

# Esame

## ▼ 14Luglio2017

### ▼ Es1

Pagina iniziale in cui ci si registra o si fa login. Dopo la registrazione ogni utente farà parte di un gruppo: si supponga già nota server-side la lista degli utenti registrati e i loro dati. Una volta registrato l'utente potrà aggiungere il numero di biglietti desiderato al suo carrello personale e tramite un pulsante chiedere di completare l'acquisto: chiedere perchè ragioniamo a gruppi, nel senso che l'acquisto verrà finalizzato soltanto dopo che tutti gli utenti (di quel gruppo) hanno pigiato il pulsante per l'acquisto. Nel caso anche un singolo utente non lo pigia (entro il termine della sessione) l'intero acquisto di gruppo deve essere annullato e tutti gli utenti devono riniziare da capo le loro operazioni.

C'è inoltre un admin che deve poter avere la possibilità in ogni momento di forzare l'acquisto o l'annullamento di ogni gruppo.

---

Nella pagina iniziale index.jsp creiamo un FORM con method = "post" uno per i login e uno per la registrazione.

La servlet Login.java gestisce appunto il login e contiene al suo interno il metodo init() nel quale crea gli utenti e l'admin. Viene invocata una

```
this.getServletContext().setAttribute("utentiRegistrati", utentiRegistrati),
```

dove utentiRegistrati è una mappa in cui sono salvati gli utenti. Vengono quindi creati i gruppi, inseriti i dati in una mappa e di nuovo

```
this.getServletContext().setAttribute("gruppi", gruppi).
```

La doPost() gestisce il Login(). Per prima cosa ottiene l'utente che fa una request e la sessione ( `HttpSession session = request.getSession()` ). Gestisce

quindi il tipo di utente:

- se è null, cioè NON è nel database, carica la `index.jsp` con una `RequestDispatcher`:  

```
RequestDispatcher rd =  
getServletContext().getRequestDispatcher("index.jsp"); rd.forward(request,  
response); return;
```
- viene fatta quindi una `session.setAttribute("currentUser", utente)`
- se l'utente è un admin, viene indirizzata la richiesta alla jsp che gestisce gli admin con una `RequestDispatcher`
- se la password è errata si ricarica la `index.jsp`
- nel caso di successo imposta la sessione dell'utente con la sessione appena creata; recupera la mappa dei gruppi e il gruppo dell'utente. aggiunge l'utente al gruppo e imposta l'attributo "gruppo" nella sessione con il gruppo utente. reindirizza l'utente al `catalogo.jsp`

`catalogo.jsp` è una jsp che appunto permette di inserire nel carrello un biglietto ed eventualmente acquistarlo.

▼ Es2

▼ Es3

▼

## 02lug2018

1. conteggio concorrente tra bean e servlet delle maiuscole in 3 file di testo
2. calcolo di un integrale?

## 14lug2017

1. acquisto biglietti di gruppo con autenticazione e admin che può forzare le operazioni
2. trasformazione dei caratteri di un file in upperCase, il tutto eseguito da una servlet che lavora in modo concorrente (s1 sulla prima metà del file, s2 sull'altra) e ritorna

anche il numero di caratteri trasformati

## **15gen2020**

1. gestione scadenza password di utenti e gruppi: gli utenti devono fare login /registrarsi e devono cambiare password dopo tot giorni. inoltre se  $n > 3$  password sono scadute all'interno di un gruppo, tutti gli utenti dello stesso devono aggiornare la password. Gli utenti hanno anche un numero max di tentativi
2. sottrazione tra matrici con admin che può invalidare tutte le sessioni se il server è in sovraccarico

## **17set2019**

1. gestione prenotazione di n alberghi con incremento di prezzo in base al numero di persone interessate allo stesso hotel e periodo.

## **07set2022**

1. scrittura di una stringa che viene inviata in automatico alla pressione di % al server che restituisce (se esistente) il testo associato. Presenti due pulsanti per abilitare (max 1 volta) / disabilitare i permessi di scrittura. Admin che può revocare quando vuole i permessi di scrittura degli utenti

## **09giu2022**

1. inserimento di una stringa alfanumerica, eliminazione occorrenze tramite S1 e conteggio caratteri maiuscoli tramite J2. Trasferimento dati in formato Json. inoltre eventuale ridirezione a pagina di attesa nel caso di sovraccarico
2. calcolo parallelo degli anagrammi di una parola, inviata direttamente all'inserimento del carattere \$ (ajax).
3. REACT. Schedina. inseriamo 5 interi validi, estraiamo casualmente 5 interi e vediamo se qualcuno fa match: in questo caso lo scriviamo e coloriamo le caselle di

testo

## TIPS

Se devo reindirizzare una richiesta e utilizzo `sendRedirect()`, i parametri vanno passati in questo modo: `sendRedirect("nome?par1="+a+"&par2="+b)`

In certi casi sono costretto ad utilizzare `sendRedirect()`: se da una jsp voglio inoltrare i dati a una servlet in formato Json ad esempio, non posso utilizzare `requestDispatcher()`.