

INF4375 - Paradigmes des échanges Internet

Ajax

Jacques Berger

Objectifs

Introduire le paradigme Ajax

Comprendre le fonctionnement

Discuter de COMET

Prérequis

Javascript

Formats de sérialisation

Types d'application

Applications souveraines

Utilisation durant plusieurs heures :

IDE

Traitement de texte

Tableur

Logiciel de courriels

Etc.

Types d'application

Applications éphémères

Utilisation sporadique :

Calculatrice

Achats en ligne

Moteur de recherche

Etc.

Modèles

Modèle d'une application web traditionnelle

Modèle d'une application Ajax

Ajax

Asynchronous Javascript And XML
Maintenant un nom commun

Objectifs :

Amener des applications souveraines sur le web (ex. : gmail, google maps, google apps)

Améliorer la convivialité et la fluidité des applications web

Principes fondateurs

Le fureteur reçoit initialement une application Javascript

Ensuite, le serveur sert des données

Les interactions avec l'utilisateur sont fluides et continues (asynchrones)

Technologies

HTML : format du document

Javascript : script client

DOM : manipulation de la page HTML

XMLHttpRequest : communication entre le
navigateur et le serveur

Échanges de données (optionnelle) :
XML, JSON, HTML

Types d'échange de données

Client seulement

Aucune interaction avec le serveur après le chargement initial

Centré-contenu

Le serveur retourne des noeuds HTML

Types d'échange de données

Centré-script

Le serveur retourne du code Javascript

Le code est exécuté par le client

Centré-données

Le serveur ne retourne que des données

Les données peuvent être brutes ou sérialisées dans un format comme XML ou JSON

Les données sont interprétées par le client

COMET

Autres noms

Ajax-Push

Reverse Ajax

Concept apparu en 2008

COMET

Ajax n'est pas vraiment asynchrone

Le fureteur doit constamment vérifier si la réponse est arrivée

La connexion HTTP est fermée après chaque envoi

COMET

Ajax utilise la technique du «Polling»

COMET permet d'être vraiment asynchrone

COMET

Polling (Ajax)

Le client fait une requête au serveur

Le client vérifie à répétition si la réponse est arrivée

La requête et la réponse sont sur deux connexions HTTP distinctes

COMET

Long-Polling (COMET)

Le client ouvre une connexion HTTP avec le serveur

Le serveur ne ferme pas la connexion avant d'avoir quelque chose à envoyer au client

Lorsqu'une réponse arrive, la connexion est fermée

Il faut ouvrir une connexion pour chaque requête

COMET

Streaming (COMET)

Le client ouvre une connexion HTTP

Cette connexion restera toujours ouverte

Le serveur écrit sur la connexion lorsqu'il doit communiquer avec le client

Liens

XMLHttpRequest

<http://www.w3.org/TR/XMLHttpRequest/>