Installing Apache Tomcat with NetBeans



Sommario

- Apache Tomcat
 - Componenti di Apache Tomcat
- Web Application
 - Componenti di un Web Application
- Configurazione ambiente con Netbeans 8.x
 - Server Apache
 - Database
- Esempio di Registrazione e Login
- Tomcat Web Application Manager

Prima di iniziare...

• Scaricare Java SDK

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

Scaricare Netbeans
 <u>https://netbeans.org/downloads/</u>

• Scaricare la versione "MySQL Server" (32-64 bit) <u>https://dev.mysql.com/downloads/mysql/</u>

Scaricare e installare Java SDK

• Scaricare Java SDK dal seguente link:

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

Java SE Development Kit 8u111

You must accept the Oracle Binary Code License Agreement for Java SE to download this software. Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may

now download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.78 MB	
Linux ARM 64 Hard Float ABI	74.73 MB	jdk-8u111-linux-arm64-vfp-hflt.tar.gz
Linux x86	160.35 MB	🚽jdk-8u111-linux-i586.rpm
Linux x86	175.04 MB	<u> </u> Jdk-8u111-linux-i586.tar.gz
Linux x64	158.35 MB	➡jdk-8u111-linux-x64.rpm
Linux x64	173.04 MB	<u> </u> Jdk-8u111-linux-x64.tar.gz
Mac OS X	227.39 MB	<u> </u> Jdk-8u111-macosx-x64.dmg
Solaris SPARC 64-bit	131.92 MB	jdk-8u111-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	93.02 MB	jdk-8u111-solaris-sparcv9.tar.gz
Solaris x64	140.38 MB	➡jdk-8u111-solaris-x64.tar.Z
Solaris x64	96.82 MB	<u> </u> ➡jdk-8u111-solaris-x64.tar.gz
Windows x86	189.22 MB	jdk-8u111-windows-i586.exe
Windows x64	194.64 MB	jdk-8u111-windows-x64.exe

Scaricare Netbeans

• Scaricare Netbeans:

https://netbeans.org/downloads/

Subscribe to newsletters: Mo Ne	onthly DWeekly tBeans can contac	/ t me at this addres	Language: L	Note: Greyed out tecl	hnologies are not sup
			NetBeans IDE	Download Bundle	:5
Supported technologies *	Java SE	Java EE	HTML5/JavaScript	PHP	C/C++
NetBeans Platform SDK	•	•			
🖲 Java SE	•	•			
🖲 Java FX	•	•			
🖲 Java EE		•			
🖲 Java ME					
HTML5/JavaScript		•	•	•	
PHP			•	•	
C/C++					•
Groovy					
Iava Card™ 3 Connected					
Bundled servers					
GlassFish Server Open Source Edition 4.1.1		•			
Apache Tomcat 8.0.27		•			
	Download	Download	Download ×86 Download x64	Download x86 Download x64	Download x86 Download x64

Installare Netbeans

(NetBeans IDE Installer

\sim
\sim

Welcome to the NetBeans IDE 8.2 Installer

The installer will install the NetBeans IDE with the Java EE pack and selected application servers.

Select the application servers to install with the IDE:

GlassFish Server Open Source Edition 4.1.1

NetBeans IDE Installer	_		\times
NetBeans IDE 8.2 Installation Choose the installation folder and JDK™.		😵 NetBer	anside
Install the NetBeans IDE to:			
C:\Program Files\NetBeans 8.2		Bro	wse
JDK™ for the NetBeans IDE:			
C:\Program Files\Java\jdk1.8.0_111		✓ Broy	wse

🛞 NetBeans IDE

Installation Size: 529.5 MB

Next > Cancel

< Back Next > Cancel

Model View Controller (MVC)

MVC e' un utile pattern architetturale per la separazione dei compiti di un software.

- **Model:** *Plain Old JavaScript Object* (POJO) che trasporta i dati.
- **View:** livello nel quale i dati sono rappresentanti in formato visuale.
- **Controller:** componente responsabile per la comunicazione tra *model* e *view*.



