

Interfete între clientii Web si servere WWW. Script-uri

Asist.drd. Carmen STANCIU,
Catedra de Informatica Economica, A.S.E., Bucuresti

Transformarea paginilor Web din statice în dinamice se realizeaza prin metode standardizate, atât pe partea de server (prin interfete browser-server Web), cât si pe partea de client (prin scripturi).

Cuvinte cheie: CGI, Perl, ISAPI, NSAPI, JavaScrip, ECMA Script.

CGI (Common Gateway Interface).

Una dintre metodele prin care putem transforma o pagina Web statica în una dinamica, este CGI. CGI-ul ruleaza pe partea de server si reprezinta o interfata standard între serverul HTTP si un program extern.

Utilizatorii interactioneaza cu o pagina Web programata, completând un formular HTML si trimitând la o aplicatie CGI. Aplicatia primeste datele de intrare, executa operatii si, eventual, returneaza un rezultat. Acest tip de aplicatie se regaseste la operatiile cu bazele de date, cautari sau vizualizari de cataloage.

Dezavantajul programelor CGI este faptul ca sunt lente, mai ales daca sunt scrise într-un limbaj interpretat lent, cum ar fi Perl, si nu în cod compilat C. Unele sisteme lanseaza mai multe copii ale aceleasi aplicatii CGI, consumând resurse sistem. Din aceasta cauza s-a cautat dezvoltarea de aplicatii pe partea server-ului, cum ar fi NSAPI (Netscape API), MSAPI (Microsoft Internet Server API), care pot elimina lipsurile CGI-ului, serverul ramânând în continuare sa faca toata munca.

Un script CGI este un program care, împreuna cu serverul HTTP, raspunde cererii clientului. El contine un identificator universal pentru o resursa, o metoda, precum si parametri si/sau date care determina mai exact cererea.

Comunicatia dintre CGI si aplicatie se realizeaza prin patru mijloace: **variabile de mediu** (parametrii abstracti pe care serverul îi pune la dispozitia script-ului CGI si

care pot fi continuti de fisiere .INI din Windows sau variabile de mediu Unix), **linie de comanda** (poate fi o linie de comanda pentru lansarea unui program sau un fisier structurat ca vectori de siruri), **intrarea standard** (reprezinta echivalentul intrarii standard din Unix), **iesirea standard** (reprezinta echivalentul iesirii standard din Unix).

Modul de lucru al unui script CGI este urmatorul:

- browser-ul va cere URL-ul de la server, la fel ca orice alt document;
- server-ul receptioneaza cererea (sesizeaza ca URL-ul specifica un script, dupa extensie sau pozitia dintr-un director anume) si apoi executa script-ul;
- script-ul efectueaza operatia ceruta (interogarea unei baze de date, calcularea unei valori, apelarea unui program etc.) pe baza datelor furnizate de browser;
- script-ul transforma rezultatul într-un format înteles de server-ul Web;
- server-ul Web receptioneaza rezultatul script-ului si îl trimite browser-ului, care îl formateaza si îl afiseaza.

PERL (Practical Extraction and Report Language)

Limbajul Perl, aparut spre sfârșitul anilor '80, a fost scris de catre *Larry Wall* în scopul usurarii unor sarcini administrative. Este un limbaj interpretat, compilat în timpul încarcării, optimizat pentru scanarea si extragerea de informatii din fisierele text si tiparirea de rapoarte. Este simplu de utilizat, eficient si complet, bun pentru multe

activitati de administrare a unui sistem Unix, precum si pentru crearea de scripturi CGI.

Perl combina cele mai bune caracteristici de la **sh**, **awk**, **sed**, având unele asemanari cu **Pascal** si **csh**. Are posibilitatea de a apela functii recursive de orice adâncime si un mecanism eficient de cautare si înlocuire într-un text, dupa un model dat .

În Perl, munca programatorului este limitata la capacitatile hardware ale calculatorului. Astfel, în functie de memoria calculatorului, se poate citi un fisier întreg într-o singura variabila de tip *string*.

Începând cu versiunea 5 se poate folosi programarea modulara si orientata obiect. Comparativ cu C, în Perl programele *setuid* sunt mult mai sigure, datorita unui mecanism de urmarire a evolutiei datelor, care poate fi activat cu parametrul *-T* din linia de comanda. În acest fel nu vor fi executate apelurile de sistem care se bazeaza pe informatii primite de utilizatori si care necesita o securitate mare.

