

به نام خدا

برنامه نویسی مبتنی بر وب



Ajax

Internet & World Wide Web HOW TO PROGRAM



DEITEL®

FIFTH EDITION

ASEMAN SOFT
PAUL DEITEL
HARVEY DEITEL
ABBAY DEITEL

HTML5: Input and Page Structure Elements
Autocomplete • Self Validation • Audio • Video
Web Storage • WebSockets • Web Workers

BROWSERS: Desktop and Mobile

CSS3: Selectors • Shadows • Image Borders
Web Fonts • Rounded Corners • Gradients
Colors • Background Images • Animation
Transitions • Transformations • Flexboxes
Media Queries • Multicolumn Layouts

ECMAScript 5 (JAVASCRIPT):

DOM • XML • JQuery • JSON • AJAX

HTML5 CANVAS: Lines/Shapes/Arcs • Colors
Gradients • Shadows • Image Manipulation
Patterns • Quadratic/Bezier Curves • Games
Compositing • Transformations • Animations

SERVER SIDE: Web Servers • Database
MySQL® • PHP • JSF • ASP.NET (C# and VB)
Web Services (Java™, C# and VB)

Traditional Web Applications

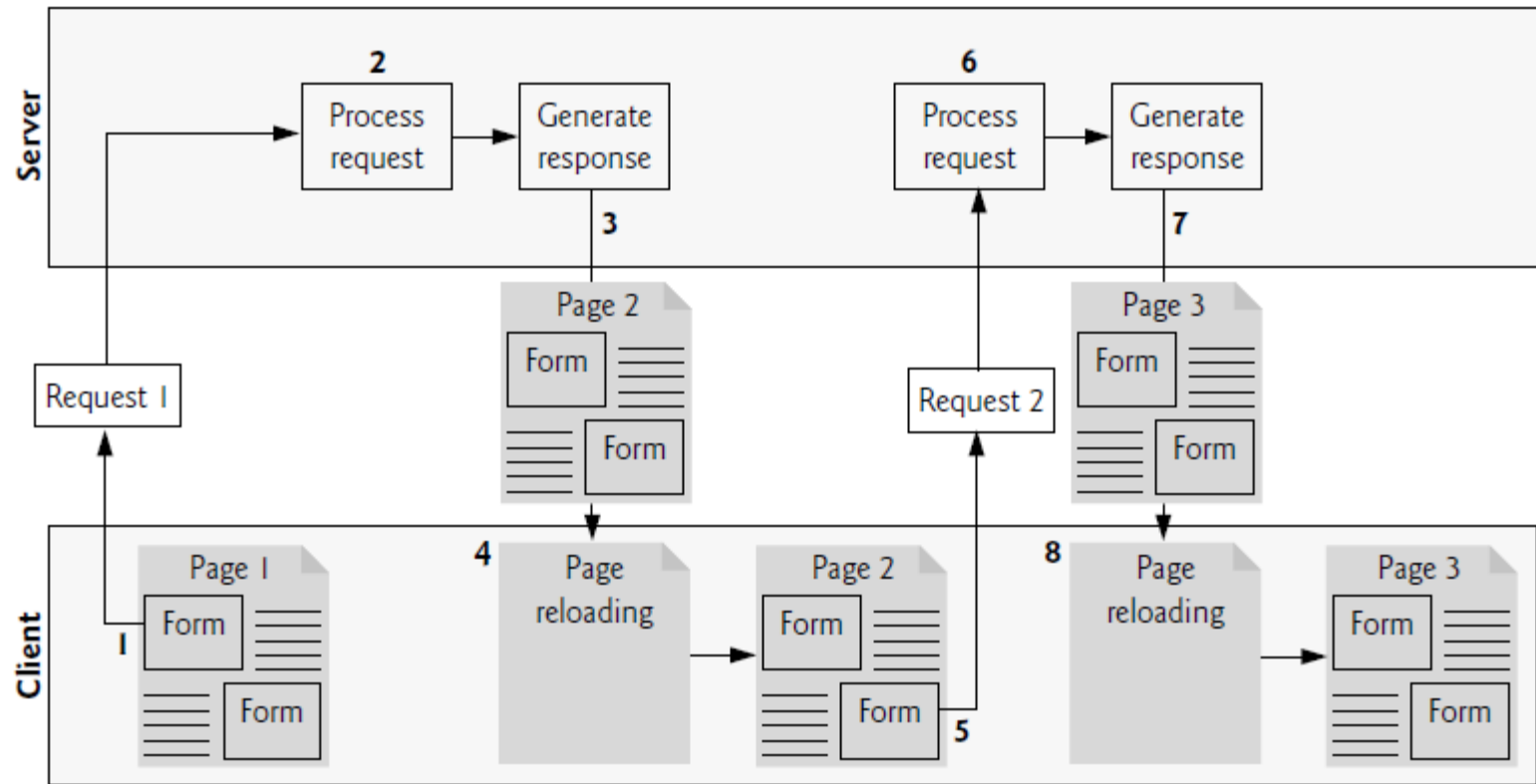


Fig. 16.1 | Classic web application reloading the page for every user interaction.

Ajax

- Ajax رویای برنامه نویسان است، زیرا به وسیله ی آن می توانید:
 - ✓ یک صفحه ی وب را بروز رسانی کنید بدون اینکه نیاز باشد آن صفحه را لود کنید.
 - ✓ بعد از لود شدن صفحه، داده ها را از سرور درخواست کنید.
 - ✓ بعد از لود شدن صفحه، داده ها را از سرور دریافت کنید.
 - ✓ در پشت پرده، داده ها را به سرور ارسال کنید.

AJAX is a developer's dream, because you can:

- Read data from a web server - after the page has loaded
- Update a web page without reloading the page
- Send data to a web server - in the background

AJAX چیست؟

- **AJAX** سرنام واژگان **Asynchronous JavaScript and XML** می باشد که به معنی جاوااسکریپت و xml غیر همزمان می باشد. کلمه ی **AJAX** کمی گمراه کننده است. شما برای استفاده از **AJAX** نیازی به یادگیری **XML** ندارید. سایت های بزرگی مثل، **Google Maps** و **Gmail** و **YouTube** و **Facebook** از **ajax** استفاده می کنند. **Ajax** به وسیله ی کلمات پیشنهاد گوگل (**Google Suggest**)، در سال ۲۰۰۵ محبوب شناخته شد.

