

AJAX

Asynchronous JAvascript and XML

Guy Lapalme
IFT3225

Ajax

- Proposé (fév 05) par Jesse James Garrett
<http://www.adaptivepath.com>
- Applications web plus *réactives* par un réaffichage sélectif d'une portion de page HTML

Ajax



- Présentation avec XHTML et CSS
- Affichage dynamique et interaction avec modèle DOM de la page
- Échange et manipulation de données avec XML ou autre (texte, HTML ou JSON)
- Requête asynchrone de données au serveur via XMLHttpRequest
- Javascript qui lie tout cela

Pour accéder à un champ de la page il faut utiliser DOM pour manipuler par après avec AJAX

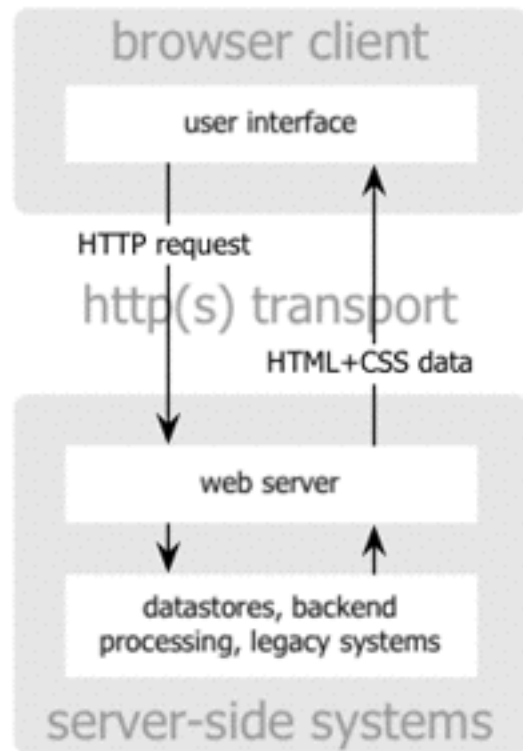
Utilisations

- Validation en temps réel via le serveur avant la soumission finale
 - ─ noms d'usagers, code postaux
- Complétion d'entrée

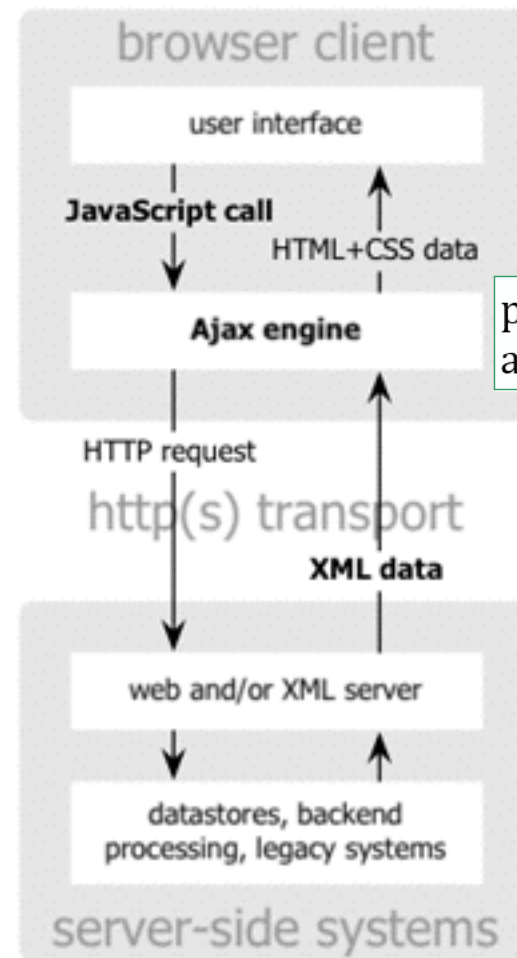
la proposition sur le champ de recherche
- Contrôles plus complexes
 - ─ affichage d'arbres, barres d'avancement, menus...
- Mise à jour de données en temps réel
- Envoi direct du serveur vers une page

