

Pràctica 7: Disseny web dinàmic amb CGI

Aplicacions i serveis d'internet — Enginyeria de Sistemes TIC

Francisco del Àguila López Aleix Llusà Serra

26 d'abril de 2023

Índex

| | |
|-----------------------------|----------|
| 1 Organització | 1 |
| 1.1 Objectius | 1 |
| 1.2 Condicions | 2 |
| 1.3 Lliurables | 2 |
| 1.4 Material necessari | 2 |
| 2 Introducció a CGI | 2 |
| 2.1 Escenari | 2 |
| 3 Servidor web + CGI | 2 |
| 3.1 Script CGI estàtic | 3 |
| 3.1.1 Capçaleres HTTP | 4 |
| 3.1.2 Cos HTTP | 5 |
| 3.2 Script CGI dinàmic | 5 |
| 3.2.1 Query String | 6 |
| 3.2.2 Formularis i CGI | 6 |
| 3.2.3 Python i CGI | 6 |
| 3.2.4 mòdul CGI de Python | 7 |
| 4 Feina a fer | 8 |
| 4.1 Enviar un correu | 8 |

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 i el portal web dissenyat dissenyats a les pràctiques anteriors.

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. Aquesta interfície web permetrà per una banda registrar el nom i les dades d'un nou usuari i per altra banda consultar aquesta informació. A més, cada nou registre d'usuari s'enviarà per correu als administradors del portal, que sereu vosaltres mateixos.

Per al disseny web del portal aprofiteu els documents HTML dissenyats a pràctiques anteriors.

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 corresponent al disseny web estàtic amb documents HTML i a servir-los en un servidor web. Cal recordar que amb JavaScript s'obté dinamicitat només en el costat del client.

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 [Pyt23a].

La documentació sobre scripts CGI a Apache es troba a [Apa23].

TASCA PRÈVIA 1 Comproveu que el mòdul de `cgi` estigui habilitat, sinó habiliteu-lo amb `a2enmod cgi`.

Un dels mecanismes per habilitar l'execució de scripts CGI és amb la directiva `ScriptAlias`.

TASCA PRÈVIA 2 Visualitzeu el contingut del fitxer `/etc/apache2/conf-available/serve-cgi-bin.conf` i comproveu que s'assembla al que es mostra a continuació.

```
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/` del vostre servidor. L'opció `ScriptAlias` és semblant a `Alias` però per a scripts. Aquests scripts han de tenir permisos d'execució i es poden consultar a partir de la URL `http://<ip_o_nom_maquina_servidor_apache_en_dns>/cgi-bin/`, sempre que estigui habilitada la visualització del contingut d'un directori per part del vostre Apache. És evident que també són consultables amb un terminal del vostre servidor al path `/usr/lib/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

TASCA PRÈVIA 3 Escriviu el fitxer `exemple.cgi` a dins del directori `/usr/lib/cgi-bin/` amb el següent contingut:

```
#!/usr/bin/python3
# -*- 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.

TASCA PRÈVIA 4 Assegureu-vos de tenir `python3` instal·lat al vostre servidor.

Aquest és un CGI simple per generar el contingut web estàtic.

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

Els scripts CGI passen el resultat al servidor web a través de la sortida estàndard (`stdout`). El resultat conforma el missatge de resposta HTTP. Per tant, en primer lloc es genera la capçalera, es deixa una línia en blanc i seguidament es genera el contingut del cos del missatge HTTP (que correspon al contingut HTML).

TASCA 5 Visiteu i visualitzeu el contingut web que genera aquest script (codi HTML) en un navegador. Escriviu la URL adequada per accedir al contingut des de la vostra màquina host.

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 `https://en.wikipedia.org/wiki/HTTP#HTTP/1.1_response_messages` o a `https://www.iana.org/assignments/message-headers/message-headers.xhtml`.

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>')
```

o utilitzar totes les propietats de les cadenes de Python 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'))
```

TASCA 6 Comproveu el diàleg entre client i servidor amb les eines de *Web Developer* de Firefox. Comproveu les capçaleres tant del missatge de petició com el de resposta entre el navegador web de la màquina host i el script CGI que heu definit al vostre servidor Apache.

3.2 Script CGI dinàmic

La generació d'un contingut HTML estàtic per part d'un programa o script CGI no té cap mena d'interès, ja que en aquest cas el servidor web pot lliurar directament el fitxer HTML corresponent. D'altra banda, sí que té interès generar un contingut HTML en funció d'alguna informació enviada per part del client. Aquest escenari és el que ha donat lloc al concepte de la Web 2.0.

Amb CGI queda definida la interfície (per passar paràmetres) entre el servidor web i els scripts CGI localitzats en algun directori del mateix servidor. D'aquesta manera un client amb protocol HTTP pot enviar certa informació al servidor web, que aquest haurà de transferir a l'script CGI. Aquesta interfície té dos possibles vies de transferència: l'entrada estàndard (stdin) o les variables d'entorn de la shell. Podeu trobar més informació a la documentació d'Apache [Apa23].

Un client pot enviar informació al servidor utilitzant qualsevol dels mètodes HTTP (get, post, put, ...) però generalment es fan servir els mètodes GET i POST. Recordeu que GET no té *body*, per tant l'enviament de la informació quan es fa servir el mètode GET es fa via la URL,

localitzada a la primera línia del missatge de petició HTTP fent servir el concepte de *Query String*.

