



Servlets & JDBC

Catalin Stoean

Continut

- **Ce este un servlet**
- De ce avem nevoie de servlet
- Structura unui servlet
- Instalarea
- Rularea unui servlet
- Metodele doGet si doPost
- Accesarea unei BD SQL Server dintr-un servlet

Ce este un servlet?

- Servletii reprezinta programe care ruleaza pe un server de web
- Sevletii genereaza in mod dinamic fisere HTML bazate pe cererea clientilor
- Extinde in mod dinamic functionalitatea unui server
- Opereaza asupra serverului *in culise*, numai rezultatele fiind intoarse utilizatorului

Continut

- Ce este un servlet
- **De ce avem nevoie de servlet**
- Structura unui servlet
- Instalarea
- Rularea unui servlet
- Metodele doGet si doPost
- Accesarea unei BD SQL Server dintr-un servlet

Atuuri ale servletilor

■ Eficienta:

- poate folosi *thread-uri* din Java pentru a rula mai multe procese odata.
- raman activi si dupa ce raspunsul a fost intors.
- ofera o infrastruttura pentru decodificare datelor trimise dintr-un HTML.

■ Portabilitate:

- poate rula pe mai multe masini de calcul.

■ Securizare buna a datelor

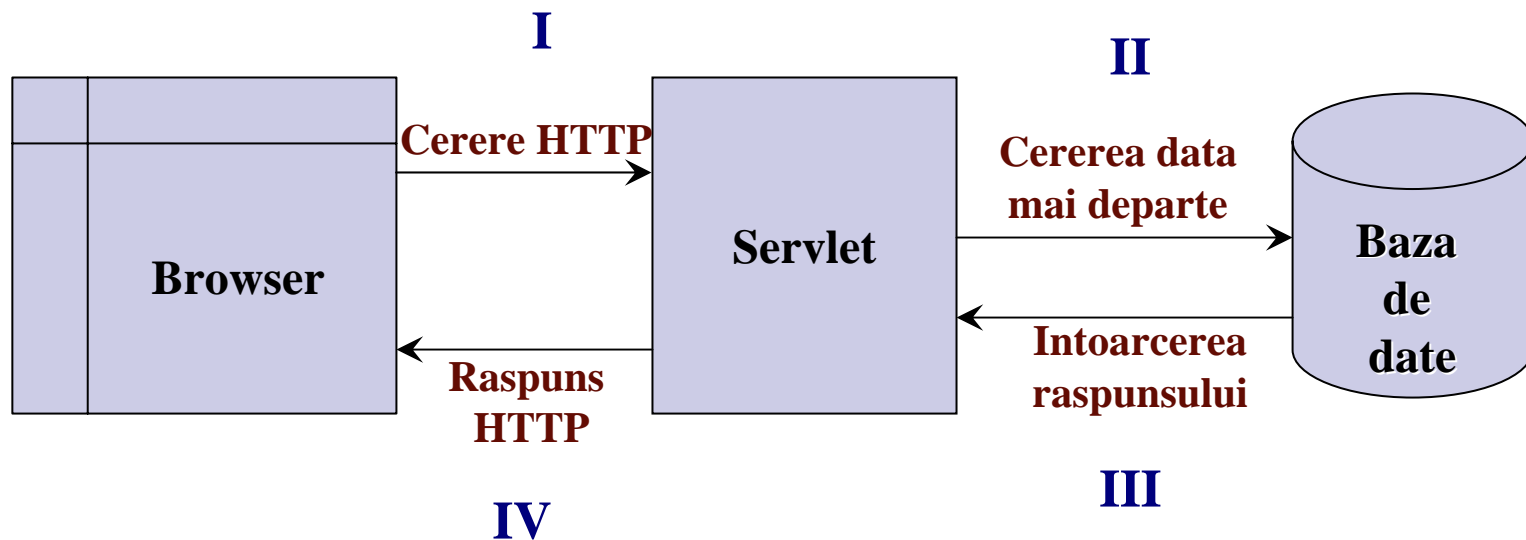
Ce mai pot face?