Exemples commerciaux

- Google Suggest
- Google Maps
- Flickr
- A9.com



classic
web application model

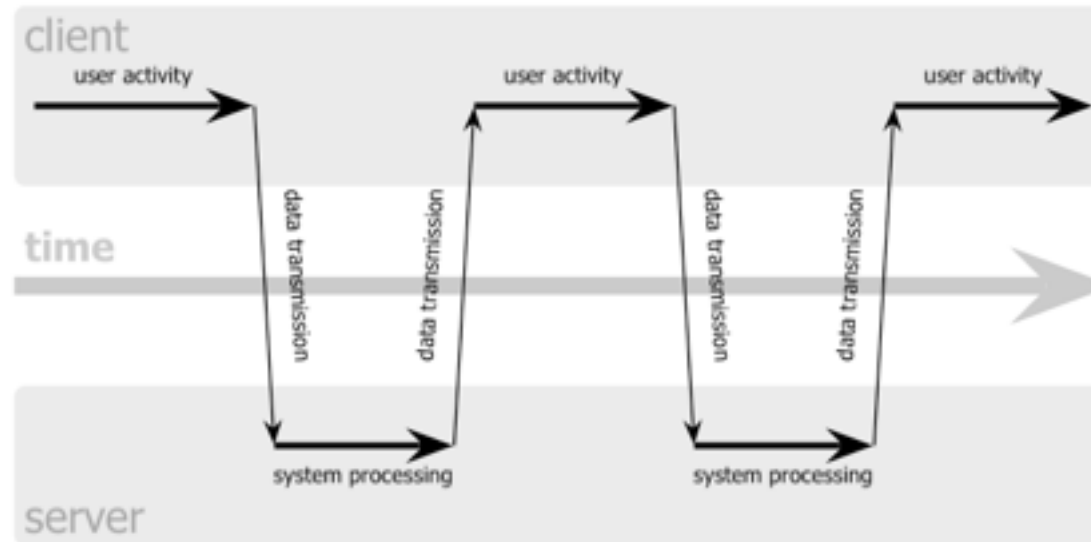


Ajax
web application model

process de la validation
ajouté

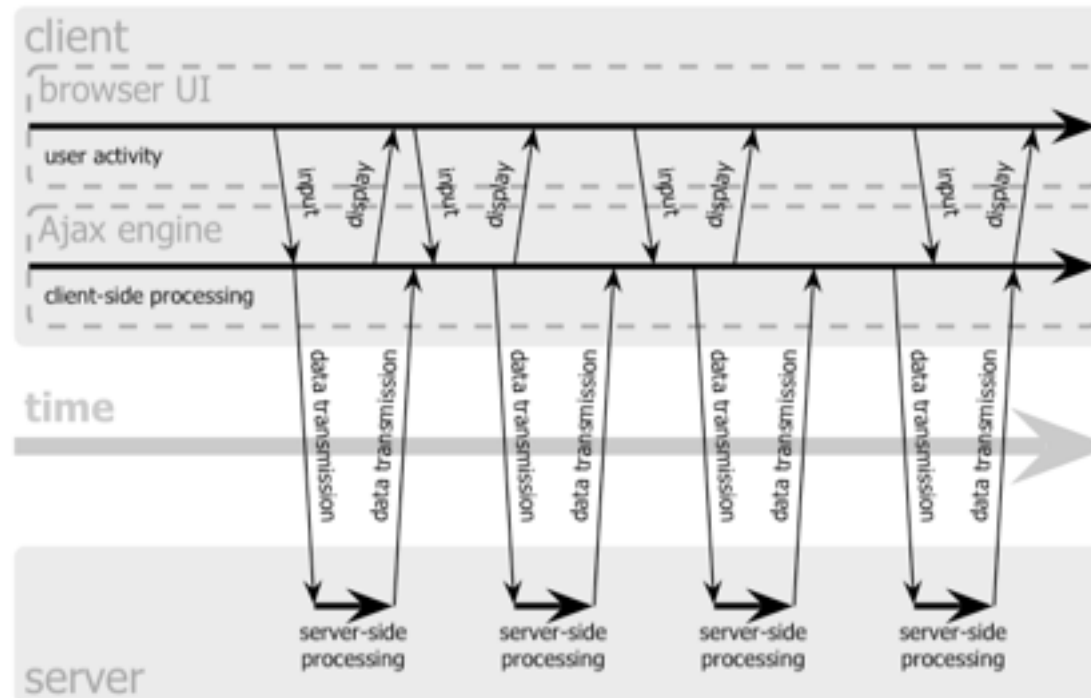


classic web application model (synchronous)



synchron: une fois qu'on envoie un changement il faut attendre pour son traitement et recevoir une reponse et puis on peut avancer

Ajax web application model (asynchronous)



