o País</option>';
                $.each(dados, function(i, obj) {
                    option += '<option value="'+obj.sigla+'">'+obj.nome+'</option>';
                })
                $('#mensagem').html('<span class="mensagem">Total de paises encontrados.: '+d
                $('#cmbPais').html(option).show();
            } else {
                Reset();
                $('#mensagem').html('<span class="mensagem">Não foram encontrados paises!</s
            }
        })
    })

    <!-- Carrega os Estados -->
    $('#cmbPais').change(function(e) {
        var pais = $('#cmbPais').val();
        $('#mensagem').html('<span class="mensagem">Aguarde, carregando ...</span>');

        $.getJSON('consulta.php?opcao=estado&valor='+pais, function (dados) {

            if (dados.length > 0) {
                var option = '<option>Selecione o Estado</option>';
                $.each(dados, function(i, obj) {
                    option += '<option value="'+obj.sigla+'">'+obj.nome+'</option>';
                })
                $('#mensagem').html('<span class="mensagem">Total de estados encontrados.: '+
            } else {
                Reset();
                $('#mensagem').html('<span class="mensagem">Não foram encontrados estados par
```

```

        }
        $('#cmbEstado').html(option).show();
    })

    <!-- Carrega as Cidades -->
    $('#cmbEstado').change(function(e){
        var estado = $('#cmbEstado').val();
        $('#mensagem').html('<span class="mensagem">Aguarde, carregando ...</span>');

        $.getJSON('consulta.php?opcao=cidade&valor='+estado, function (dados){

            if (dados.length > 0){
                var option = '<option>Selecione a Cidade</option>';
                $.each(dados, function(i, obj){
                    option += '<option>'+obj.nome+'</option>';
                })
                $('#mensagem').html('<span class="mensagem">Total de cidades encontrada
            }else{
                Reset();
                $('#mensagem').html('<span class="mensagem">Não foram encontradas cidad
            }
            $('#cmbCidade').html(option).show();
        })
    })

