
Coding Bootcamp

Session in JSP & Servlets

Session

- Η συνεδρία (Session) είναι ο όρος που χρησιμοποιείται για να δηλώσει τον «χρόνο» περιήγησης σε ένα site. Η συνεδρία ξεκινά όταν ο χρήστης ανοίγει τον browser και επισκέπτεται (στέλνει το 1^ο request) το site και τερματίζεται (φυσιολογικά) όταν ο χρήστης κλείσει το παράθυρο του browser. Η συνεδρία μπορεί να τερματιστεί και από την εφαρμογή (site).

Παράδειγμα: `<% session.invalidate(); %>`

- Σε περίπτωση που ο χρήστης δεν αλληλοεπιδράσει με το site (σελίδες του) για κάποιο χρονικό διάστημα τότε ο εξυπηρετητής τερματίζει την συνεδρία. Ο χρόνος αυτός μπορεί να εξαρτάται από τις ρυθμίσεις του web server που σερβίρει το site (πχ στον Tomcat από default είναι 30'), είτε από την ίδια την εφαρμογή (site).

```
<session-config>
  <session-timeout>30</session-timeout>
</session-config>
```

← 30 minutes (Tomcat web.xml or application web.xml)

```
<%
session.setMaxInactiveInterval(1800);
%>
```

← 1800 sec = 30 minutes (ex. in a JSP page)

Http is a stateless protocol

- Δεν απαιτεί από τον εξυπηρετητή (server) να διατηρεί τις πληροφορίες της συνεδρίας ή την κατάσταση σχετικά με κάθε χρήστη κατά την διάρκεια πολλαπλών αιτημάτων (requests).
- Αντιμετωπίζει κάθε αίτημα (request) ως ανεξάρτητη «συναλλαγή» που δεν έχει καμία σχέση με οποιαδήποτε προηγούμενη αίτηση, έτσι ώστε η επικοινωνία αποτελείται από ανεξάρτητα ζεύγη αιτήματος και απόκρισης (client request & server response)

The problem and Session Tracking Techniques

- Η εφαρμογή και ο χρήστης έχουν «επίγνωση των πληροφοριών» που ανταλλάσσουν μόνο κατά την διάρκεια του τρέχοντος request, στην συνέχεια «ξεχνάνε» ο ένας τον άλλο. Κατά συνέπεια υπάρχει πρόβλημα όταν υπάρχει η ανάγκη να διατηρηθούν-χρησιμοποιηθούν πληροφορίες που ανταλλάξαν μεταξύ τους σε προηγούμενα requests κατά την διάρκεια περιήγησης του πελάτη στο site.

Λύσεις:

- [Cookies](#)
- Hidden fields
- URL Rewriting
- [Session object](#)

(Session based) Client - Server Communication

- Ο πελάτης ανοίγει τον browser του και πληκτρολογεί το URL της εφαρμογής (site)
- Ο εξυπηρετητής (πχ Tomcat) ανοίγει μια συνεδρία με τον πελάτη και παράγει έναν τυχαίο μοναδικό αλφαριθμητικό (Session ID), δημιουργεί ένα Cookie αποθηκεύοντας το Session id και το προωθεί στον πελάτη μέσω του Http header
- Ο browser του πελάτη αποθηκεύει το Cookie
- Ο πελάτης κάθε φορά που αλλάζει σελίδα (δηλ. στέλνει request) ο browser του στέλνει μεταξύ άλλων και το Cookie με το session id
- Ο εξυπηρετητής διαβάζει το cookie και χρησιμοποιεί το session id για να ξεχωρίσει τον κάθε πελάτη που «μιλάει»
- Ο εξυπηρετητής μπορεί να διαθέσει χώρο στο δίσκο του για να αποθηκεύσει πληροφορίες (μέσω της εφαρμογής – **Session Object**) για τον πελάτη και γενικά για την συνεδρία τις οποίες «ταξινομεί» με βάση το session id

HttpSession in Servlets & JSP (1)

- Στα Servlets μπορείτε να πάρετε και να χρησιμοποιήσετε το Session μέσω του request object ([HttpServletRequest](#))

Παράδειγμα: **HttpSession session = request.getSession(true);**

- Στην JSP το Session είναι **implicit object** και μπορείτε να το χρησιμοποιείται αυτόματα (βλέπε μάθημα JSP) εκτός και εάν δηλώσετε ότι δεν επιθυμείτε να συμπεριληφθεί στην σελίδα σας (<%@ page session="false" %>)

HttpSession in Servlets & JSP (2)

- Παρακάτω γίνεται μια αναφορά σε μερικές από τις μεθόδους του interface [HttpSession](#)

Τύπος Επιστροφής	Μέθοδος	Περιγραφή
String	getId()	Returns a string containing the unique identifier assigned to this session
void	setAttribute(String name, Object value)	Binds an object to this session, using the name specified.
Object	getAttribute(String name)	Returns the object bound with the specified name in this session, or null if no object is bound under the name.
void	removeAttribute(String name)	Removes the object bound with the specified name from this session
void	invalidate()	Invalidates this session then unbinds any objects bound to it
boolean	isNew()	Returns true if the client does not yet know about the session or if the client chooses not to join the session. For example, if the server used only cookie-based sessions, and the client had disabled the use of cookies, then a session would be new on each request