Esercizi svolti - 8

Configurazione Eclipse

Selezioniamo il server tomcat 9.0 come server.

Una volta aggiunto il server possiamo iniziare a configurare il server cliccando due volte sul server stesso:

Ogni volta che dispiegheremo un'applicazione web, verrà creata una nuova direcotory nella directory **webapp**:

Creazione di progetto con eclipse

Creiamo un nuovo progetto dinamico:

A questo punto dobbiamo scrivere il codice del servlet che vogliamo dispiegare in tomcat per avere il comportamento richiesto dalla traccia 8.1; Creiamo quindi una servlet:

Possiamo inoltre selezionare diversi metodi di servizio che vogliamo implementare nella specifica servlet che stiamo realizzando:

Viene generata automaticamente la nostra classe:

Notiamo che viene generata anche **un'annotazione**; le annotazioni sono in generale delgi input forniti agli ambienti che utilizzano il codice che andiamo a scrivere. Attraverso queste annotazioni forniamo informazioni addizionali agli ambienti di programmazione.

Stiamo dicendo che si tratta di una webservlet tramite **extends HttpServlet**.

Per fare si che il server risponda con una stringa Hello World cosa dobbiamo fare?

Inviamo una risposta

Per inviare una risposta per prima cosa impostiamo il contenuto tramite MIME, dicendo `response.setContentType("text/plain");`. Dopodichè recuperiamo un writer con `PrintWriter pw = response.getWriter();`.

Successivamente non ci resta altro che inviare la nostra stringa con `pw.println("Hello servlet world! ");`

Come dispieghiamo il codice?

La prima cosa da fare è selezionare con **add/remove** il nostro Package. Dopo aver aggiunto il package al server tomcat dobbiamo effettuare un **publish**;

Per testare il funzionamento ci basta avviare il server tomcat ed effettuare una richiesta di tipo GET sull'URI: `http://127.0.0.1:8080/Esercizio8.1/Hello`

Ci basta cambiare leggermente la struttura della gestione del metodo GET per ottenere una visualizzazione non più text/plain ma text/html :

```
    response.setContentType("text/html"); // diciamo al browser che vogliamo inviare del testo semplice
    PrintWriter pw = response.getWriter();

    pw.println("<H1> Hello Servlet! </H1>");
```

Se effettuiamo la stessa richiesta fatta prima con Postman, con un browser ci viene renderizzato il messaggio di risposta:

Esercizio 8.2

Il server deve rispondere con dei parametri passati nella richiesta GET dove firstname=<nome> e lastname=<cognome>.

Per recuperare i due parametri abbiamo bisogno del metodo `request.getParameter()`; come parametro del metodo dobbiamo passare il nome del parametro che vogliamo recuperare, nel nostro caso "firstname" e "lastname".

Facciamo un semplice "documento" HTML dove prevediamo la stampa delle due variabili lette con:

```
String responseHTML = "<H1>"
        + "Ti chiami:"
        + "</H1>"
        + "" + firstname + " " + lastname;
```

Quindi ci basta visitare l'indirizzo `http://127.0.0.1:8080/Esercizio8.2/Echo?firstname=sopraIl18&lastname=siAccetta` per ottenere:

Esercizio 8.3

Dobbiamo scrivere una servlet chiamata `Calculate`: riceve due operandi ed un operatore (+,-,*,/), usando un form `calculator.html`, e che ritorna il risultato all'interno del browser.

Per realizzare una calcolatrice possiamo avvalerci di file HTML.

Ci basta aprire il file calculator.html per trovare il form, che invierà una richiesta GET: `http://127.0.0.1:8080/Esercizio8.3/Calculate?op1=2&op2=3&operation=%2b`; in questo caso stiamo facendo l'operazione 3+2.

Esercizio 8.4

Vogliamo definire una servlet che quando viene contattata dal client restituisce il numero di contatti.

Ci basta creare una variabile di istanza `contacts` che tiene conto delle volte che il server è stato contattato. Ogni volta che entriamo nel metodo **doGet** ci basta incrementare contacts e stampare sul writer.