- Procesarea sau stocarea datelor informatiilor date intr-o forma HTML.
- Proceseaza intrarile utilizatorului si intoarce un raspuns corespunzator sub forma unui fiser HTML – poate intoarce raspunsul primit printr-un *query* la o baza de date.
- Faciliteaza comunicarea (realizarea unui forum).
- Suporta interactiuni intre clienti (jocuri multiplayer).

Continut

- Ce este un servlet
- De ce avem nevoie de servlet
- **Structura unui servlet**
- Instalarea
- Rularea unui servlet
- Metodele doGet si doPost
- Accesarea unei BD SQL Server dintr-un servlet

Modul de lucru al unui serlet



Cererea si raspunsul HTTP

- HTTP: **H**yper**T**ext **T**ransfer **P**rotocol reprezinta un protocol folosit de catre WWW.
- In fiecare cerere HTTP este specificata o metoda:
 - Metoda **GET** – este folosita in principal pentru a primi informatii, iar cererea este transmisa serverului sub forma unui URL.
 - Metoda **POST** – folosita pentru a trimite date catre server; datele sunt incapsulate intr-un pachet si trimise catre server.

Structura unui servlet

```
public class ServletBD extends HttpServlet {  
  public void init(){  
  public void doGet(HttpServletRequest cerere,  
    HttpServletRequest raspuns) throws  
    ServletException, IOException {//...  
  }  
  public void doPost(HttpServletRequest cerere,  
    HttpServletRequest raspuns) throws  
    ServletException, IOException {//...  
  }  
  public void destroy(){}
```

Continut

- Ce este un servlet
- De ce avem nevoie de servlet
- Structura unui servlet
- **Instalarea**
- Rularea unui servlet
- Metodele doGet si doPost
- Accesarea unei BD SQL Server dintr-un servlet

Ce trebuie sa instalam?

- Pentru a crea si a rula servleti, este nevoie de instalarea [JavaServer™ Web Development Kit](#).
- Alte posibilitati de a rula servleti constau in instalarea:
 - [Apache Tomcat](#)
 - [Allaire JRun](#)
 - [New Atlanta's ServletExec](#)

Configurarea

- Dupa descarcarea arhivei JSWDK, dezarhivam intr-un director, sa zicem **F:\jswdk-1.0.1**.
- Daca JDK este instalat in **F:\jdk1.4.2**, pentru a ne asigura ca servletii vor functiona, avem nevoie sa copiem din **F:\jswdk-1.0.1\lib** pachetul **servlet.jar** in **F:\j2sdk1.4.2\jre\lib\ext**.
- Mai departe, putem compila un servlet fara sa ne mai apara o sumedenie de erori.

Continut

- Ce este un servlet
- De ce avem nevoie de servlet
- Structura unui servlet
- Instalarea
- **Rularea unui servlet**
- Metodele doGet si doPost
- Accesarea unei BD SQL Server dintr-un servlet

Structura unui servlet

- Pentru a fi un servlet, o clasa trebuie sa mosteneasca *HTTPServlet* si sa suprascrie *doGet* sau *doPost* sau ambele.
- Metodele au doua argumente:
 - *HttpServletRequest* – are metode care permit primirea datelor;
 - *HttpServletResponse* – are metode cu care specificam tipul de raspuns pe care il dam si, mai ales, permite obtinerea unui obiect de tip *PrintWriter* pe care il folosim pentru a trimite raspunsuri pentru clienti.