Ca sa schimbam o variabila de mediu în Perl se actioneaza asupra tabloului asociativ %ENV. Variabilele de mediu comune Unix-ului si Perl-ului sunt: PATH, IPS, SHELL. În Perl, mediul în care ruleaza un program CGI poate fi controlat prin setarea unor variabile, care au valori initiale date de serverul de Web.

În ceea ce priveste apelurile sistem, acestea pot lua diferite forme, astfel: apelând functia **"system"** (*system ("record_type \$browser")*), **'nume_program'** pus între apostrofuri (*'record_type \$browser'*), executând **"open"** (*open(P_RECORD, 'record_type \$browser'|" || die)*).

Introducerea unui script Perl într-un formular HTML se poate realiza astfel:

```
<FORM METHOD=Post ACTION= "http://server_name/cgi-bin/scriptname.pl">
```

Legatura dintre Perl si bazele de date, via CGI se poate realiza, printre altele, si prin interfata **Perl DBI** (DataBase Interface), disponibila la <http://www.hermetica.com/tehnologia/DBI>. DBI are versiuni

pentru diferite baze de date si permite scrierea de script-uri independente de baza de date. DBD (DataBase Driver) este utilizat de DBI pentru a translata apelurile API DBI în comenzi specifice bazei de date. Un exemplu freeware de SGBD relational sub Unix este MySQL (<http://www.tcx.se>), care poate administra baze de date cu milioane de înregistrari. Acesta suporta Perl BDI, JDBC si JDBC. Interogarea dintre baza de date si browser-ul Web se realizeaza printr-o interfata, un formular HTML, care transmite cereri bazei de date prin DBI/DBD. Apoi, script-ul Perl preia informatia din acest formular si construiește o fraza SQL, care este trimisa bazei de date.

◆ ISAPI (Internet Server API)

Internet Server API (ISAPI) este elaborat de Microsoft în scopul de a înlocui "batrânul" CGI, care a devenit lent si greu de utilizat în conditiile unor cereri HTTP multe si intense.

Fiind orientat pe partea de server, spre deosebire de Java, ActiveX si VBScript care sunt tehnologii Web pe partea client, ISAPI este o îmbunatatire adusa CGI-ului care este ineficient deoarece starteaza un proces de fiecare data cât este apelat, ceea ce duce la costuri ridicate în Windows NT. Pe de alta parte, CGI-ul este dificil pentru dezvoltarea de aplicatii mari client/server si profesionale.

Cadrul de lucru al ISAPI consta în definirea unei metode DLL care va fi pusa pe server-ul IIS (Internet Information Server). Extensia ISAPI, un DLL ce exporta functii specifice în concordanta cu specificatiile ISAPI, este de doua tipuri:

- **aplicatii ISAPI** – din punct de vedere conceptual au loc în interiorul proceselor si pot fi folosite pentru îmbunatatirea paginilor HTML si furnizarea de date dinamice (cum ar fi CGI);
- **filtre ISAPI** – din punct de vedere conceptual sunt situate între serverul HTTP si socket-ul HTTP; pot fi utilizate

pentru adaugarea de noi scheme de autentificare, suport pentru noi metode de criptare sau compresie, schimbarea continutului paginii clientului sau a unor scheme de autentificare, furnizarea unui istoric avansat al login-urilor etc.

Prin apelarea unei extensii ISAPI server de la un browser, prin URL este identic cu chemarea unui script CGI, chiar daca cele doua arhitecturi sunt diferite. Diferentele dintre cele doua arhitecturi sunt la nivel de server, browser-ul trimitând acelasi tip de date si primind informatii în acelasi format (astfel browser-ul ramâne independent de software-ul care ruleaza pe server).

◆ NSAPI (Netscape Server Applications Programming Interface)

NSAPI reprezinta API-ul elaborat de Netscape pentru a implementa SAF (Server Application Functions), adica functiile specifice nucleului server-ului Enterprise Server.

Aceasta interfata permite cererilor provenite de la server sa fie divizate în parti mai mici, care pot fi aranjate într-o varietate de forme.

NSAPI a fost proiectat de creatorii serverelor Web NCSA si CERN, fiind testat în cele mai diverse moduri. Este mult mai rapid si mai eficient decât CGI-urile, putând rula si în medii distribuite bazate pe IIOP (Internet Inter ORB Protocol).

NSAPI permite dezvoltatorilor de programe sa extinda posibilitatile native oferite de un server Web. Cadrul sau de lucru este determinat de CGI si server-e bazate pe API-uri Java si JavaScript.

Prin utilizarea NSAPI, utilizatorul are definit un mecanism de control al accesului, o autorizare si un istoric al proceselor rulate.

