CGI PROGRAMMING

Gianna Bellè

Aprile 1999

Programmazione CGI - parte cliente

Common Gateway Interface

Il Common Gateway Interface (CGI) è un metodo standard utilizzato da un server Web, per passare una richiesta di un utente Web ad un programma applicativo, e restituire dati all'utente.

Prima dell'avvento di Java era l'unico modo per interagire con pagine web.

Fa parte del protocollo HTTP.

Dato che l'interfaccia è consistente, un programmatore può scrivere applicazioni CGI in vari linguaggi. I più diffusi sono: C, C++, Java, and Perl.

L'HTTP server decide quali locazioni sono riservate per i programmi CGI, di solito "cgi-bin", "cgi", o "bin".

IMPORTANTE: gira sul server!

Outline

1. come raccogliere input da HTML form oppure da Java applets
2. come trasmettere i dati ad un programma CGI usando `GET` oppure `POST`

Usare HTML forms per comunicare a programmi CGI

1. la `FORM` tag specifica l'URL del programma CGI
2. ogni elemento di input viene identificato (`NAME`), ed eventualmente sono forniti di un valore iniziale (`VALUE`).
3. l'utente può fornire valori addizionali digitando in un textfield, selezionando un radio button, etc...
4. quando l'utente clicca su un bottone "Submit", o su un image map, gli elementi attivi vengono accumulati in una stringa della forma "name1=val1&name2=val2&...&nameN=valN", che vengono trasmessi all'URL specificato.
5. i dati possono venire trasmessi in due modi:

   `GET` appendendoli all'URL dopo un punto interrogativo.
   E' semplice ma impone limite al numero di caratteri che possono essere spediti.

   `POST` inviandoli in una linea separata.
   Non ha alcun limite.

HTML FORM Element

`<FORM ACTION="URL" ...> ... </FORM>`

Attributi:

ACTION specifica l' URL del programma CGI. Eventualmente anche e-mail address, ma problemi di compatibilità.

METHOD specifica come i dati verranno trasmessi al server HTTP. `GET` è il default.

ENCTYPE specifica il modo con cui i dati saranno codificati prima di essere trasmessi.
Il default è `application/x-www-form-urlencoded`, cioè

da	a
spazio	+
non alfanumerico	%XX (hex digits)

Inoltre (=) è posto tra i nomi e i valori e (&) fra entries.

FORM Input Elements

- INPUT: submit buttons, reset buttons, JavaScript buttons, checkboxes, radio buttons, textfields, password fields, ...
- TEXTAREA: crea un campo di testo a molte linee
- SELECT: combo boxes, list boxes

GET

```
<FORM ACTION="http://lyra.educ.disi.unige.it/~gbelle/cgi-bin/CgiPost.cgi">
  Campo1:
  <INPUT TYPE="TEXT" NAME="Textfield1" VALUE="Entry 1">
  <BR>
  Campo2:
  <INPUT TYPE="TEXT" NAME="Textfield2" VALUE="Entry 2">
  <BR>
  Bottone:
  <INPUT TYPE="SUBMIT" NAME="SubmitButton" VALUE="Send">
</FORM>
```

La richiesta inviata dal client consiste nella specifica della URL alla quale vengono "appesi" i dati richiesti (con un punto di domanda):

```
GET /cgi-bin/test?Textfield1=Entry+1&Textfield2=Entry+2&Sub
 mitButton=Send HTTP/1.0
User-Agent: Mozilla/4.06 [en] (X11; I; Linux 2.0.32 i586)
Host: lyra.educ.disi.unige.it:80
Accept: image/gif, image/jpeg, */*
...
```

POST

```
<FORM ACTION="http://www.apl.jhu.edu:5555/cgi-bin/test"
      METHOD="POST">
  Campo1:
  <INPUT TYPE="TEXT" NAME="Textfield1" VALUE="Entry 1">
  <BR>
  Campo2:
  <INPUT TYPE="TEXT" NAME="Textfield2" VALUE="Entry 2">
  <BR>
  Bottone:
  <INPUT TYPE="SUBMIT" NAME="SubmitButton" VALUE="Send">
</FORM>
```