Exemplu banal

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class UnServlet extends HttpServlet {
    public void doGet(HttpServletRequest cerere,
        HttpServletResponse raspuns)
        throws ServletException, IOException {
        PrintWriter out = raspuns.getWriter();
        out.println("Uite ca a realizat afisarea!!");
    }
}
```

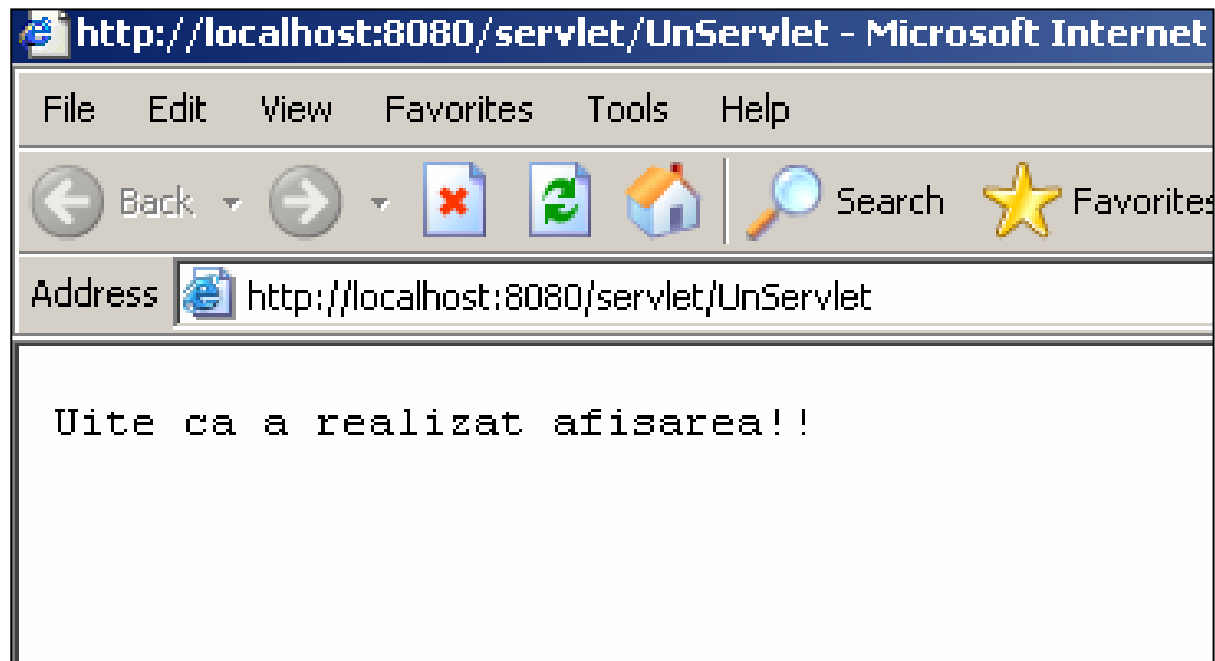

Pregatirea rularii

- Compilarea fisierului se face ca pentru o aplicatie java obisnuita.
- Clasa rezultata de la compilare trebuie copiata apoi in directorul:
F:\jswdk-1.0.1\webpages\WEB-INF\servlets
- Inainte de a rula aplicatia, avem nevoie sa pornim serverul: in directorul **F:\jswdk-1.0.1** rulam **startserver.bat**

Rularea

- Deschidem un browser (Internet Explorer) si scriem:

http://localhost:8080/servlet/UnServlet



Servlet care genereaza HTML

- Majoritatea servletilor genereaza HTML, si nu text precum cel anterior – pentru aceasta este nevoie de doi pasi in plus:
 - Sa spunem browser-ului ca generam un HTML – aceasta se face prin metoda *setContentType* a obiectului de tip *HttpServletResponse*.
 - Al doilea pas consta in modificarea datelor pe care le afisam pentru a construi corect o pagina HTML

Exemplu

