

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---



## העברת פרמטרים מהלקוח לשרת

פרק 3 – יצירת דף דינאמי ותכנות "חסר מצב"

### העברת נתונים מהלקוח לשרת

- טופס – form
- פעולה – action
- אופן השליחה – method

```
<form id="formName" action = "fileName.jsp  
method = "get">  
first name:  
<input type="text" name="userName" size = "20" /> <br />  
<input type="submit" value="try" name="send" /> <br />  
</form>
```

### שליחת נתונים באמצעות get

- המידע משורשר לשרת ה-url:

[www.myWebSite.com/myForm.jsp?userName=Haim&send=try](http://www.myWebSite.com/myForm.jsp?userName=Haim&send=try)

www.myWebSite.com/myForm.jsp	כתובת url
?	תחילת השרשור ב- ?
userName=Haim	ערך = שם שדה
&	שרשור לשדה הבא
send=try	

---

---

---

---

---

---

---

---

---

---

### שליחת נתונים באמצעות post

```
<form id="formName" action = "fileName.jsp
method = "post">
  first name:
  <input type="text" name="userName" size = "20" /> <br />
  <input type="submit" value="try" name="send" /> <br />
</form>
```

הנתונים משורשים במחרוזת הבנויה כסדרה של זוגות נתונים המורשרים בסימן &.

field1 = value1 & field2 = value2 & field3 = value3

המחרוזת מוכנסת לגוף ההודעה



---

---

---

---

---

---

---

---

---

---

### ההבדלים שבין שתי השיטות

get  
המידע גלוי, נשלח בשורת ה-url  
מוגבל בגודלו ל- 1000 תווים (אורך מרבי של שורת url)  
שיטת ברירת המחדל

post  
המידע מוסתר, נשלח בתוך גוף ההודעה, אינו מוגבל בגודלו



---

---

---

---

---

---

---

---

---

---

### גישה לנתונים שנשלחו על ידי לקוח

- request עצם המייצג את הנתונים שנשלחו מהלקוח (בשתי השיטות).
- המידע שמור במבנה: ערך.שם
- getParameter פעולה המאפשרת לאחזר את הנתונים

getParameter ("שם השדה") ← ערך הנתון

String name = request.getParameter ("userName");



---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

### בדיקת תקינות הקלט

- כדי לוודא שהלקוח אמנם שלח שם המכיל תווים, נכתוב פונקציה בוליאנית המחזירה "אמת" אם השם אינו null או מחרוזת ריקה, ו-"שקר" אחרת:

```
<%!  
boolean isValidValue (String name)  
{  
    return value != null && value != "" ;  
}  
%>
```



### JSP DIRECTIVE - הנחיית JSP

- רישום משפטי import הישמים לכל הדף:

```
<%@ page import = "java.util.Random" %>  
  
<%! Random rand = null; %>
```



### מבנה דף JSP

- הפרדה בין החלק הלוגי – קוד java וחלק התצוגה – קוד html
- חלק התצוגה יכול שימוש בעצם request, בפונקציות המוגדרות בחלק הלוגי ובמשפטי פלט: <% "טקסט" + משתנה + "טקסט" = %> או: <% out.print ("טקסט" + משתנה + "טקסט") %>
- החלק הלוגי יכול את הגדרת המשתנים הגלובליים והפונקציות: <% הגדרת משתנה / פונקציה ! %>
- ככלל – תמיד נעדיף שימוש במשתנים מקומיים והעברת פרמטרים בין הפונקציות.