Cos'e' Apache Tomcat ?

- Apache Tomcat e' un Application Server basato su Java Servlet
- Tomcat implementa diverse specifiche:
 - Java Servlet
 - JavaServer Pages (JSP)
 - ect



Architettura MVC

Componenti di Tomcat

1. Catalina

- Contenitore di *Servlet Java* di Tomcat
- Fornisce l'effettiva implementazione di Tomcat, delle specifiche della Servlet.

2. Coyote

- Componente "connettore HTTP" di Tomcat
- Supporta il protocollo HTTP per il Web Server
- Ascolta le connessioni in entrata su una specifica porta TCP, inoltra la richiesta al motore Tomcat e processa la richiesta e manda indietro la risposta al client.

3. Jasper

- Analizza e converte le porzioni di codice all'interno delle pagine JSP in Servlet e le passa a Catalina per processarle
 - Quando lanciato, cerca eventuali cambiamenti avvenuti ai file JSP e, se necessario li ricompila.

Componenti di un Web Application (WA)

- Un Web component e' un oggetto che estende le funzionalita' di un Web Server aggiugendone di nuove:
 - Servlet
 - Java Server Pages (JSP)
 - ecc
- I *Web component* sono in grado di accedere ai servizi offerti da un Web container come:
 - Smistamento delle richieste
 - Sicurezza
 - Concorrenza
 - Gestione della memoria

Java Servlet (Controller)

- E' una classe scritta in linguaggio Java che estende le capacità di un server.
- Le funzionalità principali di una *Servlet* sono:
 - Elaborazione o memorizzazione di dati provenienti da form HTML;
 - Generazione di contenuti dinamici (pagine Web) a seconda dei parametri della richiesta inviata;
 - Gestione delle informazioni di stato che non esistono nel protocollo HTTP stateless.
- Comunicano attraverso un qualsiasi protocollo Client-Server, ma sono principalmente utilizzate con il protocollo HTTP.

Java Server Page (View)

- JSP e' una tecnologia di programmazione Web in Java per lo sviluppo di WA.
- Forniscono contenuti dinamici in formato HTML o XML.
- Si basa su un insieme di tag speciali con cui possono essere invocate funzioni predefinite o codice Java (e.g., <% %>).
- Puo' essere vista come una rappresentazione ad alto livello di una Servlet.



JavaBeans (Model)

- Classe che incapsula molti oggetti in un singolo oggetto (bean).
- Classe Java appositamente costruita e programmata in accordo alle specifiche delle API di JavaBeans.
- Oggetti serializzabili che permettono l'accesso a diverse proprieta' attraverso i cosiddetti metodi *getter* e *setter*.
- Il modello rappresenta un oggetto POJO per il trasporto dei dati.

Vantaggi	Svantaggi
Controllo delle proprieta', eventi e metodi dei beans esposti ad altre applicazioni.	E' soggetto ad essere instanziato con un <i>stato invalido</i> avendo un costruttore nullo.
Registra eventi da altri oggetti e puo' generare eventi da poter inviare ad altri oggetti.	Sono oggetti intrinsecamente <i>mutabili</i> mancando cosi del vantaggio offerto dagli oggetti immutabili.
Impostazioni di configurazione di un bean possono essere memorizzati in modo persistente.	Il possedere molti metodi getter e setter puo' portare una quantita' immensa di <i>boilerplate code</i> .