Idea care a facut ca NSAPI sa înlocuiasca CGI este aceea ca pentru fiecare functie CGI se creeaza un nou proces, ceea ce duce la supraîncarcarea serverului Web în conditiile a mii de cereri. Un program care utilizeaza NSAPI este doar o functie de biblioteca care ruleaza în cadrul aceluasi

context de proces ca si serverul de Web, eliminând astfel supraîncarcarile datorate crearii de procese.

Script-uri

Script-urile sunt limbajele care au deschis drumul spre paginile Web interactive, rulând doar pe partea clientului, spre deosebire de CGI-ri care rulau pe server.

Primul limbaj de scripting a fost introdus de Sun, care a creat si limbajul Java. Alaturi de Sun s-a înscris în cursa scripting-urilor si Microsoft, care a scos pe piata propria versiune JScript. Standardul în domeniu se pare ca-l va cuceri ECMA.

◆ JavaScript

Limbajul JavaScript a aparut din dorinta de a schimba statutul celui care vizualizeaza paginile Web, dintr-un simplu vizitator care nu poate interveni asupra continutului paginilor, într-un utilizator de Web activ.

Este creat de Netscape si apare pentru prima data în Navigator 2.0. Are doua forme primare, si anume:

- partea interpretata de server (server-side), rezervata exclusiv produselor Netscape LiveWire,
- partea interpretata de client (client-side), accesibila utilizatorilor Web prin intermediul paginilor HTML si contine o colectie de comenzi incluse între tag-urile `<SCRIPT LANGUAGE = "JavaScript">.....</SCRIPT>`. Poate fi amplasat în tag-ul `<HEAD>`, în tag-ul `<BODY>` si în tag-uri HTML.

Browser-ul care întâlnește pagini cu elemente de JavaScript, daca cunoaste JavaScript, interpreteaza comenzile linie cu linie si le executa, iar în caz contrar, le trateaza ca si cum ar fi comentarii HTML (de aceea trebuie ascunse).

JavaScript este interpretat diferit de catre Netscape si Internet Explorer, Microsoft având dezvoltat JScript.

JavaScript se utilizeaza pe partea de client pentru verificarea locala a datelor introduse într-o fereastră, crearea de aplicatii

stand-alone sau pentru comunicarea cu applet-urile Java si cu aplicatiile auxiliare care ruleaza direct din Netscape. Prezinta caracteristici care sunt utile aplicatiilor bazate de browsere, cum ar fi: calcule de uz general, structuri de date de uz general, generare de continut bazat pe DHTML, controale de timp etc. Nu ofera suport pentru grafica, trebuind sa combine capacitatile oferite de browser si cele ale HTML-urilor pentru afisarea rezultatelor.

JavaScript are limitari în ceea ce priveste comunicarea cu server-ul si stocarea datelor. Nu dispune de variabile globale sau domenii de vizibilitate pentru functii, dar se bazeaza pe un model complex de obiecte ale browser-elor.

JavaScript organizeaza toate elementele unei pagini Web într-o ierarhie, fiecare element fiind vazut ca un obiect, iar obiectul având proprietati si metode.

La baza limbajului stau trei blocuri functionale, si anume: valori (tipuri de date suportate de JavaScript), obiecte (colectii de proprietati care pot fi apelate cu un singur nume), functii (proceduri care pot fi executate de o aplicatie; functiile asociate unui obiect se numesc metode).

Pentru a realiza legatura cu un applet, JavaScript poate apela metoda `init()` a applet-ului, astfel:

```
public init()
{ win=JSObject.getwindows(this);
  .....
}
```

Plug-in-urile, mici resurse de cod care ruleaza în interiorul procesului navigatorului si care extind capacitatile browser-elor, sunt facute pentru a interpreta fisiere de un anumit tip (.avi, .mov) la un cost de dezvoltare si pastrare minim. Pentru a permite unui applet sau unui plug-in sa acceseze variabilele si functiile JavaScript dintr-o pagina, trebuie sa apara cuvântul cheie `MYSCRIPT` în interiorul tag-ului `<APPLET>` sau `<EMBED>`.

◆ ECMA Script

ECMA este o asociatie a industriei europene care are printre scopurile sale si dezvoltarea unui limbaj script standardizat, multiplatforma si independent de furnizor.

În 1997 a fost lansat *ECMAScript 262*, un produs care își are originea în *JavaScript* de la Netscape si *JScript* de la Microsoft, fiind considerat un standard deschis.