Mise à jour de l'heure par le serveur

cette exemple en haut est important

- HTML
 - Création de la page à mettre à jour
- PHP
 - Création du script serveur qui envoie les infos
- Javascript
 - Création d'un objet XMLHttpRequest en indiquant son *call-back*
 - Modifier le document HTML

HTML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
  <title>Heures du client et du serveur</title>
  <script type="text/javascript" src="heures0.js"></script>
</head>
<body>
  <h2>Mise à jour via AJAX</h2>
  <p>Heure du serveur: <span id="serveur"></span></p>
</body>
</html>
```

Script Serveur

```
<?php-  
$today = getdate();-  
header('Content-Type: text/xml');-  
-  
print <<<EOX-  
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>-  
<response>-  
    <hours>{$today['hours']}    <minutes>{$today['minutes']}    <seconds>{$today['seconds']}</response>-  
EOX-  
."\\n";-
```

```
function getElementText(doc,e){- e contient hour minute and second
```

```
    return doc.getElementsByTagName(e)[0].textContent;-
```

```
}-
```

```
// initialiser le XMLHttpRequest -
```

```
var xhr; // initialisation-
```

```
if(window.ActiveXObject) xhr=new ActiveXObject("Microsoft.XMLHTTP");-
```

```
else if (window.XMLHttpRequest) xhr = new XMLHttpRequest();-
```

```
else throw new Error("Ajax not supported");-
```

```
-
```

```
xhr.onreadystatechange = function (){ // callback-
```

```
    var serverElem = document.getElementById("serveur");-
```

```
    if (this.readyState==4 && this.status>=200 && this.status<300){-
```

```
        var xml = this.responseXML;-
```

```
        serverElem.innerHTML-
```

```
            = formatHeure(getElementText(xml,"hours"),-
```

```
                           getElementText(xml,"minutes"),-
```

```
                           getElementText(xml,"seconds"));-
```

```
    }-
```

```
}-
```

```
-
```

```
function formatHeure(h,m,s){-
```

```
    return (h<10?" ":"")+h+": "+(m<10?"0 ":"")+m+": "+(s<10?"0 ":"")+s;-
```

```
}-
```

```
-
```

```
function insererHeure(){-
```

```
    xhr.open("get","heures0.php");-
```

```
    xhr.send(null);-
```

```
    setTimeout("insererHeure()",2000); // mise à jour aux 2 secondes-
```

```
}-
```

```
-
```

```
window.addEventListener("load", insererHeure,false);
```

Javascript

pour Internet
Explorer anciens

pour les autres navigateurs

entre 200 - 300 la
connexion est ok

Heure du serveur vs locale



- Plusieurs types de transfert d'information
 - text
 - xml
 - json
- Utilisation de jQuery
 - `$ ("#serveurJQ") .load ("heures.php" , "text")`
- Sources
 - html
 - php
 - javascript

★ JSON: JavaScript Object Notation

- Alternative à XML pour les cas simples
- Facile à lire et à écrire pour les humains
- Facile à analyser et à générer par les machines
- Objet
 - { } collection non-ordonnée de paires *chaine:objet*
 - [] liste ordonnée d'objets
 - éléments séparés par une virgule
- *Parsing facile* en Javascript
- API disponible dans d'autres langages

Exemple de donnée JSON

```
{ "wine-list": {  
  "wine": [{  
    "name": "Domaine de l'île Margaux",  
    "appellation": "Bordeaux supérieur",  
    "is-red": true,  
    "origin": {  
      "country": "France",  
      "region": "Bordeaux"  
    },  
    "price": 22.80,  
    "year": 2002  
  }, {  
    "name": "Riesling Hugel",  
    "appellation": "Alsace",  
    "is-red": false,  
    "origin": {  
      "country": "France",  
      "region": "Alsace and East"  
    },  
    "price": 17.95,  
    "year": 2002  
  }  
]  
}
```

Complétion dans un champ de texte [version jQuery]



- Sources
 - html [version jQuery]
 - php
 - javascript [version jQuery]

Désavantages de Ajax

- Complexité car on combine
 - logique de présentation du modèle de la page
 - logique du serveur pour générer les données
- Pas de standardisation des mécanismes
 - XMLHttpRequest Working draft W3C déc 2012
 - Javascript entre les browsers
 - mais il y a les Framework JS: jQuery ou Prototype
- Délicat à mettre au point
- Source est visible

Conclusion

- Possibilité de faire des interfaces web réactives comme des applications plus classiques
- Pas encore de méthodologie claire