Configurazione dell'ambiente

- Una volta lanciato Netbeans...
- **Creare un nuovo progetto:** *Java Web-> Web Application*

🚺 New Project		×	New Web Applic	ation		×
Steps	Choose Project		Steps	Name and Loc	ation	
1. Choose Project	۹ Filter:		1. Choose Project	Project Name:	MyfirstWebbApp	
2	Categories:	Projects:	Location	Project Location:	C:\Users\davidenardone\Documents\NetBeansProjects	Browse
	-W Java -W JavaFX -W Java Web	 Web Application Web Application with Existing Source Web Free-Form Application 	 Server and Settings Frameworks 	Project Folder:	lenardone\Documents\NetBeansProjects\MyfirstWebbApp	
	Java EE			Use Dedicate	d Folder for Storing Libraries	
				Libraries Folder:		Browse
	-♥ NetBeans Modules ⊕♥ Samples				Different users and projects can share the same compilation libraries (see Help for details).	
		< >>				
	Description:	lieption in a standard IDE project. A				
	standard project uses an IDE- and debug your project.	generated build script to build, run,				
	< Back N	ext > Finish Cancel Help			< Back Next > Finish Cancel	Help

Configurazione dell'ambiente

- Se non presente di *default*, aggiungere il Server Apache Tomcat 8.x
- e modificare le *credenziali utente* di Tomcat

🜍 New Web Applic	ation	×		Add Server Insta	nce			×
Steps	Server and Set	tings						
1. Choose Project	Add to Enterprise	e Application: <none></none>		Steps	Installation an	d Login Details	5	
Location	Server:	Apache Tomcat 8.0.27.0 Y Add		1. Choose Server	Specify the Serve	er Location (Cata	lina Home) and login details	
Settings	Java EE Version:	Java EE 7 Web 🗸		2. Installation and Login	Server Location:	e Software Four	ndation\Apache Tomcat 8.0.27	Browse
4. Frameworks	Context Path:	/MyfirstWebbApp		Details	Use Private C	onfiguration Fold	er (Catalina Base)	
					Catalina Base:			Browse
					Enter the creder	ntials of an existin	ng user in the manager or mana	ger-script role
					Username:	davide		
				5	Password:	•••••		
						Create user if	it does not exist	
		< Back Next > Finish Cancel Help	-			< Back	Next > Finish Cance	l Help

Configurare il Server Apache Tomcat

Selezionare la voce Services -> Cliccare su Properties

Projects	Files	Services × -	
🗄 🛢 Dat	tabase	s	
🗄 🧟 We	b Serv	vices	
🗆 🛅 Ser	vers		
	\pach-	Tomost or TomEE	
🗄 📠 Ma	ven	Start	
🗄 🥗 Clo	ud	Start in Debug Mode	
🗄 🐌 Hu	dsor	Start in Profile Mode	
🗄 🖶 Doo	cker	Restart	
🗄 🐻 Tas	sk Re	Stop	
	lest	Stop	
± 🛎 Sel	eniu	Refresh	
		Rename	
		Remove	
		Edit server.xml	
		View Server Log	
		View Conver Output	
		view server Output	
		Properties	

🕺 Apache Tomcat or TomEE	Server Name: Apache Tomcat or TomEE					
	Server Type: Apache Tomcat or TomEE					
	Deployment Classes Sources Java					
	Connection Startup Platform					
	Catalina Home: C:\Program Files\Apache Software Found					
	Catalina Base: C:\Program Files\Apache Software Found					
	Credentials of an existing user in the "manager-script" rol					
	Password: davide Hide					
	Server Port: 9090					
	Server Port: 9090 Shutdown Port: 9191					
	Server Port: 9090 Shutdown Port: 9191 Enable HTTP Monitor					
	Server Port: 9090 Shutdown Port: 9191 Enable HTTP Monitor Note: Changes will take affect the pext time you start the					

Lanciare il Server Apache Tomcat

Selezionare la voce Services -> Cliccare su Start



Permettere l'accesso in rete di Java e Netbeans.

