Pages WWW interactives et CGI

Code: cgi-intro

Originaux

url: **http://tecfa.unige.ch/guides/tie/html/cgi-intro/cgi-intro.html**

url: **http://tecfa.unige.ch/guides/tie/pdf/files/cgi-intro.pdf**

Auteurs et version

- Daniel K. Schneider - Patrick Jermann - Vivian Synteta

- Version: 1.1 (modifié le 15/1/07 par DKS)

Prérequis:

- Connaître le fonctionnement du WWW
- Connaître le langage HTML de base et les formulaires

  Module technique précédent: html-intro

  Module technique précédent: html-forms

Objectifs:

- Comprendre le principe de fonctionnement du CGI
- Savoir différencier les requêtes POST et GET

Suites possibles: PHP ou JSP

Module technique suivant: php-intro et ensuite php-html

Module technique suivant: java-intro et ensuite java-jsp

1. Table des matières

2. Définition de "Pages WWW interactives"

2.1 HTML interactif

A. Server-side scripting

formulaire HTML + programme "cgi-bin" ou servlet installé sur un serveur

"html-embedded scripting langages":
formulaire HTML + PHP, ASP (Active Server pages), JSP (Java Server pages), etc.

B. Client-side scripting avec JavaScript

formulaire HTML + programme JavaScript téléchargé


Interface commune: formulaires HTML

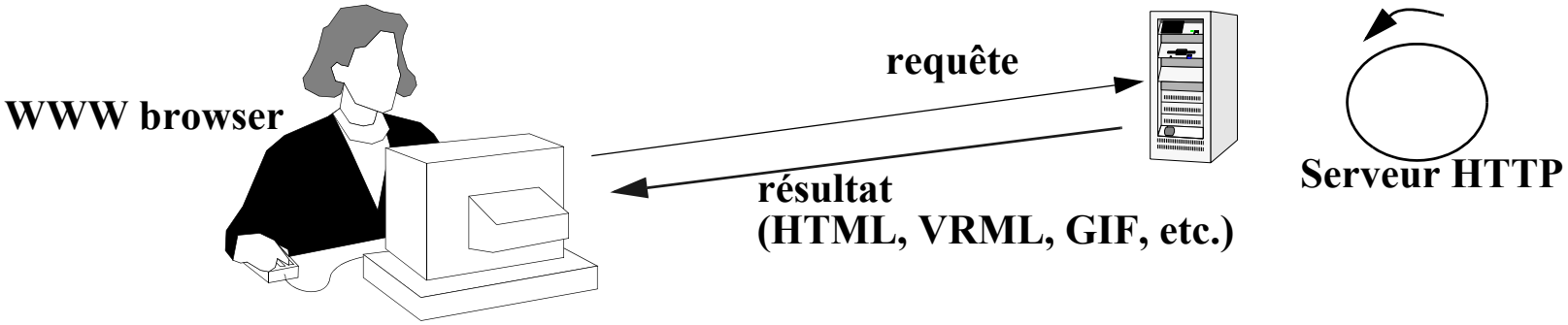
2.2 JAVA Applets

Un applet Java tourne indépendamment d'une page HTML

- ... nécessite plus de connaissances en programmation

3. Introduction à HTTP / Server-side scripting

3.1 Le principe de base du HyperText Transfer Protocol (HTTP)



- Cliquer sur un URL implique (simplifié):
  - (1) Ouvrir une connexion vers un serveur HTTP et lui envoyer quelque lignes qui formulent une requête
  - (2) Le serveur “exécute” la requête (rend simplement un fichier ou exécute un programme qui produit un fichier “on the fly”).
  - (3) Fermer la connexion (“statelessness”, pas de connexions permanentes!)
- La réponse du serveur
  - contient aussi quelques lignes “header” + un contenu
  - dit par ex. au browser de quel type de fichier il s’agit
  - Le browser tente de représenter le contenu ou de lancer un “helper” (il connaît au moins HTML, FTP, News, etc.)

A. Une requête typique envoyé par un browser (=GET)

Une requête GET (ici: http://tecfamoo.unige.ch:7778/4026/):

```
GET /4026/ HTTP/1.0
Referer: http://tecfa.unige.ch:7778/&eweb_room
Connection: Keep-Alive
User-Agent: Mozilla/3.01 (X11; I; SunOS 5.4 sun4m)
Host: tecfa.unige.ch:7778
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
```

- GET demande un "URL" (un fichier) en donnant un nom de protocol
- "***Referer***" indique la page qui contenait le "lien hypertexte"
- User-Agent indique le browser
- Accept: indique ce que le browser sait gérer comme type d'images (ici)

Voici quelques URLs qui montrent:

url: **http://tecfa.unige.ch/guides/php/examples/http-basics/request-headers.php**

url: **http://tecfa.unige.ch/cgi-bin/test-cgi.tcl**

url: **http://tecfa.unige.ch/cgi-bin/test-cgi**

url: **http://tecfa.unige.ch/cgi-bin/test-env**

Note:

- Avec JavaScript on peut obtenir de l'information supplémentaire sur votre profil car JavaScript interagit avec votre browser.

B. La réponse typique a la forme suivante:

```
HTTP/1.0 304 Not modified
Server: .....
Date: Monday, .....
Content-Type: text/html
Content-Length: ....
Last-Modified: ....

..... contenu .....
```

La première ligne doit être une “status line” :

avec la syntaxe suivante:

<HTTP-Versions> <Status-Code> <Reason-Phrase>

- Exemple: `HTTP/1.0 304 Not modified`

Ensuite vient l’en-tête:

- dans l’exemple ci-dessus, juste Server: .....

