

Web Application

Step 1 Creazione Dynamic Web Project

File -> New -> Dynamic Web Project

In Project name: nome del progetto, (ma vaaa)

Target Runtime: La versione di Tomcat che si sta andando ad utilizzare

Per definire un nuovo runtime a partire da un Tomcat installato fare New Runtime, selezionare la versione di Tomcat (e.g. v7.0) e poi fornire la directory di installazione di Tomcat.

Assegnare al progetto un working set: ad esempio si può creare il working set ISSSR

Spuntare la casella: Generate web.xml deployment descriptor

Step 2 File web.xml

Deve essere presente in WEB-INF/web.xml

In questo file sono presenti al momento della creazione il nome dei welcome file che verranno presentati quando si accede alla web application.

Per impostare una pagina iniziale (e.g. LoginPage.jsp) cambiare all'interno dei tag welcome-file index.jsp in LoginPage.jsp, in questo modo sarà la prima pagina visibile quando si accede alla web application.

Step 3 Definizione classi .jsp e classi bean

Le classi .jsp si possono definire in all'interno di WebContent.

Provare a definire una classe di esempio HelloWorld.jsp, se si fa dx -> new direttamente da WebContent esce già .jsp, così si risparmia tempo

Step 4 Deploy su Tomcat

Generazione WAR:

WAR sta per **W**eb Application **AR**chives ed è un archivio usato in Java per raggruppare Java Server Pages (JSP), Servlet, classi Java, xml,...

Per generare il WAR si fa dx sul progetto -> export -> WAR file e si indica la directory dove andare a salvare il WAR.

Un WAR contiene tutto il necessario per il deploy della nostra Web Application, ad esempio su un Application Server come Tomcat.

Deploy webapp: Data la cartella di installazione di Tomcat: ad esempio vs 7.0.47, si deve inserire il file .war prodotto all'interno della directory webapps, che contiene tutte le web application di cui viene fatto il deploy.

Per far partire il server Tomcat si deve andare nella cartella bin ed eseguire il file batch startup.

Step 5 Utilizzo web application

Una web application è per definizione fruibile da un browser, e.g Google chrome, Firefox

Per utilizzare la web application si deve quindi andare su localhost:8080/nomeProgetto/nomePagina.jsp

e.g.: <http://localhost:8080/ISPWebapp/LoginPage.jsp>

NB: Se si è impostata la pagina iniziale allora è sufficiente scrivere

<http://localhost:8080/ISPWebapp>

Step 6 Configurazione Database

Per semplicità si può andare ad utilizzare XAMPP:

<http://www.apachefriends.org/it/xampp.html>

XAMPP integra Apache http Server, MySQL e PHPMyAdmin per la gestione del database MySQL dal browser.

Innanzitutto si fa partire il server di MySQL: da XAMPP Control Panel fare start su MySQL, che automaticamente si attesta sulla porta 3306.

Per la gestione del database con PHPMyAdmin si deve lanciare anche il web server di Apache, facendo sempre start sul modulo Apache. Si dovrebbe attestare sulla porta 80.

A questo punto si può accedere a PHPMyAdmin con <http://localhost/phpmyadmin/>

Si accede con username e password: di default root, root.

Da PHPMyAdmin si può creare il database e le tabelle relative ed effettuare tutte le operazioni su un DB. Tutorial su http://www.youtube.com/watch?v=1-81n_vuwug

NB: XAMPP rileva automaticamente le azioni intraprese da Tomcat, quindi quando si avvia il server XAMPP rileva che il server è stato avviato e presenta messaggi tipo Status change detected: running, stopping,...

A questo punto quando la WebApplication, in particolare le classi DAO possono andare a interagire con MySQL ed effettuare delle query sul DB.

Quando si effettua la connessione al db deve essere fornito il seguente url:

`jdbc:mysql://localhost:3306/nomeDatabase`

Dove il nomeDatabase è lo stesso nome che è stato utilizzato nella sua creazione da PHPMyAdmin. Username e password è consigliabile lasciare quelle di default: root, root.

Esempio fornito dal progetto ISPW dell'a.a 2012/2013 in ISPWebapp.