Configurare i ruoli d'accesso a Tomcat

Image Source Hist Image Source Hist Image Close Ctrl+W Image Source Hist Image Image Close Ctrl+W Image Source Hist Image Close All Ctrl+Shift+W Image Start Image Close All Ctrl+Shift+W Image Start Image Start Close Other Image Start Start Image Start Close Other Image Start Start Image Start Start Image Start Start Image Start Start Image Start Start Start Close Other Maximize Shift+Escape Image Start Start Start Dock Alt+Shift+D Start Start Start Start Start Start Start Image Start Start Start Start Start Start Start Image Start Start Start Start Start Start Start	Projects Files Service	s × - Ctart Dage	🖻 index.	html 🔺 🖺 server	r annal X	
Web Services Source This Servers 1 Servers 1 Web Apping Start Start Start Maximize Shift+Escape Restart 8 Maximize Shift Left Shift Right Shift Left Sop 9 Maximize Shift Right Maximize Rename Selenium Server Remove Himitat WitHout View Server Log 17 View Server Output <td< th=""><th>🗉 🛢 Databases</th><th>Source Hist</th><th>Source</th><th>History 🕼</th><th>Close</th><th>Ctrl+W</th></td<>	🗉 🛢 Databases	Source Hist	Source	History 🕼	Close	Ctrl+W
Servers 1 <100 2 <1 Close Other Start Start 2 <1 3 Close Other Start Start Start 4 Contribution Maximize Shift+Escape Maximize Start Start Start Point Start Start Start Start Maximize Start Start Start Start Start Start Start Start Maximize Start	🗄 🧟 Web Services	Source This	1	xml ver</th <th>Close All</th> <th>Ctrl+Shift+W</th>	Close All	Ctrl+Shift+W
Start 3 License Start 4 contrit Maximize Shift+Escape Maximize Start 5 this wc Float Alt+Shift+D Maximize Start in Debug Mode 5 this wc Float Alt+Shift+D Maximize Start in Profile Mode 6 The ASE Dock Alt+Shift+D Maximize Start 8 the Lic Shift Left Shift Left Maximize Start 8 the Lic Shift Left Shift Left Maximize Start 8 the Lic Shift Left Shift Left Maximize Start 8 the Lic Shift Right Shift Right Maximize Start 9 11 New Document Tab Group Colone Maximize Selenium Server 13 distrit Colone New Document Tab Group Maximize Select in Projects Diff To Copy File Path Select in Projects View Server Output 18 <1 Note Editors Split Split	Servers		2	</th <th>Close Other</th> <th></th>	Close Other	
Image: Start in Debug Mode 5 this we contrine Float Image: Start in Debug Mode 5 this we float Float Image: Image: Start in Profile Mode 6 The ASE Dock Alt+Shift+D Image: Ima	Apache Tom	Start	3	License	Maximize	Shift+Escape
Image: Start in Profile Mode G The Ast Could Dock Alt+Shift+D Image: Start in Profile Mode G The Ast Could Dock Alt+Shift+D Image: Start in Profile Mode G The Ast Could Dock Alt+Shift+D Image: Start in Profile Mode G The Ast Could Shift Left Shift Left Image: Start in Profile Mode G The Ast Could Shift Left Shift Right Image: Start in Profile Mode G The Ast Could Shift Left Shift Right Image: Start in Profile Mode G The Ast Could Shift Right Shift Left Image: Start in Profile Mode Start in Profile Mode G The Ast Could Shift Right Image: Start in Profile Mode Start in Profile Mode Image: Shift Right Shift Right Shift Right Image: Start in Profile Mode Refresh Image: Shift Right Clone New Document Tab Group New Document Tab Group Image: Start in Profile Mode Image: Shift Right Select in Projects Diff To Select in Projects Image: Start in Profile Start Image: Shift Right Image: Shift Right Select in Projects		Start in Debug Mode	4	contrib this wo	Float	cime Escape
Image: Selenium Server Log 7 (the "I Shift Left Image: Selenium Server Output 8 the Lic Shift Left Image: Selenium Server Output 10 htt Clone Image: Selenium Server Output 12 Unless Collapse Document Tab Group Image: Selenium Server Output 13 distrik Select in Projects Image: Selenium Server Output 18 <1 Note Image: Selenium Server Server Output 19 defi Editors Image: Selenium Server Server Output 19 defi Editors	🧧 / 🗟 /manag	Start in Profile Mode	6	The ASE	Dock	Alt+Shift+D
Image: Cloud Image: Cloud <td< th=""><th>🗄 📾 Maven Reposit</th><th>Restart</th><th>7</th><th>(the "I</th><th>Shift Left</th><th></th></td<>	🗄 📾 Maven Reposit	Restart	7	(the "I	Shift Left	
Image: Selenium Server Addison Builde Refresh 10 htt Clone Image: Selenium Server Addison Server Addison Server Couput Rename 10 htt Clone Image: Selenium Server Addison Server Couput Remove 11 New Document Tab Group Collapse Document Tab Group Image: Selenium Server Addison Server Couput Image: Select in Projects Select in Projects Diff To View Server Couput 18 Copy File Path Selections Properties 20 Doct Split Split		Stop	8	the Lid	Shift Right	
Image: Selenium Server Rename Image: Selenium Server Remove		Refresh	10	htt	Clone	
Image: Selenium Server Remame Image: Selenium Server Image: Seleniu	🗄 🐻 Task Repositor	D	11	Imlaga	New Document Tab G	roup
Kernove 14 WITHOUT Select in Projects Edit server.xml 15 See the limitat Diff To View Server Log 17 > Copy File Path View Server Output 18 Note</td Editors Properties 20 Doct Split	I S Test Driver	Rename	12	distrib	Collapse Document Ta	b Group
Edit server.xml 15 See the limitat Diff To View Server Log 17 > Copy File Path View Server Output 18 Note</td Editors Properties 20 Docu Split		Kemove	14	WITHOUT	Select in Proiects	
View Server Log 17 > Copy File Path View Server Output 18 Note</td Editors Properties 20 Docu Split		Edit server.xml	15	See the	Diff To	
View Server Output 18 Note</th 19 defi Editors Properties 20 Docu Split		View Server Log	10	>	Copy File Path	
Properties 20 Docu Split		View Server Output	18	Note</th <th>Editors</th> <th></th>	Editors	
		Properties	19	defi Docu	Split	