3.2.1 Query String

El *query string* és una part dels esquemes URL, consulteu http://en.wikipedia.org/wiki/Query_string per a més informació. Bàsicament consisteix en codificar parelles de clau i valor amb el format: `clau1=valor1&clau2=valor2`. Aquest *string* s'afegeix al final de la URL incorporant el caràcter `?` just després de l'objecte que es vol visitar. Com podeu deduir, aquest objecte serà un CGI o equivalent.

3.2.2 Formularis i CGI

Els formularis HTML són un recurs per enviar informació al servidor des del client, integrats dins el context del navegador web. Altres eines com `curl` (que no són un navegador web) permeten establir una conversa HTTP amb qualsevol mètode (`get`, `post`, `put`, ...).

Quan un formulari HTML s'envia (*submit*) intervenen dos atributs que determinen com es tractarà:

- **action** indica l'adreça URL a la qual s'envia la informació associada al formulari, 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 informació. 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 informació. Consulteu https://www.w3schools.com/tags/att_form_enctype.asp per a més informació. En el cas del mètode POST, les dades viatgen en el cos del missatge HTTP.

En el cas del mètode GET, les dades viatgen formant part de la URL. Noteu que definir el mètode GET és equivalent a visitar la URL `?camp1=valor1;camp2=valor2...`. On `campX` correspon a cada **input** del formulari i `valorX` correspon als valors entrats al formulari.

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

TASCA 7 Quan hagueu definit el vostre formulari, comproveu com es realitza l'intercanvi de informació entre client i servidor escollint el mètode GET i POST en el formulari. Aproveiteu les *eines de desenvolupador* de Firefox per monitorar els missatges HTTP intercanviats.

3.2.3 Python i CGI

Quan el script CGI està fet amb Python i es vol atendre l'enviament de dades d'un formulari s'ha de fer un tractament o un altre, depenent del mètode HTTP que es faci servir per part del formulari. Apache lliurarà les dades del formulari al script CGI:

- En el cas del mètode GET, les dades es troben a la variable d'entorn *query_string*. En el mòdul `os` de Python existeix la propietat `environ` de tipus diccionari on es pot consultar la variable d'entorn `QUERY_STRING`.

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

- En el cas del mètode POST, les dades es troben a l'entrada estàndard. En el mòdul `sys` de `Python` es troba la propietat `stdin` que es pot tractar com un fitxer i per tant llegir les dades enviades pel formulari.

```
import sys
data = sys.stdin.readlines()
```

3.2.4 mòdul CGI de Python

A data de 26 d'abril de 2023 està disponible a `Python` el mòdul `cgi` que facilita el tractament de les dades enviades per un formulari enfront de `os.environ` i `sys.stdin`. Però aquest mòdul desapareixerà en futures versions de `Python` a favor d'altres mecanismes de traspàs de la informació entre servidor web i codi de programa. Per al cas específic de `Python` existeix el mecanisme WSGI més eficient que CGI, però també més complex. La virtut de CGI és la seva senzillesa, on per a cada petició HTTP es genera un procés CGI que l'atén. Això mateix també és el seu inconvenient, ja que generar un nou procés de sistema per a cada petició HTTP pot ser molt costós en quan a recursos de SO.

El mòdul `cgi` de `Python` ofereix una interfície de programació d'alt nivell que tracta els formularis de la mateixa manera independentment de si s'ha fet servir el mètode GET o POST.

El primer pas és importar el mòdul:

```
import cgi
```

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

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

El tractament dels formularis es pot fer 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` [Pyt23a] per a entendre el funcionament dels objectes `FieldStorage`.

Recordeu que en el cas de GET, possibles entrades (URL) són:

- `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')
```

4 Feina a fer

TASCA 8 Llegir, provar i entendre la documentació d'aquesta pràctica. Les referències donades poden ser de gran utilitat per entendre els diferents conceptes.

TASCA 9 Dissenyar i implementar un portal web que atengui el formulari de la pràctica de disseny web i envii correus.

- El formulari sollicita 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. També respon a l'usuari amb una pàgina web per tal que sàpiga que tot s'ha fet correctament.
- Els administradors web poden veure les dades enregistrades descarregant-se directament el fitxer o bé consultant-lo a través d'una interfície web.

TASCA 10 Comprovar el bon funcionament del portal web i el d'altres companys. Observeu amb *Wireshark* o amb les *Eines de Desenvolupador* del navegador com es fan les sollicituds 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 [Pyt23b].

Referències

[Apa23] Apache Software Foundation. *Apache Tutorial: Dynamic Content with CGI*. Versió 2.4. Apache HTTP Server. httpd.apache.org, 2023. URL: <http://httpd.apache.org/docs/current/howto/cgi.html> (consultat 24 d'abr. de 2023).

- [Pyt23a] Python Software Foundation. *cgi – Common Gateway Interface support*. Versió 3.11.3. Python documentation. [docs.python.org](http://docs.python.org/library/cgi.html), 2023. URL: <http://docs.python.org/library/cgi.html> (consultat 24 d'abr. de 2023).
- [Pyt23b] Python Software Foundation. *Library and Extension FAQ – Python. How do I send mail from a Python script?* Versió 3.11.3. Python documentation. [docs.python.org](https://docs.python.org/3/faq/library.html#how-do-i-send-mail-from-a-python-script), 2023. URL: <https://docs.python.org/3/faq/library.html#how-do-i-send-mail-from-a-python-script> (consultat 26 d'abr. de 2023).