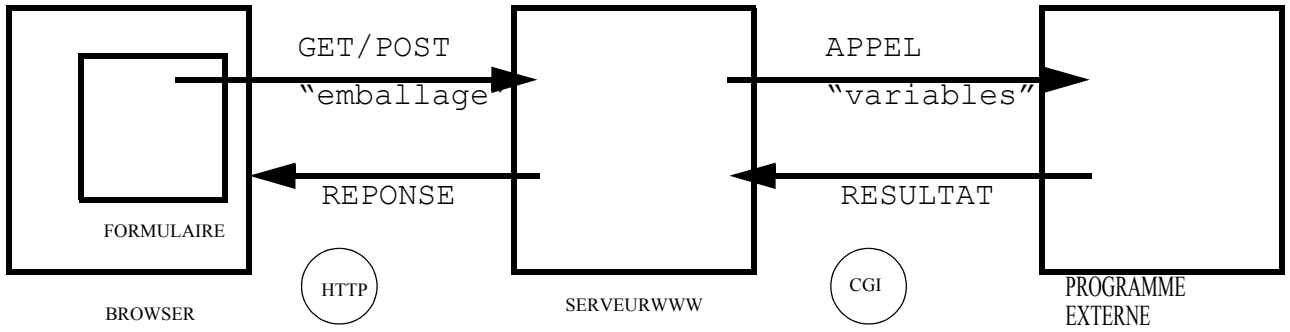
Ensuite le descriptif (et c’est important à connaître)

- Content-Encoding: Définit l’encodage de l’information
- Content-Length: Indique la taille du document
- Content-Type: Indique le type du document
- ... et il en a d’autres

Le contenu

- vient à la fin, séparé par une ligne blanche.

3.2 Principe du “Common Gateway Interface” (CGI)



Ceci est superficiel (!), voir aussi:

- les indexes “WebMaster” (http://tecfa.unige.ch/guides/toolbox.html)

L’emballage en “query string”

- Si vous utilisez un formulaire, votre client s’en occupe:
  chaque “***name=valeur***” est transmis au serveur
- Convention de base:
  - tout en un seul “string”
  - les caractères spéciaux sont traduits (par exemple l’espace en “+”)

```
name1=valeur&name2=valeur2&name3=valeur3
```

L'interface entre le serveur et les script

- normalement le serveur transmet l'information au script sous forme de variables d'environnement
  Notez que le serveur donne encore d'autres informations
  (comme l'hôte du client, l'heure, etc....)
- Ensuite le script doit "écrire" une réponse qui contient:
  - les bons headers HTTP
  - le "texte" (contenu) de la réponse (HTML, VRML, GIF ....)
- avec PHP c'est plus simple .....

Il existe 2 interfaces: GET et POST

- dans les 2 cas le principe consiste à "emballer" un "query" sous forme de "query string".
- GET permet de "bookmark" une requête
  POST permet d'envoyer des "query" plus longs

A. CGI avec GET

- La requête GET “CGI” envoyé ressemble à la commande GET “normale”

Exemple 3-1: Exemple fictif "GET" avec 2 variables:

url: http://tecfamoo.unige.ch:7778/~mooboy?q1=nulle&q2=nulle

Lignes envoyées par le client au serveur

```
GET /~mooboy?q1=nulle&q2=nulle HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/4.04 [en] (X11; I; SunOS 5.6 sun4m)
Pragma: no-cache
Host: tecfa.unige.ch:7778
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
Accept-Language: fr,en,de
Accept-Charset: iso-8859-1,*,utf-8
```

Voici les variables d’environnement données au programme CGI

```
HTTP_Connection = Keep-Alive
HTTP_Pragma = no-cache
REMOTE_HOST = fpssun19
TIME = 893235031
REQUEST_METHOD = GET
HTTP_USER_AGENT = Mozilla/4.04 [en] (X11; I; SunOS 5.6 sun4m
PATH_TRANSLATED = mooboy?q1=nulle&q2=nulle
HTTP_ACCEPT = image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*fr,en,deiso-8859-1,*,utf-
8"
.......
```

B. CGI avec POST

La requête HTTP contient en plus:

- une ligne indiquant la longeur du “POST”
- le “POST” string, précédé par une ligne blanche

Exemple 3-2: Une requête POST

url: **http://tecfa.unige.ch/tecfa/teaching/staf14/files/moo-form.html**

```
POST /cgi/&e_thing HTTP/1.0
Referer: http://tecfa.unige.ch/tecfa/tecfa-teaching/staf14/files/moo-form.html
Connection: Keep-Alive
User-Agent: Mozilla/3.01 (X11; I; SunOS 5.4 sun4m)
Host: tecfa.unige.ch:7778
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
Content-type: application/x-www-form-urlencoded
Content-length: 179

Edu-Comp=&name=&first_name=&institution=&site-url=&personal-url=&e-
mail=&activity=Education&hours=under5&archive=CICA&Color=fluo&Comments=%E9jhl%E9kjh%0D
%0ASLKshS&Rating=Dix+MILLE"
```

A retenir

- Content-type (un string encodé qui contient la requête)
- Content-length (la longeur du script)
- Le “query-string”