Aprire il file *tomcat-users.xml* al path precedentemente copiato
Inserire la seguente riga in basso al file:

<user username="..." password="..." roles="manager-script,admin-gui,manager-gui" />

Accedere alla GUI di Apache Tomcat

- Aprire un browser e accedere al server locale di Tomcat all'indirizzo: localhost:<server port> (e.g., localhost:9090)
- Server Status: GUI per il monitoraggio delle risorse occupate dalla/e WA.
- Manager App: GUI per la visualizzazione dei dettagli di una sessione e/o per la rimozione di attributi/variabili di sessione.

Home Documentation Configuration	Examples Wiki Mailing Lists	Find Help
Apache Tomcat/8.0.27	*	The Apache Software Foundation http://www.apache.org/
If you're seeing th with the security Considera Manager Application Clustering/Session Developer Quick Start Tomcat Setup Realms &	nis, you've successfully installed Tomo eading: ntions HOW-TO on HOW-TO Replication HOW-TO	eat. Congratulations! Server Status Manager App Host Manager
First Web Application JDBC Date	aSources	Tomcat Versions
Managing Tomcat For security, access to the <u>manager webapp</u> is restricted. Users are defined in: \$CATALINA_HOME/conf/tomcat-users.xml In Tomcat 8.0 access to the manager application is split between different users. <u>Read more</u> Release Notes Changelog <u>Migration Guide</u> Security Notices	Documentation Tomcat 8.0 Documentation Tomcat 8.0 Configuration Tomcat 8.0 Configuration Tomcat Wiki Find additional important configuration information in: \$CATALINA_HOME/RUNNING.txt Developers may be interested in: Tomcat 8.0 Bug Database Tomcat 8.0 JavaDocs Tomcat 8.0 SVIN Repository	Getting Help EAQ and Mailing Lists The following mailing lists are available: Tomat-announcements, releases, security vulnerability notifications. (Low volume). Merse Support and discussion Taglibs-user User support and discussion for <u>Apache Taglibs</u> Development mailing list, including commit messages
Other Downloads Other Documental Iomcat Connectors Iomcat Connectors Tomcat Native mod ik Documentati Taglibs Tomcat Native Deplover Deplover	tion Get Involved Miscell Overview Contact on SVN Repositories Legal Mailing Lists Sponsor Wiki Thanks	aneous Apache Software Foundation <u>Who We Are</u> <u>Heritage</u> <u>Apache Home</u> <u>Resources</u>

