

Sessions (Συνεδρίες)

(Κεφ 9 απο Servlets και JSP)

- Το πρωτόκολλο HTTP είναι ένα πρωτόκολλο χωρίς διατήρηση κατάστασης (stateless). Κάθε αίτηση ενός χρήστη θεωρείται ανεξάρτητη απο την προηγούμενη.
- Σε εφαρμογές διαδικτύου χρειάζεται μερικές φορές να διατηρούνται πληροφορίες από προηγούμενες κλησεις ιστοσελίδων ενός χρήστη (π.χ. Πιστοποιημένη σύνδεση – login, καλάθι αγορών κλπ)
- Το περιβάλλον των Servlets παρέχει τις απαραίτητες δομές για την διατήρηση αυτών των πληροφοριών.

HttpSession

- HttpSession είναι μια κλάση που χρησιμοποιείται για να διατηρεί πληροφορίες στον διακομιστή κατά την διάρκεια προσπέλασης μιας εφαρμογής από ένα χρήστη.
- Ένα αντικείμενο HttpSession δημιουργείται καλώντας την μεθοδο getSession() ενός αντικειμένου **HttpRequest**
- HttpSession session = request.getSession()

getSession(Boolean)

- Η μέθοδος getSession() χωρίς κανένα όρισμα επιστρέφει το αντικείμενο HttpSession να έχει δημιουργηθεί διαφορετικά θα δημιουργήσει ένα νέο αντικείμενο και θα το επιστρέψει.
- Υπάρχει και η μέθοδος getSession(boolean create) με μια boolean παράμετρο. Αν η τιμή που δίνεται είναι false τότε αν ένα αντικείμενο HttpSession δεν υπάρχει δεν θα δημιουργηθεί νέο και η μέθοδος θα επιστρέψει την τιμή null. Αν η τιμή είναι true τότε αν δεν υπάρχει αντικείμενο HttpSession θα δημιουργηθεί.

Χρήση ενός HttpSession

- Ένα αντικείμενο session μπορεί να θεωρηθεί σαν ένας πίνακας (hashtable) στον οποίο μπορούν να τοποθετηθούν διαφορα άλλα αντικείμενα με την χρήση κλειδιών (keys).

HttpSession Μεθοδοι

- `public Object getAttribute (String name)`
- `public Enumeration<String> getAttributeNames();`
- `public void setAttribute (String name, Object value);`
- `public void removeAttribute (String name)`
- `public void invalidate()`
- `public String getId()`
- `public long getCreationTime()`
- `public long getLastAccessTime()`
- `public int getMaxInactiveInterval ()`
- `public void setMaxInactiveInterval (int seconds)`

Συνδεση των HttpSession με τον “Πελατη” (Client)

- Καθε συνεδρία (session) εχει ενα μοναδικό αριθμό (JSESSIONID).
- Ο αριθμός αυτός στέλνεται στον φυλομετρητη του πελάτη και στέλνεται παλι από τον πελατη στο διακομιστή με την επόμενη αίτηση/κλήση.
- Υπάρχουν δύο μέθοδοι που μπορεί να χρησιμοποιηθούν για την αποστολή του jsessionid :
 - Χρήση Cookies (Μπισκότα)
 - Αντιγραφή URL (URL rewriting)

Cookies

- Τα “Μπισκότα” (Cookies) είναι ζεύγη ονόματων-τιμών που στέλνονται από ένα Web Server σε ένα Web Browser. Οι πληροφορίες αυτές μπορούν να σταλούν από τον Web Browser πίσω στον server όταν ο χρήστης επισκευθεί ξανά την σελίδα με την οποία συνδέονται τα cookies.
- Υπάρχουν μόνιμα (persistent) και προσωρινά (session) cookies.
- Το περιβάλλον των Servlets χρησιμοποιεί προσωρινά cookies για να αποθηκεύεται το ID ενός HttpSession.