    <!-- Resetar Selects -->
    function Reset(){
        $('#cmbPais').empty().append('<option>Carregar Países</option>>');
        $('#cmbEstado').empty().append('<option>Carregar Estados</option>>');
        $('#cmbCidade').empty().append('<option>Carregar Cidades</option>');
    }
});
</script>

```

Como todas as chamadas são iguais, é interessante explicar alguns pontos importantes desse código JavaScript para que todas sejam entendidas.

Observem que para carregar os países, existe um button com texto "Carregar Pais". No evento Click desse botão será disparada a requisição AJAX no formato JSON. Antes dessa ação, é ocultado o button e exibida uma mensagem para o usuário "Aguarde, carregando ...".

Em seguida, é executada a requisição "\$.getJSON(URL, function(retorno))" que recebe 2 parâmetros:

- URL para onde será enviada a requisição, opcionalmente pode-se adicionar parâmetros nesse URL, com será o nosso caso;
- Uma função callback que recebe como parâmetro os dados recebidos da requisição, isso se a mesma for bem sucedida.

Observação: Essa requisição "\$.getJSON()" é uma forma abreviada da requisição "\$.ajax()".

O próximo passo é tratar as informações que retornaram do servidor dentro da função callback. Primeiro verificando se existem dados, caso existam, através de um laço de iteração carrega os dados criando "<option>" dinamicamente para o select. Ao final do laço, exibe uma mensagem com a quantidade de registros retornados do servidor e inclui os dados que foram formatados dinamicamente dentro do laço de iteração.

Caso não seja retornado nenhum dado pelo servidor, é exibida uma mensagem informando que "Não foram encontrados países!" e executada a função "Reset()", essa função tem a finalidade de limpar todas as options dos selects e adicionar somente a opção inicial, nesse caso, "Carregar Países".

Observem que as outras 2 chamadas são exatamente iguais, alterando somente a forma como são disparadas e os parâmetros passados:

1. Carrega todos os países no evento "Click" do button;
2. Carrega os estados a partir do evento "Change" do select "cmbPais", onde conforme o país selecionado, este será enviado como parâmetro para instrução SQL da função "getFilterEstado()".
3. Carrega as cidades a partir do evento "Change" do select "cmbEstado", onde conforme o estado selecionado, este será enviado como parâmetro para instrução SQL da função "getFilterCidade()".

Agora podemos testar os filtros, para iniciar basta pressionar o botão "Carregar País".



Figura 3: Página inicial sem filtros ativos

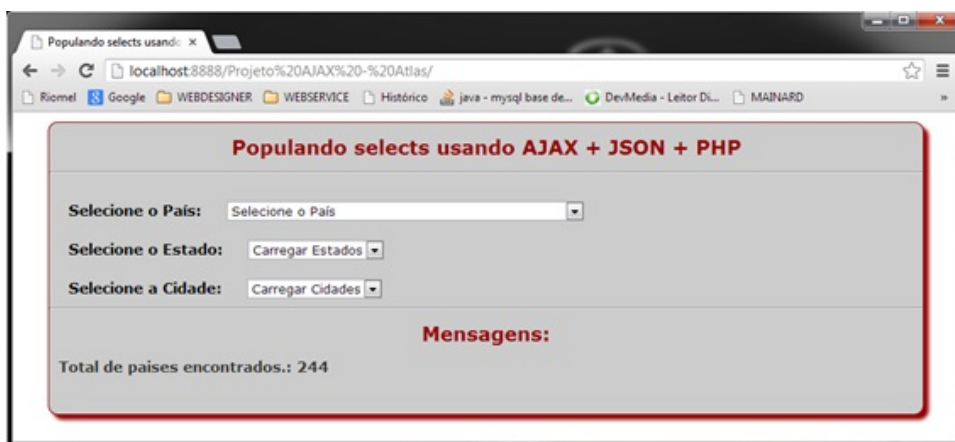


Figura 4: Select "cmbPais" com países carregados



Figura 5: País selecionado e select "cmbEstado" carregado

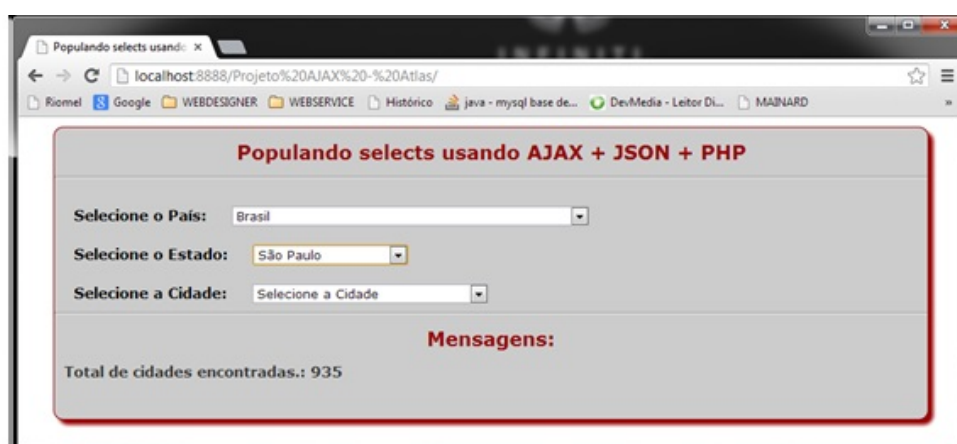


Figura 6: Estado selecionado e select "cmbCidade" carregado

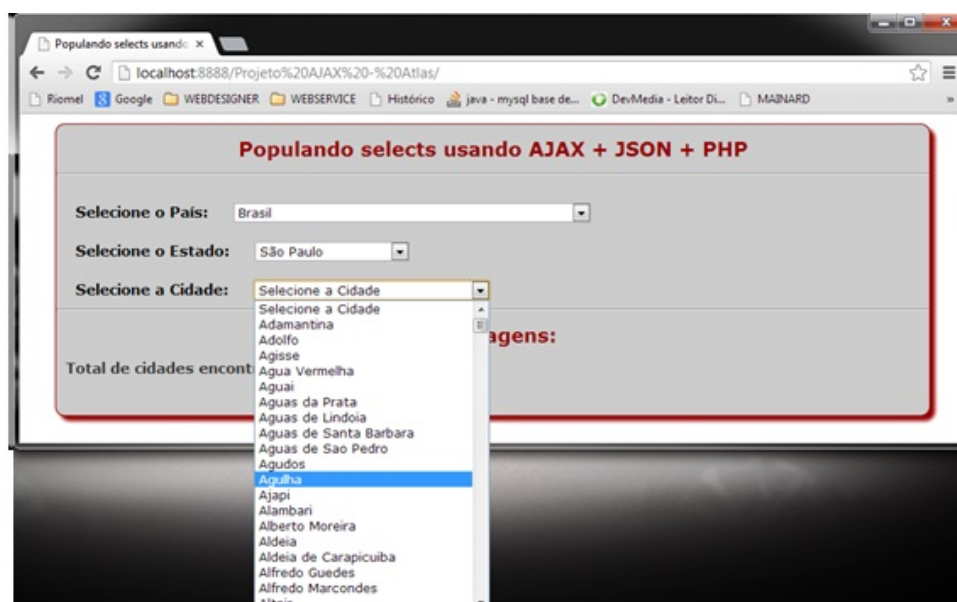


Figura 7: Todas as cidades do estado de São Paulo

Bom, pessoal, neste artigo foi demonstrado um exemplo simples de como executar requisições AJAX para o formato JSON. Esse tipo de requisição pode tornar a experiência do usuário com a página mais interessante e agradável, pois como foi visto, não existe o típico refresh para carregar os dados nos devidos selects. Existem outras variações desse tipo de requisição, no site oficial do jQuery existe uma vasta documentação sobre essa e diversas outras funções.

Foi possível observar também que, sabendo trabalhar com os eventos do select, pode-se deixar os filtros bem dinâmicos.

Links

- [Download WampServer](#)
- [Download jQuery](#)
- [Download Base de dados com países do mundo](#)

Caso surja alguma dúvida, meu e-mail é wllfl@ig.com.br. Fiquem à vontade também para usar a seção de comentários.

Abraço a todos!



William