Copyright ©1999-2016 Apache Software Foundation. All Rights Reserved

Installare MySQL Server

- Lanciare l'installer;
- 2. Scegliere come tipologia di installazione qualle 'custom';
- 3. Scegliere come prodotti:
 - 1. MySQL Server 5.x
 - 2. Connector/J 5.x
- Impostare la password per l'utente `root';
- Aggiungere (eventualmente) un nuovo utente con relativa password;
- 6. Applicare/eseguire le configurazioni impostate.

Nota: In alternativa allo stesso modo di *MySQL Server* e' possibile utilizzare la versione di *JavaDB* pre-installata in Netbeans.

Configurazione Database Netbeans

- Una volta scaricato e installato MySQL Server, effettuare i seguenti steps:
 - Services -> Register MySQL Server;
 - 2. Inserire il <u>nome dell'utente</u> e <u>password amministratore</u> impostati nella fase d'installazione;
 - 3. Cliccare *Admin Properties* ed inserire il path dell'eseguibile di MySQL (mysqld.exe) alla voce *Path to start command;*
 - 4. Connettersi al database, creare un database di test e concedere all'utente tutti i permessi al db appena creato.
 - 5. Infine connettersi al database.

Configurazione Database Netbeans (cont.)

Creare una nuova tabella con gli attributi sotto riportati



Attributi

Name <VARCHAR> Surname <VARCHAR> City <VARCHAR> University <VARCHAR> Phone <VARCHAR> Mail <VARCHAR> Password <VARCHAR>

Configurazione Database Netbeans (cont.)

• Aggiungere i driver JDBC all progetto

		Start Page					
₩ <mark>myn</mark> ⊟ ₿ We	New	>	Project Propertie	s - myfirstwebapp			
	Build		Categories:				
	Clean and Build		Sources	Java Platform:	JDK 1.8 (Default)	~	Manage Platforms.
E So	Clean Generate Javadoc		 Frameworks Libraries 	Libraries Folder:	:		Browse
			• JavaScript Librari	ies Compile Proces	ssor Compile Tests Run Tes	sts	
	Run		• Bower	Compile-time L	ibraries:		
	Deploy		• CDNJS	Name		Package	Add Project
	Profile		 CSS Preprocessor ■ Build 	rs			Add Library
	Test RESTful Web Service	e	 Compiling 				Add JAR/Folde
Те	Test Test	Alt+F6	· · Packaging				
🔉 Co	Run Selenium Tests	Altro	• Documenting				Eait
	Run Scienium resis		License Headers				Remove
	Open Required Projects	>	Formatting				Move Up
	Close						Move Down
	Rename						
	Move						
	Copy						
	Delete	Delete					
				Compile-time II	braries are propagated to a	li library categories	
	Find	Ctrl+F		Build Require	ed Projects (Libraries and ac	ditional WAR conte	ent)
	Inspect and Transform						
- 10	Versioning	>					
vr X	History	>				ОК	Cancel Hel
hers	Properties						

I driver JDBC Mysql sono al path:

C:\Program Files\NetBeans 8.2\ide\modules\ext\mysql-connector-java-5.1.23-bin.jar

Esempio di Registrazione e Login

- In questo semplice esempio vedremo come gestire:
 - **1**. Registrazione utente
 - 2. Login utente
 - 3. Visualizzazione utenti
- Sara' creata:
 - Una classe Database per effettuare semplici operazioni di connessione al db, inserimento e reperimento dati da esso.
 - Diverse pagine JSP per la gestione dell'interfaccia grafica della WA.
 - Infine alcune Servlet per l'elaborazione dei dati ricevuti dal front-end.