I dati venono inviati nel corpo del messaggio:

```
POST /cgi-bin/test HTTP/1.0
User-Agent: Mozilla/4.06 [en] (X11; I; Linux 2.0.32 i586)
Host: lyra.educ.disi.unige.it:80
Accept: image/gif, image/jpeg, */*
Content-type: application/x-www-form-urlencoded
Content-length: 55

Textfield1=Entry+1&Textfield2=Entry+2&SubmitButton=Send
```

Usare Java Applets per mandare dati via GET a programmi CGI

```
String someData =
  name1 + "=" + URLEncoder.encode(val1) + "&" +
  name2 + "=" + URLEncoder.encode(val2) + "&" +
  ...
  nameN + "=" + URLEncoder.encode(valN);
try {
  URL cgiURL = new URL(baseURL + "?" + someData);
  getAppletContext().showDocument(cgiURL);
} catch(MalformedURLException mue) {...}
```

Programmazione CGI - parte server

CGI interaction process

Ci si aspetta che il programma CGI possa fare solo due cose:

- generare al volo una pagina HTML, che poi sarà restituita dal server al client;
- indicare al server una pagina preesistente da restituire al client.

Un programma CGI (output) dovrebbe eseguire i seguenti passi:

1. leggere i dati (inviati con `GET` o `POST`)
2. generare di HTTP headers (mimare parzialmente HTTP server)
3. inviare una riga bianca
4. generare un documento (nella maggior parte dei casi un documento HTML)

Esempio molto semplice:

```
#!/bin/sh
echo "Content-Type: text/plain"
echo ""
echo "Hello, world."
```

Programmazione CGI - parte server

Le variabili d'ambiente

Le variabili di ambiente sono utilizzate per la comunicazione dal server Web al programma CGI

QUERY_STRING la lista di coppie nome=valore, usata con il metodo `GET`

```
#!/bin/sh
echo "Content-Type: text/plain"
echo ""
echo "Query String is ´$QUERY_STRING´"
```

PATH_INFO informazioni aggiuntive messe dopo l'URL (usate per esempio con le clickable map)

CONTENT_TYPE usata con il metodo `POST`, indica il tipo MIME delle informazioni che arriveranno

CONTENT_LENGTH usata con il metodo `POST`, indica quanti byte arriveranno sullo standard input del programma CGI

```
http://lyra.educ.disi.unige.it/~gbelle/cgi-bin/
  /test-cgi.cgi?a=gianna&b=gianna
```

Programmazione CGI - parte server

Le variabili d'ambiente

```
#!/bin/sh

echo Content-type: text/plain
echo
echo CGI test script report:

echo SERVER_SOFTWARE = $SERVER_SOFTWARE
echo SERVER_NAME = $SERVER_NAME
echo GATEWAY_INTERFACE = $GATEWAY_INTERFACE
echo SERVER_PROTOCOL = $SERVER_PROTOCOL
echo SERVER_PORT = $SERVER_PORT
echo REQUEST_METHOD = $REQUEST_METHOD
echo HTTP_ACCEPT = "$HTTP_ACCEPT"
echo HTTP_USER_AGENT = "$HTTP_USER_AGENT"
echo PATH_INFO = $PATH_INFO
echo PATH_TRANSLATED = $PATH_TRANSLATED
echo SCRIPT_NAME = $SCRIPT_NAME
echo QUERY_STRING = $QUERY_STRING
echo REMOTE_HOST = $REMOTE_HOST
echo REMOTE_ADDR = $REMOTE_ADDR
echo REMOTE_USER = $REMOTE_USER
echo CONTENT_TYPE = $CONTENT_TYPE
echo CONTENT_LENGTH = $CONTENT_LENGTH
echo AUTH_TYPE = $AUTH_TYPE
echo DOCUMENT_ROOT = $DOCUMENT_ROOT
echo REMOTE_IDENT = $REMOTE_IDENT
```

Programmazione CGI - parte server

CgiGet script - chiamata programma Java

```
#! /bin/csh

/share/pub/jdk/bin/java -classpath /share/pub/jdk:/sh
 are/pub/jdk/lib/classes.zip:/home/stud/gbelle/public
 _html/Examples CgiGet "$QUERY_STRING"
```

