

La spécification CGI.

Patrick Poulingeas.

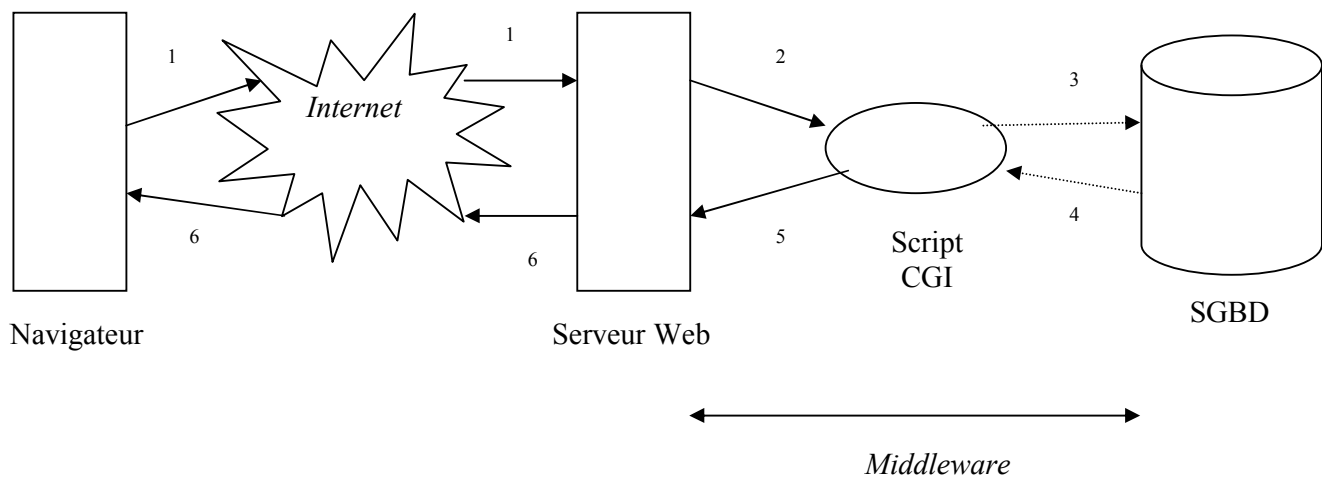
6 avril 2005.

1. Introduction.

CGI (*Common Gateway Interface*) est un ensemble de spécifications pour l'échange d'informations entre un serveur Web et un programme externe. Ce programme externe peut être écrit dans n'importe quel langage, dans la mesure où le langage permet de récupérer des données depuis l'entrée standard, d'écrire sur la sortie standard et d'accéder aux variables d'environnement du système. Ce programme externe, appelé programme CGI ou script CGI, est souvent écrit en Perl car ce langage permet de manipuler facilement des chaînes de caractères, le contenu des fichiers, etc.

Les scripts CGI servent à rendre le contenu d'un site dynamique, à intégrer partiellement celui-ci dans le système d'information de l'organisme ou de l'entreprise possédant le site.

2. Le flux d'information avec un site utilisant CGI.



1 : Envoi d'une requête HTTP du navigateur au serveur Web
2 : Le serveur Web fournit des informations au script CGI, informations en rapport avec les données que le serveur a reçues du navigateur (les associations champ/donnée d'un formulaire par exemple).

- 3 : Interrogation *éventuelle* d'un système de gestion de base de données (SGBD).
- 4 : Réponse du SGBD à la requête *éventuelle* du script CGI.
- 5 : Le script retourne un document au serveur Web. Pour cela, il écrit sur sa sortie standard. Les données envoyées par le script suivent le protocole HTTP.
- 6 : Le serveur Web renvoie au navigateur le document qui a été fabriqué par le script CGI.

Le script CGI peut retourner au serveur Web un document HTML, ou une image, ou un fichier texte, etc. C'est pourquoi, dans sa réponse au serveur Web, il commence par indiquer le type MIME du document qu'il a produit, ceci avec l'en-tête HTTP : Content-type. Le script renvoie ses données au serveur en écrivant sur la sortie standard (que la méthode soit GET ou POST).

3. Quelques variables d'environnement renseignées par le serveur.

- REQUEST_METHOD : méthode utilisée pour la requête (GET ou POST).
- REMOTE_HOST : nom de l'hôte d'où provient la requête.
- REMOTE_ADDR : adresse IP de l'hôte d'où est issue la requête.
- QUERY_STRING : contient les informations transmises au programme CGI (quand l'attribut « method » d'un formulaire vaut la valeur « GET »).
- CONTENT_LENGTH : longueur en octets des données disponibles dans l'entrée standard (pour la méthode POST).
- GATEWAY_INTERFACE : version de la spécification CGI employée par le serveur Web.
- SERVER_PORT : port sur lequel la requête HTTP du navigateur est arrivée sur le serveur.

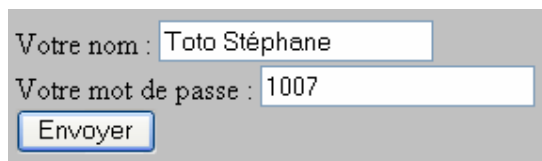
4. Gestion des données avec la méthode GET pour un formulaire.

Quand on soumet le formulaire au serveur, une URL spéciale est constituée par le navigateur.

Supposons que l'on trouve dans un fichier référencé par l'URL <http://www.boofr/membres/index.html> le formulaire suivant :

```
<form name="saisieDonnees" method="GET" action="/cgi-bin/traiter.pl">  
  Votre nom : <input type="text" name="nom"><br />  
  Votre mot de passe : <input type="text" name="passwd"><br />  
  <input type="submit" value="Envoyer">  
</form>
```

L'internaute saisit les données suivantes puis presse le bouton de soumission du formulaire :



The image shows a web form with a light gray background. It contains two text input fields. The first field is labeled 'Votre nom :' and contains the text 'Toto Stéphane'. The second field is labeled 'Votre mot de passe :' and contains the text '1007'. Below these fields is a blue button with the text 'Envoyer'.

L'URL générée est alors :

<http://www.boofr/cgi-bin/traiter.pl?nom=Toto+St%E9phane&passwd=1007>

Ce qui suit le « ? » dans l'URL sera stocké dans la variable d'environnement QUERY_STRING.

On voit qu'il y a un encodage des données :

- + correspond à un espace
- %E9 correspond à « é » (On a un codage similaire pour tout ce qui n'est pas caractère ASCII)
- %2B remplace le symbole « + » (si celui-ci se trouve dans les données saisies par l'internaute)
- %3D remplace le symbole « = » (si celui-ci se trouve dans les données saisies par l'internaute)
- %26 remplace le symbole « & » (si celui-ci se trouve dans les données saisies par l'internaute)
- %25 remplace le symbole « % » (si celui-ci se trouve dans les données saisies par l'internaute)

Ceci est dû au fait que dans une URL, selon la RFC 1738, ne peuvent apparaître que des chiffres, des lettres, les caractères spéciaux : « ; / ? : @ = & » et les caractères suivants : « \$ - _ . + ! * ' () , ». De plus, « % » est un caractère d'échappement permettant d'introduire de façon sécurisée dans l'URL les autres caractères.

Pour une URL HTTP, on dispose d'une spécification plus précise (Usage de « # » pour les ancrs, ...).

Dans l'URL formée par le navigateur, les différentes associations nom/valeur des éléments du formulaire sont séparées par un « & ». Entre un nom et une valeur apparaît le symbole « = ».

5. Gestion des données avec la méthode POST pour un formulaire.

Avec la méthode POST, les données du formulaire sont transmises dans le corps de la requête HTTP envoyée au serveur. Ces données sont ensuite fournies par le serveur Web au script CGI par l'entrée standard de ce dernier.

Exemple *extrêmement simplifié* de script CGI en Bash pour la méthode POST :

```
#!/bin/sh
```

```
# Script : cgi.sh
```

```
echo Content-type: text/plain
```

```
echo
```

```
echo Les donnees etaient les suivantes :
```

```
read x
```

```
echo $x
```

```
exit 0
```

Ce script se contente de renvoyer au navigateur, sous forme de fichier texte, les données que le serveur lui a transmises.

Exemple de résultat obtenu si l'on reprend le formulaire précédent avec les mêmes données fournies par l'utilisateur (mais avec utilisation de la méthode POST et appel au script cgi.sh pour traiter les données) :

Les donnees etaient les suivantes :
nom=Toto+St%E9phane&passwd=1007

Remarques :

- Les « vrais » programmes CGI doivent utiliser la variable d'environnement CONTENT-LENGTH pour déterminer le volume de données à lire.
- Sans bibliothèque adaptée, le décodage des données reçues par le script n'est pas forcément trivial.

6. Appel d'un script CGI hors du cadre d'un formulaire.

Il n'est pas obligatoire de passer par un formulaire pour utiliser le mécanisme CGI.

On pourrait gérer un compteur d'accès à une page HTML de la sorte :

```

```

Le script acces.pl calculera le nombre d'accès à la page (Pour cela, il peut par exemple utiliser un fichier). Il renverra ensuite une image correspondant à ce nombre (dans un format jpeg ou png par exemples). Le script CGI générera donc une image adéquate à chaque fois que la page sera visitée.

On peut aussi bien évidemment appeler un script CGI en formant une URL adaptée que l'on entre directement dans la barre d'adresse du navigateur.

7. Problèmes avec les scripts CGI.

Le script peut rentrer dans une boucle infinie et ne rien retourner au serveur. Dans ce cas-là, au bout d'un certain temps, le serveur renverra un message HTTP d'erreur au client. D'une façon générale, une erreur au niveau d'un script CGI se traduira par l'envoi d'un message d'erreur au navigateur, sans possibilité pour le serveur d'intervenir de façon précise.

Les scripts CGI posent souvent des problèmes de sécurité car ils sont exécutés avec l'UID du serveur qui peut posséder des droits élevés. Sans évoquer des scripts setuid appartenant à l'utilisateur « root ». Des failles peuvent être facilement exploitées car il suffit de forger une URL « malicieuse » dans la barre d'adresse d'un navigateur.