```
public void doGet(HttpServletRequest cerere,  
                  HttpServletResponse raspuns)  
    throws ServletException, IOException {  
    raspuns.setContentType("text/html");  
    PrintWriter out = raspuns.getWriter();  
    out.println("<HTML>\n<HEAD><TITLE>" +  
        "HTML generat din Servlet" +  
        "</TITLE></HEAD>\n<BODY>\n" +  
        "<H1>lata ca functioneaza si asta!</H1>\n" +  
        "</BODY></HTML>");  
}
```

Continut

- Ce este un servlet
- De ce avem nevoie de servlet
- Structura unui servlet
- Instalarea
- Rularea unui servlet
- **Metodele doGet si doPost**
- Accesarea unei BD SQL Server dintr-un servlet

Rescrierea metodelor *doGet* si *doPost*

Get si *Post* sunt denumite astfel din punctul de vedere al clientului

- Daca un client cere date de la server, este indicat ca browserul sa trimita o cerere de tip GET catre servlet care face ca metoda *doGet* sa fie apelata.
- Cand clientul doreste doar sa trimita date catre servlet, este indicat ca browserul sa trimita o cerere POST catre servlet care face ca metoda *doPost* sa fie cea apelata.

Exemplu de utilizare a metodei *doPost*

```
<HTML>
<BODY>
<FORM ACTION="http://172.16.0.1:8080/servlet/PostEx"
METHOD="POST">
Nume:
<INPUT TYPE="text" NAME="nume"><br>
E mail:
<INPUT TYPE="text" NAME="email"><br>
<INPUT TYPE=reset> <INPUT TYPE=submit VALUE="Send">
</FORM>
</BODY>
</HTML>
```

Metoda doPost

