

Ajax

Programação para Internet I
IFB - Tecnologia em Sistemas para Internet

Marx Gomes van der Linden

AJAX

- Sigla para “Asynchronous JavaScript and XML”
 - » Atualmente JSON substituiu XML na maior parte das aplicações
- Enviar e receber dados por mensagens HTTP sem recarregar a página

Requisição AJAX simples

- **load(url)**

- » Carrega o conteúdo da URL fornecida e o coloca no elemento que está chamando o método

```
$('#aqui').load('http://www.exemplo.com.br/meutexto');
```

Requisição com callback

- **load(url, funcao)**

- » Carrega o conteúdo da URL fornecida e o coloca no elemento que está chamando o método
- » Quando a requisição é finalizada, a função passada é chamada

```
$( '#aqui' ).load(  
    'http://www.exemplo.com.br/api/blablabla',  
    function() {  
        // conteúdo da função callback  
    }  
);
```

Web services

- **Web services** são tecnologias usadas para fornecer dados em formatos padronizados através da web
 - » Dados podem ser consumidos e interpretados por aplicações construídas em qualquer linguagem
- Utilizam o protocolo HTTP
 - » O mesmo usado pelo navegador
- API: Application Programming Interface
 - » Padrão de URLs usada para acessar os serviços

Protocolo HTTP

- Toda requisição HTTP começa com um verbo:
 - » **GET** → Obter dados (padrão)
 - » **POST** → Enviar novos dados
 - » **PUT** → Atualizar dados existentes
 - » **DELETE** → Excluir dados
 - » **PATCH** → Atualizar dados parcialmente
- As APIs devem especificar qual verbo deve ser usado com que requisição

Requisições HTTP com jQuery

– `$.ajax(obj)`

- » Realiza uma requisição HTTP
- » `obj` é um objeto com os seguintes campos:
 - › **type**: verbo HTTP (padrão: GET)
 - › **url**: URL a ser requisitada
 - › **data**: dados a serem enviados
 - › **success**: função a ser executada em caso de sucesso
 - › **error**: função a ser executada em caso de falha
 - › **dataType**: tipo de dado que vai ser retornado: `'text'`, `'html'`, `'xml'`, `'json'`, `'jsonp'`, ou `'script'`.

Exemplo

```
$.ajax({  
  type: 'GET',  
  url: 'http://www.exemplo.com.br/api/blabla',  
  success: function(dados) {  
    // Faz alguma coisa com dados  
  },  
  error: function(dados) {  
    // Faz alguma coisa com dados  
  }  
});
```


Métodos de atalho

- Para os dois verbos HTTP mais comuns, jQuery fornece os métodos equivalentes:
- **\$.get(url, função) / \$.get(obj)**
- **\$.post(url, função) / \$.post(obj)**

```
$.get(  
    'http://www.exemplo.com.br/api/blabla',  
    function(dados) {  
        // Faz alguma coisa com dados  
    }  
);
```

JSON

- Javascript Object Notation
- Padrão universal para transferência de dados em aplicações web
 - » Usado não apenas em Javascript
- Similar a um objeto Javascript convencional, mas:
 - » Aspas duplas sempre!
 - » Não pode ter funções

Javascript vs JSON

```
{  
  nome: 'Tyrannosaurus rex',  
  periodo: "Cretáceo",  
  peso: 14000,  
  
  especimens:  
    [ 15, 13.2, 12.3],  
  
  ref: {  
    id: 1245,  
    cod: 'trex'  
  }  
}
```

Javascript OK

X JSON Inválido

```
{  
  "nome": "Tyrannosaurus rex",  
  "periodo": "Cretáceo",  
  "peso": 14000,  
  
  "especimens":  
    [ 15, 13.2, 12.3 ],  
  
  "ref": {  
    "id": 1245,  
    "cod": "trex"  
  }  
}
```

Javascript OK

JSON OK

Interpretando uma string JSON

- O método nativo **JSON.parse(str)** recebe uma string contendo JSON e devolve um objeto Javascript

```
let str = '{"nome":"Tyrannosaurus  
rex","periodo":"Cretáceo","peso":14000,"especimens":  
[15,13.2,12.3],"ref":{"id":1245,"cod":"trex"}}';  
  
let obj = JSON.parse(str);  
  
console.log(obj.nome);  
console.log(obj.ref.id);  
for(let s of obj.especimens){  
    console.log(s);  
}
```

Objendo e interpretando JSON

- Os métodos de requisição do jQuery automaticamente detectam e interpretam JSON, quando corretamente marcado pelo servidor
 - » Para garantir, basta especificar **'json'** no parâmetro **dataType** do objeto da requisição

Exemplo

```
$.ajax({  
  type: 'GET',  
  url: 'http://www.exemplo.com.br/api/blabla',  
  success: function(dados) {  
    // dados será um objeto interpretado a  
    // partir do JSON retornado pelo servidor  
  },  
  dataType: 'json'  
});
```

Enviando dados JSON

- Para enviar dados JSON junto com a requisição, use o parâmetro data no objeto da requisição:

```
$.ajax({  
  type: 'POST',  
  url: 'http://www.exemplo.com.br/api/blabla',  
  data: {  
    // informação enviada para o servidor  
  },  
  success: function(dados) {  
    // dados será um objeto interpretado a  
    // partir do JSON retornado pelo servidor  
  },  
  dataType: 'json'  
});
```

Códigos HTTP

– Toda resposta HTTP vem com um **código de status**

- **1xx** → Informação
- **2xx** → Sucesso
 - » **200**: OK
 - » **204**: Sem conteúdo
- **3xx** → Redirecionamento
- **4xx** → Erro no cliente
 - » **403**: Acesso negado
 - » **404**: Não encontrado
- **5xx** → Erro no servidor

Obtendo o código HTTP

- A função de sucesso do objeto tem o formato completo:

```
success: function(dados, textost, xhr){  
    ...  
},
```

- Onde:
 - » **textost** → String descritiva do status (por exemplo, 'success')
 - » **xhr** → objeto com várias propriedades sobre a requisição
 - › **xhr.status** → Código de status HTTP