Αντιγραφή URL

- Με την μέθοδο αυτή κάθε URL που περιέχεται σε μια σελίδα που παράγεται από την εφαρμογή και δείχνει προς την ίδια την εφαρμογή δημιουργείται ξανά τοποθετώντας το Session Id στην λιστα των παραμέτρων (query string) του URL. Έτσι το SessionID μπορεί να σταλεί πίσω στον server.
- Το Servlet API παρέχει μεθόδους για την διευκόλυνση της επανεγγραφής URL όπως η encodeURL που ενσωματώνει το JSESSIONID

```
String encodeURL = response.encodeURL(originalURL)
```

```
out.println (“<a href=\\” + encodeURL + “\\”>...</a>”)
```


ServletConfig

- Η κλάση `ServletConfig` αντιπροσωπεύει ένα αντικείμενο που δημιουργείται από έναν περιέκτη `Servlet` (πχ `apache tomcat`) και παραδίδεται σέ ένα `servlet` όταν αρχικοποιείται.
- Το αντικείμενο αυτό περιέχει κυρίως παραμέτρους αρχικοποίησης (`initialization parameters`) που ορίζονται στο αρχείο περιγραφέα εκδίπλωσης (`deployment descriptor`) `web.xml` της εφαρμογής.
- Το αντικείμενο αυτό επιστρέφεται με την μέθοδο `getServletConfig()` της κλάσης `HttpServlet`

Μέθοδοι της ServletConfig

- **String getInitParameter(String)** Επιστρέφει την τιμή μιας παραμέτρου αρχικοποίησης το όνομα της οποίας δίνεται σαν παράμετρος της μεθόδου.
- **Enumeration getInitParameterNames()** Επιστρέφει όλα τα ονόματα παραμέτρων αρχικοποίησης
- **String getServletName()** Επιστρέφει το όνομα που έχει δοθεί στο Servlet
- **ServletContext getServletContext()** Επιστρέφει ένα αντικείμενο ServletContext με πληροφορίες σχετικές με την εφαρμογή στην οποία ανήκει το Servlet

Παράδειγμα Ορισμού Παραμέτρων σε αρχείο (web.xml)

```
<servlet>
```

```
  <servlet-name>HelloWorld</servlet-name>
```

```
  <servlet-class>TestServlet</servlet-class>
```

```
  <init-param>
```

```
    <param-name>myprop</param-name>
```

```
    <param-value>value</param-value>
```

```
  </init-param>
```

```
</servlet>
```

ServletContext

- Η κλάση **ServletContext** παρέχει πληροφορίες για το περιβάλλον (context) μιας εφαρμογής. Περιέχει κυρίως παραμέτρους αρχικοποίησης για όλη την εφαρμογή (και όχι μόνο για ένα Servlet όπως η ServletConfig)
- Αποτελεί και μια δομή που μπορεί να χρησιμοποιηθεί για να ανταλλάσσουν πληροφορίες διάφορα servlets μιας εφαρμογής.
- Η μέθοδος **getServletContext()** που παρέχεται από την διασύνδεση HttpServlet επιστρέφει ένα αντικείμενο ServletContext.

Μέθοδοι της ServletContext

- **String getInitParameter(String)** Επιστρέφει την τιμή της παραμέτρου αρχικοποίησης με το όνομα που δίνεται σαν παράμετρος
- **Enumeration getInitParameterNames()** Επιστρέφει όλα τα ονόματα των παραμέτρων.
- **Object getAttribute(String)** Επιστρέφει ένα αντικείμενο που έχει τοποθετηθεί στο ServletContext με κάποιο όνομα.
- **setAttribute(String, Object)** Θέτει κάποιο αντικείμενο στο ServletContext με κάποιο όνομα
- **Enumeration getAttributeNames()** Επιστρέφει όλα τα ονόματα των αντικειμένων που έχουν τοποθετηθεί στο ServletContext.

Παράδειγμα ορισμού παραμέτρων εφαρμογής στο web.xml

```
<web-app>
```

```
  <context-param>
```

```
    <param-name>Country</param-name>
```

```
    <param-value>India</param-value>
```

```
  </context-param>
```

```
  <context-param>
```

```
    <param-name>Age</param-name>
```

```
    <param-value>24</param-value>
```

```
  </context-param>
```

```
</web-app>
```

Κρυφά Πεδία

- Μια άλλη μέθοδος που μπορεί να χρησιμοποιηθεί από εφαρμογές για διατήρηση κατάστασης μεταξύ διαδοχικών αιτήσεων / αποκρίσεων HTTP και χρήση φορμών HTML είναι τα κρύφα πεδία (hidden fields). Για παράδειγμα:
- `<input type="hidden" name="mysecret" value="notvisible">`
- Το προηγούμενο πεδίο μπορεί να τοποθετηθεί σε μια δυναμικά δημιουργούμενη φόρμα και όταν η φόρμα υποβληθεί θα σταλεί το ζεύγος `mysecret=notvisible` σαν παράμετρος στον server.