Ajax Web Applications

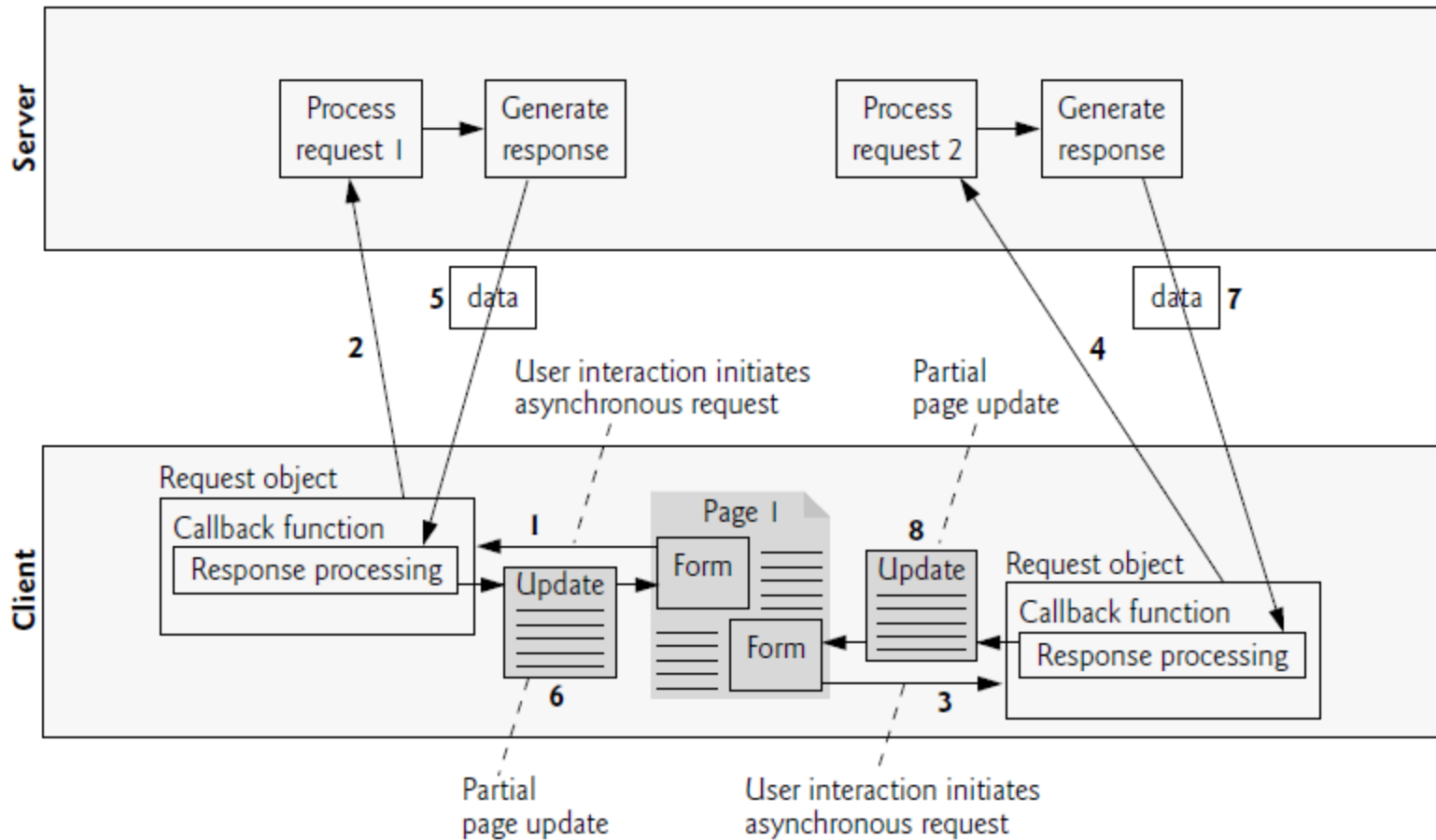