Codul ECMAScript consta din declaratii individuale, blocuri de declaratii si comentarii. Contine sase tipuri de date, astfel:

- **numar** – este conform cu standardul IEEE-754 de reprezentare a valorilor numerice si cuprinde numere întregi, în virgula mobila, infinit pozitiv si negativ, zero pozitiv si negativ;

- **sir de caractere** – face distinctie între sirurile cu *apostrof simplu sau dublu*;

- **boolean** – poate lua valorile *adevarat* (orice valoare diferita de 0) sau *fals* (valoare 0);

- **nedefinit** – este un tip special de date care reprezinta valoarea pentru o variabila, dupa ce aceasta a fost creata, dar înainte de a lua valoarea data prin program;

- **null** – arata ca nu sunt date valide; se poate defini explicit valoarea *null* pentru o variabila, sau aceasta poate rezulta dintr-o expresie;

- **obiect** – instanta oricarui tip se creeaza cu operatorul *new*; suporta trei tipuri de obiect:

- *primul tip de obiecte* – contine 9 obiecte implicite, cum ar fi *Array*, *Number*, *String* etc;

- *al doilea tip de obiecte* – include cele proprii create de o aplicatie;

- *al treilea tip de obiecte* – rezulta din mediul de lucru al gazdei pentru ECMAScript; daca de foloseste pentru lucrul cu Web acest mediu este browser-ul.

In continuare este ilustrat un exemplu de utilizare ECMAScript, având ca efect afisarea datei si orei, cu un mesaj corespunzator momentului zilei.

```

<html><script language=javascript>
var d = new Date()
var h = d.getHours()
document.write("<BR> According to your clock,")
if (h<12)
    document.write("it is" + h + ":" + d.getMinutes() + " in the mornig")
else
    if (h<17)
        document.write("it is" + h + ":" + d.getMinutes() + " in the afternoon")
    else
        document.write("it is" + h + ":" + d.getMinutes() + " in the evening")
</script></html>

```

◆ Scriptlet-uri

Scriptleturile, solutii ale Microsoft alaturi de DHTML pentru crearea paginilor Web interactive, permit încapsulare DHTML sau HTML în componente reutilizabile.

Când se creaza un scriptlet trebuie sa se lucreze cu doua fisiere: unul care contine scriptlet-ul, iar altul de tip .html, Word sau orice altceva care suporta obiecte COM (Component Object Model), care va fi gazda a scriptlet-ului.

Pentru descarcarea scriptlet-ului se va utiliza ca browser cel puțin *Internet Explorer 4*, care trebuie sa aiba setat nivelul de securitate la scazut sau mediu.

Pentru proiectarea scriptlet-urilor se folosesc doua metode, pentru fiecare trebuind creata o functie:

- **metoda prefix** – valabila si cu JScript si VBScript, numita astfel datorita modului în care sunt marcate functiile si variabilele

```

<OBJECT ID="TestS1" TYPE="text/x-
STYLE="position:absolute; top=75;left=0">
<PARAM NAME =url value="Simple.html">
</OBJECT></HTML>

```

Exemplu de scriptlet:

```

<HTML> <HEAD>
<TITLE> Exemplu de scriptlet </TITLE>
<SCRIPT>
function public_setUnit()
{
    if (document.all.Unit.innerText=="word")
    {
        document.all.Unit.innerText ="sentence" ;

```

care se doresc a fi expuse; trebuie inserat cuvântul *public_* înaintea numelui de functie;

- creaza un obiect *Public_Description*, care ofera acces la metodele si proprietatile expuse, tehnica care functioneaza doar în JScript.

În continuare este prezentat un exemplu de scriptlet, scriptlet-ul DHTML fiind definit în blocul `<script> ... </script>`.

Fisierul gazda va avea eticheta OBJECT care va face referire la scriptlet prin urmatoorii parametri: nume de identificare pentru invocarea scriptlet-ului (TestS1), PARAM, NAME sau DATA – specifica fisierul scriptlet, TYPE – atribut care specifica tipul MIME al scriptlet-ului si i se da valoarea test/x-scriptlet, ID – reprezinta obiectul în cadrul gazdei.

```

scriptlet" WIDTH=400 HEIGHT=250

```

```

}
    else        document.all.Unit.innerText = "word" ;
}
function selectMe()
{
    var r= document.body.createTextRange();
    r.moveToPoint(window.event.x, window.event.y);
    if (document.all.Unit.innerText== "sentence")
    {
        r.expand("sentence");
    }
    else        r.expand( "word");
    r.select() ;
}
</SCRIPT>
</HEAD>
<BODY>
<P onclick=public_setUnit() >
Alege între cuvânt si fraza. </P>
<H1 onclick = selectMe() id=myH1>
Selectează <SPAN ID = "Unit"> cuvântul </SPAN> si va fi subliniat.
</BODY></HTML>

