

תוכנה 1 – Servlets

תרגול 12

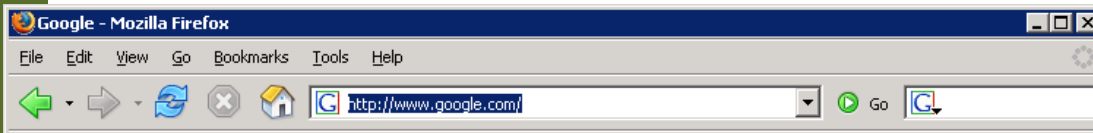
ליאור שפירא ומתי שמרת

אינטרנט על קצה המזלג

- האינטרנט היא רשת מחשבים (= מחשבים + חוטים)
- לכל מחשב ניתנת כתובת (IP Address) המורכבת מ-4 מספרים בין 0 ל-255
- כל תוכנית הרצה במחשב מקבלת מספר שלוחה (port) בין 0 ל-65,535
- ניתן למפות בין שמות לוגיים לכתובת IP (DNS) www.cs.tau.ac.il ממופה ל 132.67.252.100

שרת HTTP

- שרת HTTP מחכה לבקשות על Port 80
- בקשה לקבלת קובץ מהמחשב המרוחק
- בהינתן בקשה, השרת מחזיר תשובה... בעצם קובץ

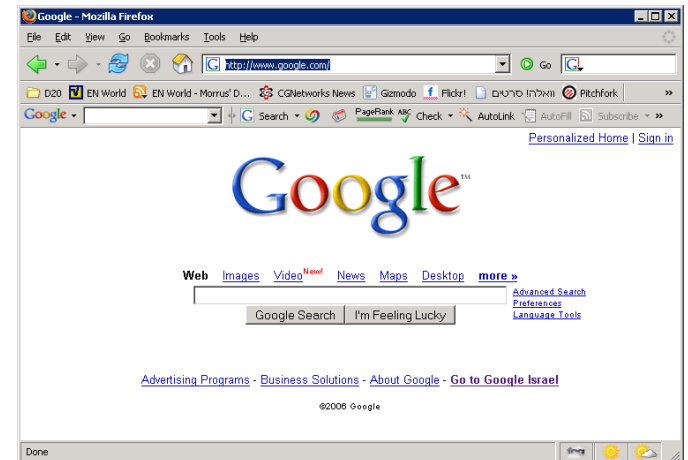


משתמש מקליד כתובת אינטרנט

השרת של GOOGLE מטפל בבקשה



תוכנה 1 בשפת Java
אוניברסיטת תל אביב



מחזיר דף HTML המוצג בדפדפן

Uniform Resource Locator (URL)

■ מייצג משאב כלשהוא ברשת, דף אינטרנט, קובץ, שירות כלשהוא...



URL (Uniform Resource Locator)

דפדפן

- דפדפנים מחלקים את הכתובת שהוכנסה בשורת הכתובת (URL) ל-5 חלקים, ובונים את ההודעה המתאימה. לדוגמא:

`http://www.cs.tau.ac.il:80/cs/index.html`



ברירת מחדל

ברירת מחדל

החלק החמישי עשוי להכיל משתנים כגון:

<http://www.google.co.il/search?hl=iw&q=cow&btn&meta=>

הבעיה

- **בעיה:** שרת האינטרנט אינו דינאמי – הוא יכול להציג רק דפים שהוכנו מראש
- תיבת הדואר שלי ב Gmail למשל נוצרת דינאמית
- היינו רוצים שרת שיענה על הדרישות הבאות:
 - שיידע לטפל בהיבטי התקשורת (ואולי גם היבטים אחרים)
 - שיאפשר לנו לכתוב לוגיקה נוספת (ולא רק להחזיר דפים שהוכנו מראש)

הרעיון

■ הפרדה בין:

■ **המסגרת (framework):** החוזרת על עצמה

■ "ההבטים": טיפול בלקוח חדש, יצירת חוט, יצירת שקע ועוד...