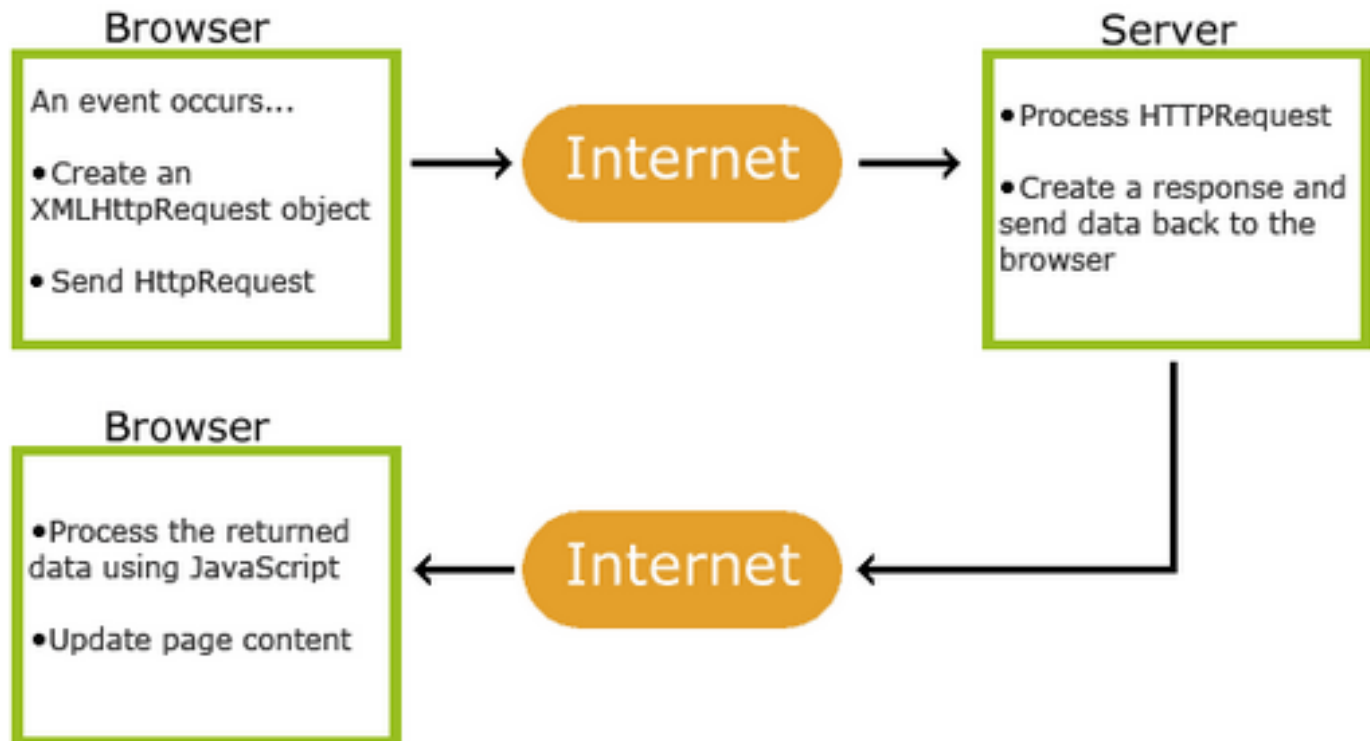