```

Bibliografie

- *** Netscape.com/kb/server/960513-119.html, 1998
- *** Developer.netscape.com/docs/manuals/enterprise/nsapi/, 1998
- *** Help.netscape.com/kb/server/971112-3.html, 1998
- *** Learning.netscape.com/courses/desc/nsapi.htm, 1998
- *** Developer.netscape.com/docs/examples/nsapi.html, 1998
- *** Www.genusa.com/isapi/, 1998
- *** Www.webhub.com/, 1998
- *** Www.webadvisor.com/isapi.html, 1998
- *** Www.genusa.com/isapi/c40inst.html, 1998
- *** Www.justpc.com/iaxfaq.html, 1997
- *** Www.aaa.org/sld012.htm, 1997
- *** Distributed Object Computing in the Internet Age, www.visigenic.com/pdf/visbrok.pdf, 1998
- *** IOP and the Distributed Objects Model, home.netscape.com/comprod/columns/techvision/iop.html, 1998
- /COX98/** COX, T., Perl si bazele de date, BYTE România, mai, 1998
- /CSAB97/** CSABA, P., Programare CGI sigura în Perl, BYTE România, iulie, 1997
- /DECE96/** DECEMBER, J., Prezentarea limbajului Java, Editura Teora, Bucuresti, 1996
- /DOBS98/** DOBSON, R., ECMAScript standardul salvator?, BYTE România, august, 1998
- /MATE98/** MATTESCU, C., Introducere în JavaScript, PC Report, sept., 1998
- /MATE98/** MATEI, C., Visual Page, PC Report, aug., 1998
- /MATE98/** MATEI, C., Adobe PageMill 3.0, PC Report, apr., 1998
- /MATE98/** MATEI, C., Prezentari Java – Medii de dezvoltare, PC Report, martie, mai, sept., oct., 1998;
- /NAST98/** NASTASE, F., Internet, World Wide Web, JavaScript, HTML, Java, Editura Economica, Bucuresti, 1998
- /PROD97/** PRODAN, A., s.a., Mediul Java pentru Internet, Editura Promedia Plus, Cluj-Napoca, 1997
- /RADO96/** RADOIU, D., Interactivitate cu JavaScript, BYTE România, aug., 1998

/RENS98/ RENSCHAW, D., Faceti componente portabile cu JavaBeans, BYTE România, feb., 1998;

/ROSC00/ Ion Gh. Rosca, Nicolae Tapus, Valentin Cristea, Irina Atanasiu, Bogdan Costinescu, Gavril Gozdea, Floarea Nastase, Stanciu Carmen, Octavian Paiu – *INTRA-NET – partea a II-a-*, Editura ASE, Bucu-resti, 2000

/ROSC00/ Ion Gh. Rosca, Nicolae Tapus, Valentin Cristea, Irina Atanasiu, Bogdan Costinescu, Gavril Gozdea, Floarea Nastase, Stanciu Carmen, Octavian Paiu - *Solutii arhitecturale pentru subretele de comunicatii utilizate în INTRANET si solutii software suport pentru activitati cooperative în INTRANET*, faza 2.1. - *Solutii pentru protectia si securizarea unui INTRANET*, Contract nr. 131/1999, Act aditional nr. 46/I/18.01.2000, Beneficiar - Agentia Nationala pentru Stiinta, Tehnologie si Inovare, 2000

/ROSC99/ Ion Gh. Rosca, Nicolae Tapus, Valentin Cristea, Irina Atanasiu, Bogdan Costinescu, Gavril Gozdea, Floarea Nastase, Stanciu Carmen, Octavian Paiu – *INTRANET* - (133 pagini), Editura ASE, Bucu-resti, 1999

/ROSC99/ Ion Gh. Rosca, Stanciu Carmen, Octavian Paiu, s.a. - *Noi paradigme ale managementului modern al întreprinderii - Intranet*, Contract nr. 18, cod CNCSIS 55/1999, beneficiar Ministerul Educatiei Nationale, 1999

/ROTA96/ ROTARIU, E., Limbajul Java, Computer Press Agora SRL, 1996

/SWOD97/ SWOBODA, N., HTML - based Interfaces, cs.indiana.edu, feb.,1997

/VAID98/ VAIDA, C., Publicatii Web, CHIP, nov., 1998