Ajax - Généralités

- Ajax est un ensemble de techniques utilisées pour permettre au client 1) de communiquer avec le serveur en arrière-plan et 2) intégrer les réponses dans la page courante, sans la recharger
- Ajax propose des communications asynchrones : l'utilisateur n'est pas bloqué pendant l'exécution de la requête
- Côté client, Ajax repose sur une extension de JavaScript, l'objet XMLHttpRequest

Ajax – Exemple de suggestions de recherche

- La barre de recherche de Google (<input type="text" ...>) réagit aux actions du clavier : dès qu'un nouveau caractère est tapé, une fonction JavaScript est appelée
- Cette dernière envoie le texte en cours au serveur, pour une pré-recherche rapide
- Dès la réception de la réponse, une liste déroulante (<select>) est affichée sous la barre de recherche, affichant les différentes suggestions
- Le principal défi est de réagir au plus vite aux changements : comme l'utilisateur n'arrête pas de taper pendant l'exécution des premières requêtes Ajax, il faut p.ex. éviter d'afficher les suggestions pour les 5 premiers caractères, si on dispose déjà de celles pour les 7 premiers, à moins que l'utilisateur n'ait effacé une partie de sa chaîne...

Ajax – Avec JQuery

- Manipuler l'objet XMLHttpRequest directement n'est pas aisé
- JQuery fournit un ensemble de fonctions pratiques pour faire des appels Ajax (http://api.jquery.com/category/ajax/) :
  - fonction générique – jQuery.ajax()
  - fonctions spécifiques pour les cas les plus fréquents – jQuery.get(), jQuery.post(), jQuery.getJSON()
  - pour envoyer un formulaire en arrière-plan, la fonction .serialize() permet de l'encoder en chaîne de caractères compréhensible par le serveur
- Les appels (get(), post(),...) renvoient un objet jqXHR qui englobe XMLHttpRequest et ajoute de nouvelles fonctions : .done(), .fail(),...

Ajax – Exemple avec JQuery

```
...
var url = 'http://www.grenoble-inp.fr/'
...
jQuery.get(url, function(data) {
    var size = data.length
    var $logo = $(data).find('img[alt="logo"]')
    $('#entete img').attr('src', $logo.attr('src'))
}).fail(function() {
    $('#entete img').remove()
})
...
```

requête GET envoyée à l'url

fonction appelée lorsqu'une réponse correcte (code 200) arrive : data contient alors le corps de la réponse

fonction appelée en cas de réponse en erreur (code 4xx ou 5xx)

Ajax – Limitations et inconvénients

- Faire évoluer les pages en arrière-plan sans les recharger rend plus difficile de poser des marque-pages et de gérer l'historique de navigation
- Il faut fournir des efforts supplémentaires pour permettre d'indexer correctement des pages dynamiques
- Same origin policy : le navigateur refusera le code JavaScript provenant d'un site A d'accéder aux données provenant d'un autre site B. (Même origine = mêmes protocole, serveur et port)