■ **הלוגיקה העסקית:** מה התוכנית עושה

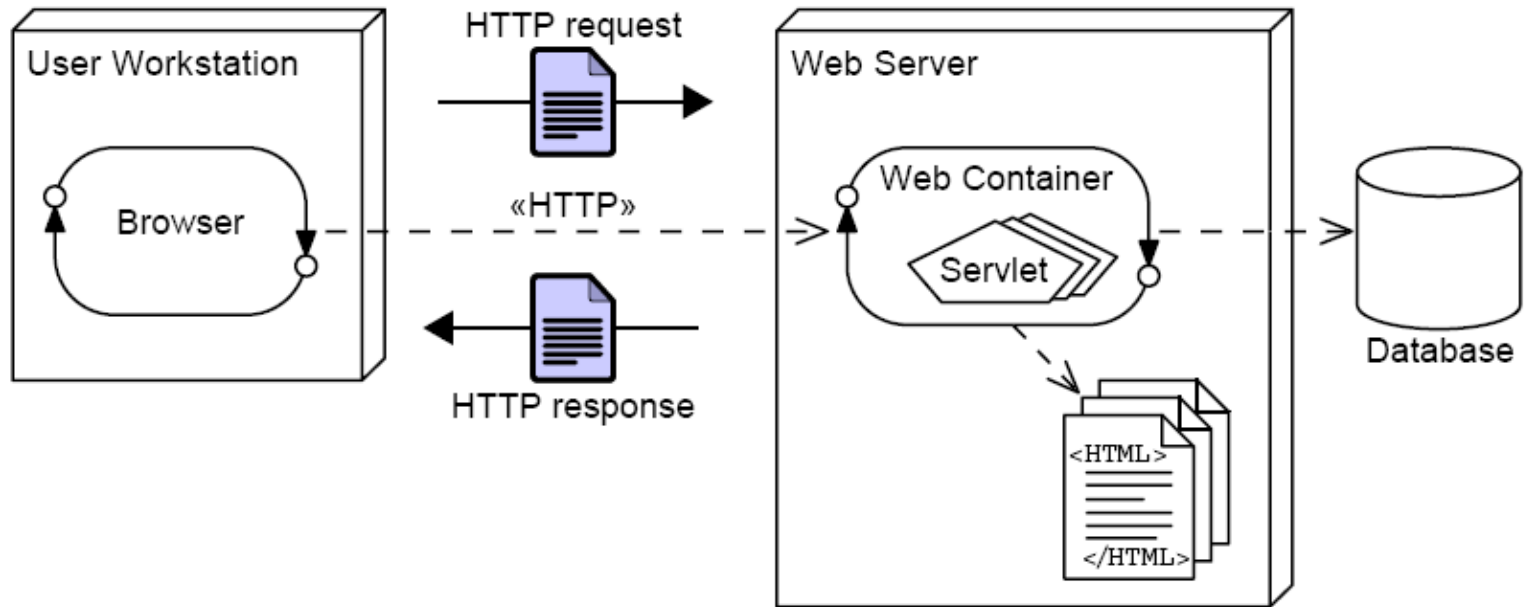
■ את השרת (web container) נכתוב פעם אחת ונטען לתוכו מחלקות Java בשם servlets (שרותונים?) אשר מממשות כל אחת לוגיקה עסקית משלה

■ בפרט נוכל להתקין על אותו שרת כמה שרותונים במקביל

■ נגדיר ממשק בין השרת ובין השירותונים

■ נוכל בהמשך להחליף את השרת מבלי להחליף את השרותונים

ארכיטקטורת Web Container



פעולת השרת: `HttpServer`

1. יצירת שקע והמתנה ללקוחות
2. עבור כל לקוח חדש (`accept`) בצע:
 1. קרא את השורה הראשונה בהודעה (`header`)
 2. אם מדובר בהודעת `GET`:
1. נתח את המחרוזת במקום המיועד לשם הקובץ וחלץ ממנה את שם ה `Servlet` המבוקש ואת שמות המשתנים וערכיהם (לפי התווים: `'/'`, `'='`, `'&'`, `'?'`)
2. קבל הפנייה לשרותון המתאים (`getInstance`)
3. צור את מחלקות העזר (`HttpServletRequest`, `HttpServletResponse`)
4. אם השרותון נמצא:
 - קרא ל `doGet` על השרותון עם מחלקות העזר המתאימות אחרת
 - אחרת: הצג הודעת שגיאה

