

# USANDO JSON

Na maioria das vezes, transmitir apenas uma frase ou um número não é o suficiente para que nosso código Java Script se comunique com o servidor. Nesses casos, precisamos enviar e receber listas, valores e objetos complexos contendo toda a informação necessária. Existem inúmeras soluções para isso, sendo que as mais comuns são **XML** e **JSON**.




**JSON** é uma forma de serializar, ou converter para texto, objetos do Java Script de modo que sejam legíveis inclusive por humanos. A grande vantagem do JSON sobre qualquer outra forma de serialização é que o Java Script lida com o formato de modo transparente, convertendo a informação em formato texto para um tipo nativo da linguagem, evitando o uso de bibliotecas e ferramentas que possam complicar o seu código. Você pode combinar livremente objetos e listas com os seis tipos de dados válidos, representando valores complexos que contêm toda a informação necessária para a sua requisição. Para demonstrar que cara tem o JSON, vamos usar os seguintes dados para que você entenda melhor como funciona:

```
{  
  "nome": "Plínio Jose",  
  "Endereco": "Rua consolação 80",  
  "anonascimento" : 1979,  
  "telefones": [  
    "3522 – 0001",  
    "3521 – 0011",  
  ]  
}
```

Se olharmos com mais atenção ao conteúdo do nosso código JSON, vamos ver que seu conteúdo nada mais é do que um dicionário, contendo Strings, valores numéricos e lógicos, ou booleanos. Não por coincidência, o formato JSON é um código Java Script válido, podendo ser atribuído a uma variável e tratado como um objeto qualquer. Dessa forma, podemos acessar cada valor usando a chave como uma propriedade de um objeto Java Script.

Para demonstra usaremos JSON para consumir um servidor que retorna uma estrutura JSON com os dados do CEP informado, voltando os dados de logradouro, cidade, bairro e estado. Inicie codificando a seguinte pasta e sua estrutura:

te Computador > Disco Local (C:) > xampp > htdocs > exemploJSON

Nome	Data de modificaç...	Tipo	Tamanho
 estilos	11/08/2015 19:28	Documento de fol...	1 KB
 index	11/08/2015 19:25	Chrome HTML Do...	1 KB
 jquery-1.11.3	31/07/2015 11:36	Arquivo JavaScript	278 KB

Abra no VISUAL CODE e crie o seguinte formulário em index:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <script src="jquery-1.11.3.js"></script>
    <link href="estilos.css" rel="stylesheet" type="text/css">
    <script>
      $(document).ready( function(){

        });
    </script>
  </head>
```

```

  <body>
    <div id = "formulario">
      <form action="" method="GET">
        <fieldset>
          <h1 align=center>CADASTRO - ENDEREÇO DO CLIENTE</h1>
          <p>NOME DO CLIENTE: <br>
          <input type="text" id="nome" size = 45>
          <p>CEP: <br>
          <input type="text" id="CEP" value="0">
          <input type="button" id="btBuscar" value="buscar CEP">
          <p>Endereço: <br>
          <input type="text" id="endereco" size = 80 >
          <p>Número: <br>
          <input type="text" id="numero" size = 80 >
          <p>Bairro: <br>
          <input type="text" id="bairro" size = 80>
          <p>Cidade: <br>
          <input type="text" id="cidade" size = 80>
          <p>Estado:<br>
          <input type="text" id="estado" size = 2 >
          <p>
          <input type="submit" value = "cadastrar">
        </form>
      </fieldset>
    </div>
  </body>
</html>
```

Em estilos.css copie o código abaixo para a aparência do formulário:

```
#formulario{
  width: 640px;
  height:480px;
  margin: 0 auto;
}
p{
  margin: 5px 0 15px;
```

```
    font-size: 18px;
    line-height: 34px;
    padding: 10px;
    font-family: Calibri
}
h1{
    margin: 5px 0 15px;
    font-size: 18px;
    line-height: 34px;
    padding: 10px;
    font-family: Calibri;
    background-color: #666;
    color: white;
}
fieldset{
    padding: 4px;
    margin: 0 auto;
    border: 2px solid saddlebrown;
}
input[type="text"]{
    background: none repeat scroll 0 0 #FFFFFF;
    border: 1px solid #C9C9C9;
    box-shadow: 0 1px 3px rgba(0, 0, 0, 0.15) inset, -5px -5px 0 0 #F5F5F6,
        5px 5px 0 0 #F5F5F6, 5px 0 0 0 #F5F5F6, 0 5px 0 0 #F5F5F6, 5px -5px 0 0 #F5F5F6,
-5px 5px 0 0 #F5F5F6;
    color: #545658;
    padding: 8px;
    font-size: 14px;
    border-radius: 2px 2px 2px 2px;
}
input[type="button"]{
    background: none repeat scroll 0 0 #FFFFFF;
    border: 1px solid #C9C9C9;
    box-shadow: 0 1px 3px rgba(0, 0, 0, 0.15) inset, -5px -5px 0 0 #F5F5F6,
        5px 5px 0 0 #F5F5F6, 5px 0 0 0 #F5F5F6, 0 5px 0 0 #F5F5F6, 5px -5px 0 0 #F5F5F6,
-5px 5px 0 0 #F5F5F6;
    color: #545658;
    padding: 8px;
    font-size: 14px;
    border-radius: 2px 2px 2px 2px;
```