```
public void doPost(HttpServletRequest cerere,  
    HttpServletResponse raspuns) throws ServletException,  
    IOException {  
    raspuns.setContentType("text/html");  
    PrintWriter out = raspuns.getWriter();  
    out.println("<html> <head><title> Forma Post </title>" );  
    out.println("<body>");  
    out.println("<h3>Salutare, " );  
    out.println(raspuns.getParameter("nume") + "!</h3>");  
    out.println("Adresa ta de e mail... ");  
    out.println(raspuns.getParameter("email"));  
    out.println("</body></html>");  
    out.close();  
}
```


Rezultate

C:\jswdk-1.0.1\webpages\post.html - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search

Address C:\jswdk-1.0.1\webpages\post.html

Nume:

E-mail:

Forma Post - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search

Address http://localhost:8080/servlet/PostEx

Salutare, Ionel!

Adresa ta de e-mail... ionel@yahoo.com

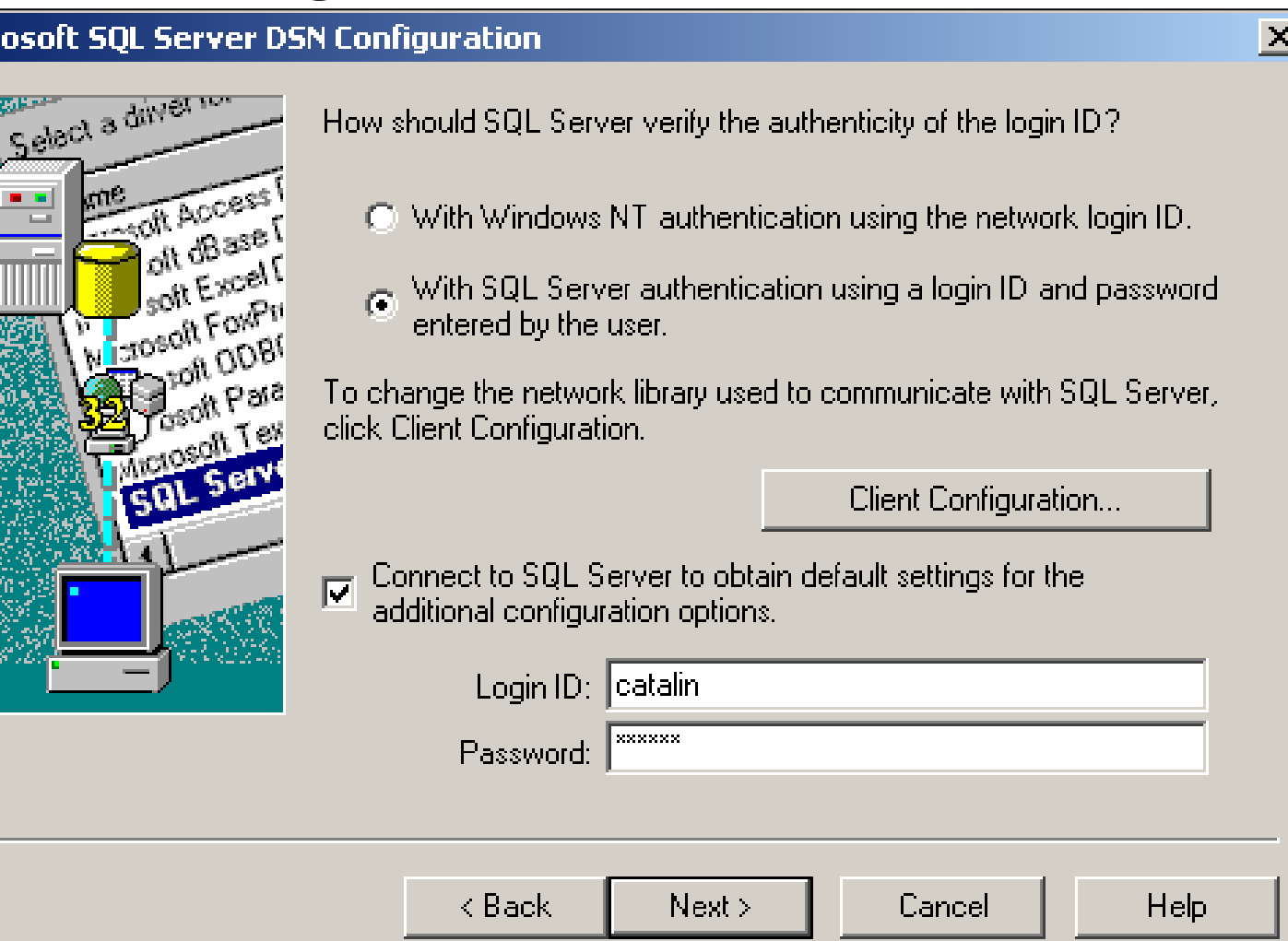
Continut

- Ce este un servlet
- De ce avem nevoie de servlet
- Structura unui servlet
- Instalarea
- Rularea unui servlet
- Metodele doGet si doPost
- **Accesarea unei BD SQL Server dintr-un servlet**

Cream un DSN

- Start – Control Panel – Administrative Tools – Data Sources (ODBC) – Add.. – SQL Server – Finish.
- La *name*, specificam numele DSN-ului (asa cum il specificam din programul Java)
- La *server* trebuie completat cu numele calculatorului unde este instalat serverul.
- Continuum apasand butonul *next*.

Inregistrarea DSN



Microsoft SQL Server DSN Configuration

How should SQL Server verify the authenticity of the login ID?

- With Windows NT authentication using the network login ID.
- With SQL Server authentication using a login ID and password entered by the user.

To change the network library used to communicate with SQL Server, click Client Configuration.

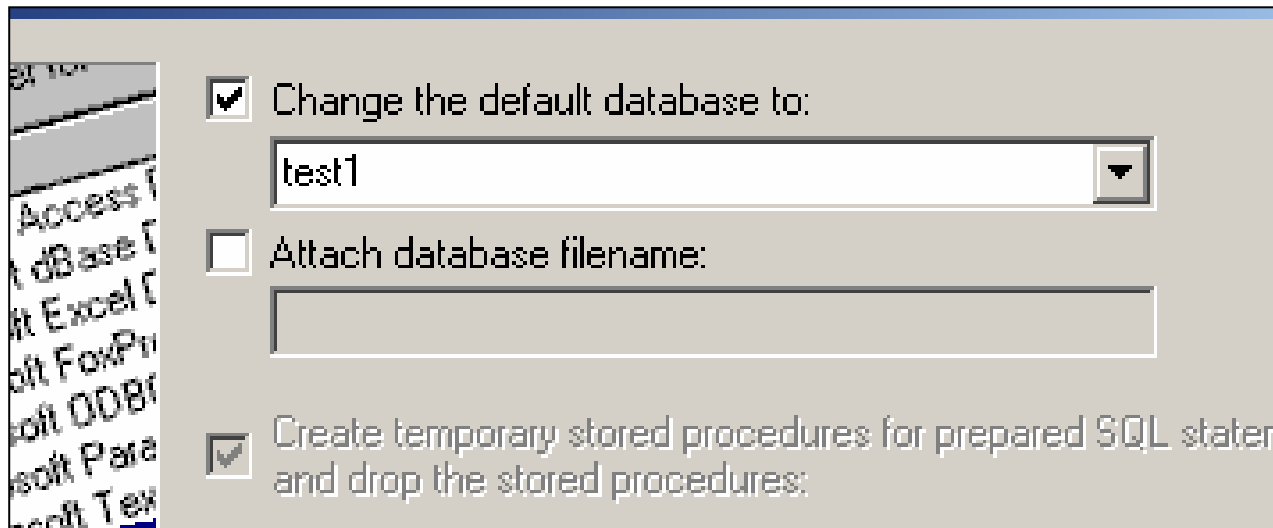
Connect to SQL Server to obtain default settings for the additional configuration options.

Login ID:

Password:

Specificam in noua fereastră cum ne vom conecta la baza de date SQL.

Specificarea bazei de date



In următoarea fereastră specificam care este baza de date care ne interesează și apăsăm *next*, apoi *finish*.

Structura tabelii pe care o vom accesa

- In tabela avem urmatoarele campuri:
 - id_user
 - nume
 - parola
 - data
- Vom citi din un browser un *ID* si vom afisa toate datele persoanei cu *ID*-ul respectiv.

Pagina HTML