CgiGet.java

```
public class CgiGet {
public static void main(String[] args) {
    CgiGet app = new CgiGet("CgiGet", args);
    app.printFile();
  }

  private String name;
  private String[] args;
  protected String type = "GET";

  public CgiGet(String name, String[] args) {
    this.name = name;
    this.args = args;
  }

  public void printFile() {
    printHeader(name);
    printBody(args);
```

```
    printTrailer();
  }

  protected void printHeader(String name) {
     System.out.println
        ("Content-Type: text/html\n" +
         "\n" +
         "<HTML>\n" +
         "<HEAD>\n" +
         "<TITLE>The " + name + " Program</TITLE>\n" +
         "</HEAD>\n" +
         "\n" +
         "<BODY>\n" +
         "<H1>The " + name + " Program</H1>\n" +
         "This program blah blah " + type + " requests.\n" +
         "<P>");
  }


  protected void printBody(String[] data) {
      System.out.println("Data was ´" + data[0] + "´.");
  }

  protected void printTrailer() {
      System.out.println("</BODY>\n</HTML>");
  }
}
```

Passare le variabili come parametri con nome

Quando si devono passare variabili multiple, non è molto conveniente elencarle come argomenti in linea.

Un'alternativa è di fornire le variabili come proprietà, consentendo al programma Java di individuarle tramite nome. Le entries sono passate all'interprete Java via "-Dname=value", e possono essere accedute dal programma via `System.getProperty(``name´´)`, che ritorna una stringa

```
java -DVAR1="$VAR1" -DVAR2="$VAR2" ... CgiProgram
```

PropertyTest.java

```
public class PropertyTest {
  public static void main(String[] args) {
    System.out.println
     ("Java: " + System.getProperty("java") + "\n" +
      "CGI: " + System.getProperty("cgi") + "\n" +
      "C++: " + System.getProperty("cpp"));
  }
}
```

Output

```
> java -Djava=cool -Dcgi=hot -Dcpp=stale PropertyTest
Java: cool
CGI: hot
C++: stale
```

Leggere dati POST

Contro:

- un Java applet non può mostrare una pagina generata da un programma CGI
- una richiesta POST non può essere generata semplicemente aprendo un particolare URL in un WWW browser

Pro:

- non limite alla lunghezza dei dati
- informazioni private dato che non appaiono nel URL
- ...

CgiPost script

```
#! /bin/csh

/share/pub/jdk/bin/java -classpath /share/pub/jdk:/sh
 are/pub/jdk/lib/classes.zip:/home/stud/gbelle/public
 _html/Examples CgiPost
```

CgiPost.java

```
import java.io.*;

public class CgiPost extends CgiGet {
  public static void main(String[] args) {
    try {
      DataInputStream in = new DataInputStream(System.in);
      String[] data = { in.readLine() };
      CgiPost app = new CgiPost ("CgiPost", data);
      app.printFile();
    } catch(IOException ioe) {
      System.out.println
        ("IOException reading POST data: " + ioe);
    }
  }

  public CgiPost(String name, String[] args) {
    super(name, args);
    type = "POST";
  }
}
```

Interazione

1. il client invia una richiesta contenente
   - identità del programma CGI;
   - dati della form;
2. il server riceve i dati;
3. il server lancia il programma CGI e gli passa i dati;
4. il programma CGI estrae dai dati i valori immessi dall'utente;
5. il programma CGI usa tali valori per portare avanti il suo compito (es. lanciare a sua volta un'altra applicazione, chiedendole qualcosa)
6. l'applicazione restituisce dei risusltati al programma CGI;
7. il programma CGI utilizza tali risultati per confezionare al volo una pagian HTML che restituisce al server;
8. il server invia tale pagina al client, che la visualizza sullo schermo.

Alternative al CGI

CGI è un metodo flessibile ma poco efficiente per connettere pagine Web a database ed altri programmi esterni. Il motivo principale è che il server HTTP deve creare un nuovo programma o aprire una nuova connessione al database per ogni richiesta.

Java servlet API è un piccolo programma Java che gira sul server. Il vantaggio è che possono essere eseguiti più velocemente delle applicazioni CGI dato che non creano un processo ma un thread.

Server APIs

Server-Side Includes una SSI è un valore di una variabile che il server può includere in un file HTML. Può essere considerata come una forma limitata di CGI

LiveWire:Server-Side JavaScript

direct socket connections (including JDBC)