הממשק בין השרת ל Servlet

Servlet אבסטרקטי מוגדר ע"י המחלקה HttpServlet

Method Summary

void	<u>doGet</u> (<u>HttpServletRequest</u> req, <u>HttpServletResponse</u> resp) Called by the server (via the service method) to allow a servlet to handle a GET request.
void	<u>doPost</u> (<u>HttpServletRequest</u> req, <u>HttpServletResponse</u> resp) Called by the server (via the service method) to allow a servlet to handle a POST request.
void	<u>doPut</u> (<u>HttpServletRequest</u> req, <u>HttpServletResponse</u> resp) Called by the server (via the service method) to allow a servlet to handle a PUT request.
void	<u>service</u> (<u>ServletRequest</u> req, <u>ServletResponse</u> res) Dispatches client requests to the protected service method.

HttpServletRequest

מייצגת בקשה שנשלחה לשרת ■

Method Summary

String	<u>getQueryString()</u> Returns the query string that is contained in the request URL after the path.
Enumeration	<u>getParameterNames()</u> Returns an Enumeration of String objects containing the names of the parameters contained in this request.
String	<u>getParameter()</u> (java.lang.String name) Returns the value of a request parameter as a String, or null if the parameter does not exist.

HttpServletResponse

תשובת השרת לבקשה ■

Method Summary

void	<u>setContentTypes</u> (java.lang.String type) Sets the content type of the response being sent to the client, if the response has not been committed yet.
PrintWriter	<u>getWriter</u> () Returns a PrintWriter object that can send character text to the client.

contentType – הפורמט של התשובה ■

text/plain – עבור טקסט רגיל ■

text/html – תשובה בפורמט HTML ■

SimpleServlet

- נתחיל בדוגמא פשוטה: `SimpleServlet` אשר מחזיר כתשובה את רשימת הפרמטרים שקיבל
- שרותון הוא Component – הוא מקבל במתנה מהמיכל (מה-`web container`) את כל הדברים שהוא זקוק להם (`aspects`)
- עליו לממש רק לוגיקה עסקית בתוך השרות `doGet` (`hook`)

SimpleServlet

```
public class SimpleServlet extends HttpServlet {  
  
    public void doGet(HttpServletRequest req,  
                      HttpServletResponse resp)  
        throws IOException, ServletException {  
  
        PrintWriter out = resp.getWriter();  
  
        resp.setContentType("text/plain");  
  
        Enumeration e = req.getParameterNames();  
        while (e.hasMoreElements()) {  
            String pname = (String)e.nextElement();  
            out.println(pname + " = " + req.getParameter(pname));  
        }  
    }  
}
```

Output is plain text

Use Java 1.4 syntax on university servers

SimpleHtmlServlet

```
public class SimpleHtmlServlet extends HttpServlet {  
  
    public void doGet(HttpServletRequest req,  
                      HttpServletResponse resp)  
        throws IOException, ServletException {  
  
        PrintWriter out = resp.getWriter();  
  
        resp.setContentType("text/html");  
        out.println("<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">");  
        out.println("<HTML>");  
        out.println("<HEAD><TITLE>SimpleHtmlServlet</TITLE></HEAD>");  
  
        out.println("<BODY>");  
        Enumeration e = req.getParameterNames();  
        while (e.hasMoreElements()) {  
            String pname = (String)e.nextElement();  
            out.println(pname + " = " + req.getParameter(pname) + "<BR>");  
        }  
        out.println("</BODY>");  
        out.println("</HTML>");  
    }  
}
```



Output is HTML

ChatServlet

- השרותונים הם Singletons
- אם שני לקוחות (דפדפנים) ניגשים לאותו ה Servlet הפניות מופנות לאותו עצם בזכרון
- ננצל עובדה זו כדי לכתוב ChatServlet
 - השרותון מציג טופס html (form)
 - ערכי השדות שהמשתמש מקליד משורשרים כפרמטרים לשורת הכתובת והעמוד נטען בשנית
 - השרותון מציג בנוסף לטופס את היסטוריית השיחה (השמורה בפרמטרים)

Sample Servlets

- <http://ibmop-01.cs.tau.ac.il:40212/examples/servlet/SimpleServlet?key=value>
- <http://ibmop-01.cs.tau.ac.il:40212/examples/servlet/SimpleHtmlServlet?key=value>
- <http://ibmop-01.cs.tau.ac.il:40212/examples/servlet/ChatServlet>