Fig. 16.2 | Ajax-enabled web application interacting with the server asynchronously.

نحوه عملکرد Ajax

How AJAX Works



XMLHttpRequest

- تمام مرورگرهای مدرن، از شیء `XMLHttpRequest` پشتیبانی می کنند. از شیء `XMLHttpRequest` برای تبادل داده با یک سرور از پشت پرده، استفاده می شود.

ایجاد یک شیء XMLHttpRequest

```
variable = new XMLHttpRequest();
```

Example

```
var xhttp = new XMLHttpRequest();
```

Access Across Domains

- For security reasons, modern browsers do not allow access across domains.
- This means that both the web page and the XML file it tries to load, must be located on the **same server**.

متدهای شی XMLHttpRequest

XMLHttpRequest Object Methods

Method	Description
<code>new XMLHttpRequest()</code>	Creates a new XMLHttpRequest object
<code>abort()</code>	Cancels the current request
<code>getAllResponseHeaders()</code>	Returns header information
<code>getResponseHeader()</code>	Returns specific header information
<code>open(method, url, async, user, psw)</code>	Specifies the request <i>method</i> : the request type GET or POST <i>url</i> : the file location <i>async</i> : true (asynchronous) or false (synchronous) <i>user</i> : optional user name <i>psw</i> : optional password
<code>send()</code>	Sends the request to the server Used for GET requests
<code>send(string)</code>	Sends the request to the server. Used for POST requests
<code>setRequestHeader()</code>	Adds a label/value pair to the header to be sent

خاصیت های XMLHttpRequest

XMLHttpRequest Object Properties

Property	Description
onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
responseText	Returns the response data as a string
responseXML	Returns the response data as XML data
status	Returns the status-number of a request 200: "OK" 403: "Forbidden" 404: "Not Found" For a complete list go to the Http Messages Reference
statusText	Returns the status-text (e.g. "OK" or "Not Found")

ارسال یک درخواست به سرور

```
xhttp.open("GET", "ajax_info.txt", true);  
xhttp.send();
```

متد	توضیحات
<code>open(method, url, async)</code>	نوع درخواست را مشخص می کند method: نوع درخواست، که می تواند GET یا POST باشد. url: مکان (فایل) سرور را مشخص می کند. async: true (غیرهمزمان) یا false (همزمان).
<code>send()</code>	درخواست را به سرور ارسال می کند (برای GET از آن استفاده می کنند)
<code>send(string)</code>	درخواست را به سرور ارسال می کند (برای POST از آن استفاده می کنند)

GET یا POST؟

- متد **GET** ساده تر و پرسرعت تر از متد **POST** است و می تواند در بیشتر موارد بکار برده شود.
- اما در مواقع زیر شما باید همواره از متد **POST** استفاده کنید:
 - ✓ آپدیت کردن یک فایل یا پایگاه داده در سرور.
 - ✓ ارسال داده های حجیم به سرور (متد **POST** دارای محدودیت اندازه نیست).
 - ✓ ارسال ورودی های کاربر (که ممکن است دارای کاراکترهای نامعلومی باشد). زیرا متد **POST** قوی تر و امن تر از متد **GET** است.