```
}  
input[type="submit"]{  
border: padding-top: padding-bottom: 5px;  
padding-right: 10px;  
padding-left: 10px;  
background-color: #666;  
    color: white;  
    min-width: margin-top: -4px;  
    margin-bottom: 4px;  
    background: #ed5050;  
    box-shadow: 0px 4px 0px 0px #666, 0px 7px 6px 0px rgba(0,0,0,.25);  
}
```

O resultado visual será:

CADASTRO - ENDEREÇO DO CLIENTE

NOME DO CLIENTE:

CEP:

0

buscar CEP

Endereço:

Número:

Bairro:

Cidade:

Estado:

cadastrar

Usar AJAX com JSON é algo muito simples com jQuery. Vamos usar um site de consulta de CEP chamado Postmon para demonstrar. Ao informar um CEP, o serviço vai retornar um JSON com os dados do logradouro se o código for válido, ou um erro se o CEP não for encontrado na base dos Correios. Codifique o seguinte lógica de programação no bloco JQUERY em index.html:

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <script src="jquery-1.11.3.js"></script>

  <link href="estilos.css" rel="stylesheet" type="text/css">
  <script>

    function onCepDone(data) {
      // preenche os campos do formulário
      // vaçl atualiza os campos de formulário
      $("#endereco").val(data.logradouro);
      $("#bairro").val(data.bairro);
      $("#cidade").val(data.cidade);
      $("#estado").val(data.estado);
    }

    $(document).ready( function(){
      // pressionou o botão buscar?
      $("#btBuscar").click( function(){
        // endereço do serviço
        var servico = "http://api.postmon.com.br/v1/cep/";
        // cep a ser procurado
        var cep = $("#CEP").val();
        // conecta-se no servidor JSON
        $.getJSON(servico + cep)
          .done(onCepDone); // se bem sucedido executa a função em done
      });
    });

  </script>
</head>
```

Perceba que a variável data, que contém o retorno do serviço, já está convertido em um objeto Java Script. Com isso você pode acessar qualquer informação retornada, como bairro ou cidade, como se uma propriedade do objeto. Usamos a propriedade e Val do JQUERY para modificar os valores do formulário para os campos retornados. O JQUERY tem 3 funções para modificar o conteúdo de um campo id com JQUERY:

<b>text()</b>	Usado com componentes de elementos selecionados, como textos de parágrafos e colunas de tabelas
<b>html()</b>	Usamos quando desejamos manipular elementos com tag HTML

val()

Usado normalmente com valores de campos de formulários

Teste o formulário:

## CADASTRO - ENDEREÇO DO CLIENTE

NOME DO CLIENTE:

CEP:

18400640

buscar CEP

Endereço:

Rua Professor João Santana

Número:

Bairro:

Vila Bom Jesus

Cidade:

Itapeva

Estado:

SP

cadastar

Podemos incluir também uma rotina em caso de falha na obtenção do CEP, modifique o seu código de acordo com as linhas marcadas abaixo.

```
function onCepDone(data) {  
    // preenche os campos do formulário  
    // val atualiza os campos de formulário  
    $("#endereco").val(data.logradouro);  
    $("#bairro").val(data.bairro);  
    $("#cidade").val(data.cidade);  
    $("#estado").val(data.estado);  
}  
  
function onCepError(error) {  
    // limpa os campos do formulário se erro  
    $("#endereco").val("");  
    $("#bairro").val("");  
    $("#cidade").val("");  
    $("#estado").val("");  
    alert("CEP inválido:");  
}  
  
$(document).ready( function(){  
    // pressionou o botão buscar?  
    $("#btBuscar").click( function(){  
        // endereço do serviço  
        var servico = "http://api.postmon.com.br/v1/cep/";  
        // cep a ser procurado  
        var cep = $("#CEP").val();  
        // conecta-se no servidor JSON  
        $.getJSON(servico + cep)  
        .done(onCepDone) // se bem sucedido executa a função em done  
        .fail(onCepError); // se a conexão não é bem sucedida executa a rotina de erro  
    });  
});
```

O resultado será:

