Pràctica 6: Disseny web dinàmic amb CGI

Aplicacions i serveis d'internet — iTIC

Francisco del Águila López Aleix Llusà Serra Alexis López Riera

9 d'abril de 2014

Índex

Resum

Protocols d'internet: HTTP, CGI
Llenguatges web: HTML

1 Organització

1.1 Objectius

El objectius d'aquesta pràctica són:

1. Entendre el disseny web dinàmic.
2. Dissenyar dinàmicament mitjançant scripts CGI.
3. Usar els mòduls `cgi` de la llibreria de `Python`.
4. Atendre el formulari de la pràctica de disseny web.

1.2 Condicions

- La pràctica està calibrada per a ésser treballada en equips de dues persones.
- La durada de la pràctica és d'1 setmana.

1.3 Lliurables

Heu d'entregar l'script CGI que dissenyeu.

1.4 Material necessari

En aquesta pràctica utilitzarem la infraestructura de servei web dissenyada a la pràctica 5 i el portal web dissenyat a la pràctica 4.

2 Introducció a CGI

El CGI (*Common Gateway Interface*) és una tècnica web d'execució al servidor (*server-side scripting*) que permet que un servidor web interaccioni amb programes externs, els quals en aquest context s'anomenen scripts o programes CGI.

Amb CGI es pot incloure contingut dinàmic a les pàgines web mantenint la independència entre els servidors web i el disseny web. Quan un usuari sol·licita una URL que és un script CGI, el servidor web envia la informació d'aquesta sol·licitud a l'script i retorna la sortida de l'script al client.

2.1 Escenari

En el vostre domini d'autoritat `gXX.asi.itic.cat` heu de dissenyar un portal web que permeti treballar amb dades d'usuaris remotament. Aquesta interfície web permetrà per una banda registrar el nom i les dades de l'usuari i per altra banda consultar aquesta informació. A més, cada nou registre d'usuari s'enviarà per correu als administradors del portal, els quals sereu vosaltres mateixos.

Per al disseny web del portal aprofiteu els documents HTML dissenyats a la pràctica 4.

3 Servidor web + CGI

El disseny de webs dinàmiques amb CGI es compon de dues parts:

- El disseny d'scripts CGI
- L'execució dels scripts CGI per part del servidor web

Aquestes dues parts de disseny web dinàmic complementen les parts vistes en pràctiques anteriors del disseny web estàtic amb documents HTML i de servir-los en un servidor web.

En aquesta pràctica utilitzarem `Python` com a llenguatge d'script CGI i `Apache` com a servidor web.

Per al disseny CGI amb `Python` utilitzarem el mòdul CGI [Pyt13a].

Per a l'execució d'scripts CGI a `Apache` hi ha diverses maneres [Apa13]. Comproveu que el mòdul de cgi estigui habilitat, sinó activeu-lo amb `a2enmod cgi`. Seguirem la configuració del fitxer d'exemple inicial: segons la versió d'`Apache` `/etc/apache2/sites-available/default` o bé `/etc/apache2/conf-available/serve-cgi-bin.conf`:

```
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Require all granted
</Directory>
```

Amb aquesta configuració tots els scripts CGI s'han de situar al directori `/usr/lib/cgi-bin/` (l'opció *ScriptAlias* és semblant a *Alias* però per a scripts), han de tenir permisos d'execució i es poden consultar a partir de la URL `http://localhost/cgi-bin/`.

Recordeu que `Apache` s'executa com a usuari *www-data* i per tant no pot crear fitxers a `/usr/lib/cgi-bin/`. Si un script necessita escriure fitxers caldrà situar-los en directoris escaients.

3.1 Script CGI estàtic

En primer lloc, escrivim un exemple bàsic d'script CGI per al disseny d'una web estàtica.

Per exemple, en un fitxer de nom `/usr/lib/cgi-bin/exemple.cgi`:

```
#!/usr/bin/python
# -*- encoding: utf-8 -*-

print "Content-Type: text/html" # HTML is following
print # blank line, end of headers

print """<html>
 <head>
 <meta charset=''utf-8'' />
  <title>La pàgina</title>
 </head>
```

```
 <body>
  <p>Un paràgraf</p>
 </body>
</html>
"""
```

Recordeu que l'script tingui permisos d'execució i visiteu-lo a `http://localhost/cgi-bin/exemple.cgi`

Els scripts CGI comuniquen el resultat al servidor web a través de la sortida estàndard. El resultat són les capçaleres i el cos d'un missatge de resposta HTTP; ambdós se separen per una línia en blanc. A continuació detallem aquestes dues parts.

3.1.1 Capçaleres HTTP

El primer que ha d'escriure l'script CGI són les capçaleres del missatge de resposta HTTP. Les capçaleres HTTP són paràmetres del missatge definits amb línies de parelles de clau i valor, en podeu consultar una llista a `http://en.wikipedia.org/wiki/HTTP_response#Responses` o a `http://www.iana.org/assignments/message-headers/message-headers.xml`.

Per exemple:

```
print "Content-Type: text/html"
print "Content-Language: ca"
print # blank line, end of headers
```

Una capçalera obligatòria és la primera, que determina el tipus MIME del contingut que hi ha en el cos HTTP. Per exemple el tipus MIME text/html indica que el cos conté un document HTML i image/png indicaria que el cos és una imatge PNG. Quan és un document de text, a la declaració de contingut també s'hi pot incloure la codificació del fitxer, p.ex.

```
Content-Type: text/html; charset="utf-8"
```

Podeu escriure altres capçaleres com per exemple la segona, que indica l'idioma del contingut. Quan el servidor web executa l'script i retorna la resposta també omple algunes capçaleres més.

3.1.2 Cos HTTP

Un cop s'han escrit les capçaleres, ja es pot escriure el cos del missatge HTTP. El tipus d'aquest contingut s'ha de correspondre amb el que s'ha declarat a la capçalera.

Seguint amb l'exemple d'escriure un document HTML estàtic, recordeu que també el podeu escriure línia a línia:

```
print '<html>'
print '<head>'
print '<title>La pàgina</title>'
print '</head>'
print '<body>'
print '<p>Un paràgraf</p>'
```

```
print '</body>'
print '</html>'
```

i utilitzar totes les propietats de les cadenes que ja coneixeu:

```
s = """<html>
 <head>
  <title>{titol}</title>
 </head>
 <body>
  <p>{p0}</p>
 </body>
</html>
"""
print s.format(titol='La pàgina',p0='Un paràgraf')
```

3.2 Script CGI dinàmic

Els scripts CGI reben la transacció del servidor web per l'entrada estàndard i per les variables d'entorn de la shell. El mòdul CGI de `Python` és una interfície per consultar aquesta entrada.

Mòdul CGI:

```
import cgi
```

A més, hi ha unes eines per ajudar a debugar:

```
import cgitb
cgitb.enable()
```

3.2.1 Variables d'entorn

Les trobareu en el diccionari cgi.os.environ. Proveu de mostrar-les amb cgi.print_environ().

A `http://en.wikipedia.org/wiki/HTTP_response#Requests` podeu trobar una llista de les capçaleres dels missatges de petició HTTP.

3.2.2 Atendre l'entrada GET o POST

Una sol·licitud GET o POST d'HTTP pot venir acompanyada de dades que l'usuari ha afegit. Aquestes dades d'entrada en el cas de GET són part de la URL de sol·licitud i en el cas de POST són el cos del missatge.

L'usuari pot haver afegit dades a les sol·licituds de maneres diferents, algunes són:

- Enviament d'un formulari HTML
- Consulta d'una URL amb paràmetres específics
- Execució de sol·licituds des de JavaScript

A `http://www.w3schools.com/tags/ref_httpmethods.asp` hi ha una comparació dels usos dels mètodes GET i POST.

Atendre sol·licituds de formularis i semblants Una manera típica d'entrar les dades és com a *query string*. El *query string* és una part dels esquemes URL, consulteu `http://en.wikipedia.org/wiki/Query_string` si no ho recordeu. Bàsicament consisteix en codificar parelles de clau i valor amb el format: `clau1=valor1&clau2=valor2`.

En el context dels formularis HTML, el format de *query string* es formalitza com a tipus MIME *application/x-www-form-urlencoded*, consulteu `http://www.w3.org/TR/html401/interact/forms.html#form-content-type` per a més detall. En les sol·licituds GET es codifica a la URL precedit per un interrogant i en les POST es codifica en el cos del missatge.

La llibreria CGI permet obtenir les dades d'entrada de les sol·licituds que estan escrites com a *query string*, independentment de si han entrat amb mètode GET o POST i de si l'origen és un formulari, url o JavaScript.

En el cas dels formularis d'HTML, quan l'usuari escull enviar-lo és atès segons la configuració de dos atributs HTML del formulari:

- `action` indica l'adreça URL a la qual s'envia la sol·licitud; per tant es correspon amb l'script CGI que l'ha d'atendre.
- `method` indica el mètode HTTP – GET o POST – amb què s'envia la sol·licitud. Lligat al mètode POST, en els formularis HTML hi ha l'atribut `enctype` que defineix el tipus MIME amb què s'ha de codificar la sol·licitud. Noteu que definir el mètode GET és equivalent a visitar la URL `?camp1=valor1;camp2=valor2...`.

Per atendre les sol·lcituds dels formularis i semblants, `CGI` ofereix estructures d'alt nivell. En el cas de `Python` proveu de mostrar les dades que ha enviat l'usuari amb:

```
form = cgi.FieldStorage()
cgi.print_form(form)
```

form és un objecte implementat sobre els diccionaris, al qual podeu aplicar els mètodes escaients:

- form['nom']
- form['nom'].value
- 'nom' **in** form
- form.getvalue('nom')
- etc.

Seguiu la documentació CGI de `Python` [Pyt13a] per a entendre el funcionament dels objectes FieldStorage. Penseu que hi ha una interfície d'entrada d'usuari i per tant el vostre script CGI ha de ser robust. Possibles mètodes GET que poden fer els usuaris:

- `exemple.cgi?nom=tal;edat=23`
- `exemple.cgi?edat=23`
- `exemple.cgi?nom=tal;edat=23;nom=qual`

Fixeu-vos que en el segon cas s'ha omès un camp i en el tercer hi ha un camp repetit. Els objectes FieldStorage de CGI tenen mètodes robusts per a aquests casos:

```
form.getfirst('nom')
form.getlist('nom')
```

Atendre GET o POST directament Quan les dades no tenen format de formulari, cal atendre directament la sol·licitud sense l'ajuda dels objectes d'alt nivell de CGI. En aquest cas cal entendre com el servidor web es comunica amb els scripts CGI:

- En el cas del mètode GET, podeu trobar la sol·licitud a la variable d'entorn *query_string* de la shell:

  ```
  cgi.os.environ["QUERY_STRING"]
  ```

- En el cas del mètode POST, la sol·licitud és a l'entrada estàndard:

  ```
  cgi.sys.stdin
  ```

Podeu trobar més informació a la documentació d'`Apache` [Apa13].

4 Feina a fer

1. Llegir, provar i entendre la documentació d'aquesta pràctica. La documentació de referència és el CGI de `Python` [Pyt13a].
2. Dissenyar i implementar un portal web que atengui el formulari de la pràctica de disseny web i enviï correus.
   - El formulari sol·licita dades a l'usuari (nom, cognoms, etc.)
   - Un script atén el formulari i autoregistra dades de l'usuari (navegador, sistema operatiu, etc.). Emmagatzema les dades en un fitxer i les envia per correu a l'administrador del web.
   - Els administradors web poden veure les dades enregistrades descarregant-se directament el fitxer o bé consultant-lo a través d'una interfície web.
3. Comprovar el bon funcionament del portal web i el d'altres companys. Observeu amb `Wireshark` o amb la consola web del navegador com es fan les sol·licituds GET o POST.

4.1 Enviar un correu

En l'atenció del formulari web us hem proposat que envieu un correu a l'administrador web a cada entrada nova. Preneu-vos-ho com una millora de la vostra pràctica; requereix haver entès el servei de correu.

L'enviament del correu s'ha de programar en el CGI. No obstant això, seguiu els vostres coneixements de programació i organitzeu-ho convenientment en un mòdul a part.

Us proposem dues maneres per a enviar el correu:

- Execució de `mail` o `sendmail` mitjançant la llibreria subprocess de `Python`
- Llibreria smtp de `Python`

Teniu algunes pistes a les preguntes freqüents de `Python` [Pyt13b].

5 Extensions

5.1 Server Side Includes

Server Side Includes (SSI) és una tècnica web d'execució al servidor que consisteix en escriure directives en els documents HTML que després els servidors web saben interpretar. És una tècnica simple de disseny web dinàmic.

Per a `Apache` podeu consultar-ne les directives a `http://httpd.apache.org/docs/2.4/howto/ssi.html`.

5.2 Servidor web Python

Podeu usar `Python` com a servidor web dels scripts CGI. Useu el mòdul CGIHTTPServer, el qual estén la funcionalitat del mòdul SimpleHTTPServer amb la possibilitat d'executar scripts CGI.

Es pot invocar directament amb python −m CGIHTTPServer. Consulteu-ne la documentació a `http://docs.python.org/library/cgihttpserver.html`.

Referències

[Apa13] Apache Software Foundation. *Apache Tutorial: Dynamic Content with CGI*. Versió 2.4. Apache HTTP Server. `httpd.apache.org`, 2013. URL: `http://httpd.apache.org/docs/current/howto/cgi.html` (consultat 27 de mar. de 2013).

[Pyt13a] Python Software Foundation. *cgi – Common Gateway Interface support*. Versió 2.7.3. Python documentation. `docs.python.org`, 2013. URL: `http://docs.python.org/library/cgi.html` (consultat 27 de mar. de 2013).

[Pyt13b] Python Software Foundation. *Library and Extension FAQ – Python. How do I send mail from a Python script?* Versió 2.7.4. Python documentation. `docs.python.org`, 2013. URL: `https://docs.python.org/2/faq/library.html#how-do-i-send-mail-from-a-python-script` (consultat 23 d'abr. de 2013).