```
<html>
<head>
<title>Accesam baza de date!</title>
</head>
<body>
<FORM ACTION="http://localhost:8080/servlet/ServletBD"
METHOD="GET">
I.D.<input type="text" name="id"><br>
<input type="submit" value = "Afiseaza tot!">
</form>
</body>
</html>
```

Metoda *init* a servletului

```
public void init(){  
try{  
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  
con =  
    DriverManager.getConnection("jdbc:odbc:student",  
    "catalin", "parola");  
stmt = con.createStatement();  
}catch(Exception e){System.out.println(e);}  
}
```



Metoda doGet


```
public void doGet(HttpServletRequest cerere,  
    HttpServletResponse raspuns) throws ServletException,  
    IOException {  
//...  
try{  
tab = stmt.executeQuery("Select * from tblUser where id_user =  
    " + cerere.getParameter("id"));  
out.println("Nume: " + tab.getString(2) + "<BR>");  
out.println("Parola: " + tab.getString(3) + "<BR>");  
out.println("Data: " + tab.getDate(4));  
}catch(Exception e){System.out.println(e);}  
out.println("</body>\n</html>");  
out.close();  
}
```



Metoda *destroy()*

```
public void destroy()
{
    try{
        stmt.close();
        con.close();
    }catch(SQLException e)
        {System.out.println(e);}
}
```

Rezultate

Address  C:\Documents and Settings\R

 **My Search** ▾

  ▾

I.D.

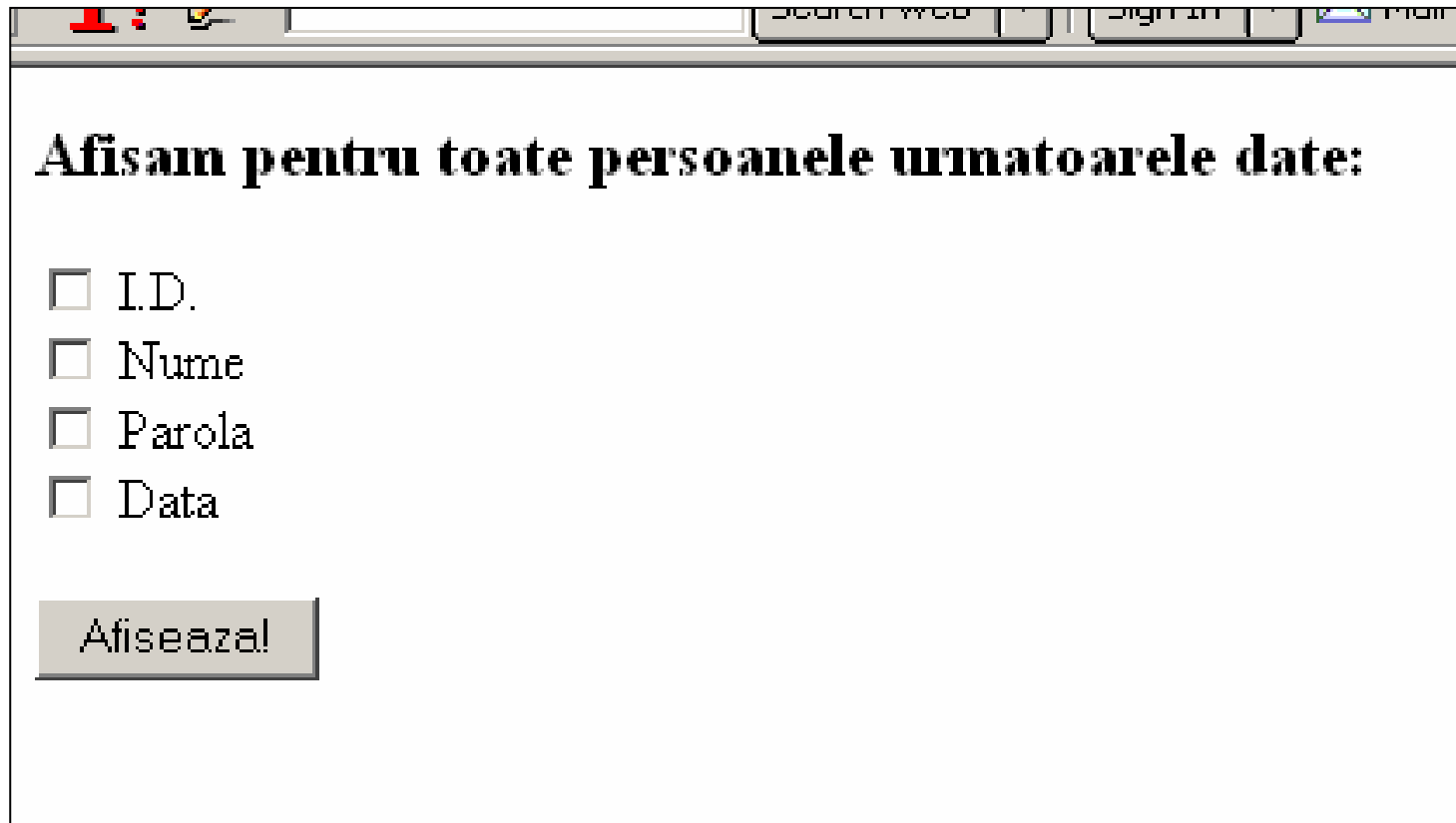
Vi afisam datele persoanei cu ID-ul = 3

Nume: Maricica

Parola: pitul

Data: 1978-07-22

With regard to excellence, it is not enough to know, but we must try to have and use it.(Aristotel)




A screenshot of a web browser window. The browser's address bar contains the text "Search web" and "Sign in". The main content area of the browser displays the following text:

Afiam pentru toate persoanele urmatoarele date:

- I.D.
- Nume
- Parola
- Data

Below the list of checkboxes is a button labeled "Afiseaza!".



Q&A