Example

```
xhttp.open("GET", "demo_get.asp", true);  
xhttp.send();
```

نکته

- در مثال زیر ممکن است شما یک نتیجه ی ذخیره شده در کش را مشاهده کنید، برای اجتناب از این کار، یک ID یکتا به URL اضافه می کنیم:

Example

```
xhttp.open("GET", "demo_get.asp", true);  
xhttp.send();
```

Example

```
xhttp.open("GET", "demo_get.asp?t=" + Math.random(), true);  
xhttp.send();
```

نکته

- اگر می خواهید که اطلاعاتی را به وسیله ی متد GET ارسال کنید، اطلاعات را به URL اضافه کنید:

Example

```
xhttp.open("GET", "demo_get2.asp?fname=Henry&lname=Ford", true);  
xhttp.send();
```


پارامتر url

- پارامتر url فایلی است که در سرور قرار دارد. این فایل می تواند از هر نوعی باشد. مانند txt. یا xml. یا فایل های اسکریپتی مانند asp. یا php. که می توانند قبل از برگرداندن پاسخ از سرور، اعمالی را انجام دهند.

```
xhttp.open("GET", "ajax_info.txt", true);
```

```
xhttp.open("GET", "ajax_test.asp", true);
```

غیر همزمان - True or False? Asynchronous

- آجاکس بر پایه ی جاوااسکریپت و XML غیر همزمان می باشد و برای اینکه شیء XMLHttpRequest بصورت آجاکسی عمل کند، پارامتر async از متد open () باید برابر با true باشد.

```
xhttp.open("GET", "ajax_test.asp", true);
```

مزایای غیر همزمانی

- ارسال درخواست ها به صورت غیر همزمان، برای برنامه نویسان وب بسیار مفید واقع شده است. بسیاری از کارهای ارائه شده در سرور، زمان بر هستند. بنا براین قبل از ajax این عملکرد می توانست باعث هنگ کردن یا متوقف شدن اپلیکیشن شود.
- به وسیله ی ajax، جاوااسکریپت نیاز ندارد تا برای پاسخ دادن سرور، انتظار بکشد، اما به جای آن می تواند:
 - ✓ در هنگام انتظار برای پاسخ سرور، دیگر اسکریپت ها را اجرا کند.
 - ✓ پرداختن به درخواست، هنگامی که درخواست آماده شده است.

Synchronous XMLHttpRequest (async = false) is not recommended because the JavaScript will stop executing until the server response is ready. If the server is busy or slow, the application will hang or stop.

Synchronous XMLHttpRequest is in the process of being removed from the web standard, but this process can take many years.

Modern developer tools are encouraged to warn about using synchronous requests and may throw an InvalidAccessError exception when it occurs.

onreadystatechange

- هنگامی که یک درخواست به سرور ارسال می شود، ما می خواهیم تا بر اساس پاسخ مورد نظر، برخی اقدامات انجام شوند.
- ✓ هر زمان که `readyState` تغییر می کند، رویداد `onreadystatechange` روی می دهد.
- ✓ خصوصیت `readyState` در واقع وضعیت `XMLHttpRequest` را در خود نگه داری می کند.
- رویداد `onreadystatechange` پنج بار روی می دهد (موارد ۰ تا ۴)، هر بار برای یک تغییر در `readyState`.
- هنگامی که `readyState` برابر با ۴ است و `status` برابر با ۲۰۰ است، پاسخ آماده است.

onreadystatechange

Property	Description
onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
status	200: "OK" 403: "Forbidden" 404: "Page not found" For a complete list go to the Http Messages Reference
statusText	Returns the status-text (e.g. "OK" or "Not Found")

```
<body>

<div id="demo">
<h2>The XMLHttpRequest Object</h2>
<button type="button" onclick="loadDoc()">Change Content</button>
</div>
```

Example

```
function loadDoc() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("demo").innerHTML =
                this.responseText;
        }
    };
    xhttp.open("GET", "ajax_info.txt", true);
    xhttp.send();
}
```

ادامه دارد...