Singleton Class Database (1)

- 1. Creare un proprio *package* (e.g., *classes*) nella cartella *Source Packages*
- 2. Creare la Classe *Database* nel package appena creato

package classes; import java.sql.*; import java.util.logging.*;

public class Database {

//static reference to itself
private static Database dblsntance;
public static final String URL = "jdbc:mysql://localhost/test";
public static final String USER = "<tuo_utente>";
public static final String PASSWORD = "<tua_password>";
public static final String DRIVER_CLASS = "com.mysql.jdbc.Driver";
private static Connection con ;
private static Statement stmt;

private Database() { /* private constructor*/ }

Singleton Class Database (2)

```
public static Database getInstance(){
     if(dblsntance == null) {
          dblsntance = new Database();
     return dblsntance;
public Connection getConnection(){
     if(con==null) {
          try {
               Class.forName(DRIVER_CLASS);
               con = DriverManager.getConnection(URL, USER, PASSWORD);
          } catch (SQLException | ClassNotFoundException ex) {
               Logger.getLogger(Database.class.getName()).log(Level.SEVERE, null, ex);
     return con;
```

JSP per SignUp

<%

session = request.getSession();
String flash_message = (String) session.getAttribute("flash_message");

%>

</head>

<div class="container">

<div class="row">

<% if (flash_message=="1") {%> <div class="alert alert-success text-center"> Registrazione effettuata con successo! <button type="button" class="close" data-dismiss="alert"></button>

</div>

<%}%>

<form id="register-form" action="SignUp" method="post" role="form" style="display: none;">

<div class="form-group">

<input type="text" name="name" id="name" tabindex="1" class="form-control" placeholder="Name" value=""> </div>

••

```
<div class="form-group">
```

<div class="row">

<div class="col-sm-6 col-sm-offset-3">

<input style="background-color: #029f5b" type="submit" name="register-submit" id="register-submit" tabindex="4"
class="form-control btn btn-register" value="Register Now">

</div>

</div>

</div>

Servelet Class SignUp

public class SignUp extends HttpServlet {

private HttpSession session;

protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

```
response.setContentType("text/html;charset=UTF-8");
```

try {

session = request.getSession();
String name = request.getParameter("name");

String password = request.getParameter("password"); ArrayList<String> credential = new ArrayList<String>(); credential.add(name);

credential.add(password);

Connection con = Database.getInstance().getConnection("test"); Database.getInstance().insertRow("users_db", credential);

```
session.setAttribute("flash_message", "1");
request.getServletContext().getRequestDispatcher("/first_page.jsp").forward(request, response);
```

```
} catch (SQLException ex) {
    Logger.getLogger(SignUp.class.getName()).log(Level.SEVERE, null, ex);
```

@Override protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException { processRequest(request, response);

@Override protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException { processRequest(request, response);

JSP per Login

<%

session = request.getSession();
String flash_message = (String) session.getAttribute("flash_message");

%>

</head>

<div class="container">

<div class="row">

<form id="register-form" action="CheckUser" method="post" role="form" style="display:block;">

<div class="form-group">

<input type="text" name="mail" id="name" tabindex="1" class="form-control" placeholder="mail" value="">

</div>

<div class="form-group">

<input type="password" name="password" id="password" tabindex="1" class="form-control" placeholder="Password" value="">

</div>

<div class="form-group">

<div class="row">

<div class="col-sm-6 col-sm-offset-3">

<input style="background-color: #029f5b" type="submit" name="login-submit" id="login-submit" tabindex="4" class="formcontrol btn btn-login" value="Log In">

</div>

</div>

Servlet Class Check User

public class Check User extends HttpServlet {

```
private HttpSession session;
```

protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

```
response.setContentType("text/html;charset=UTF-8");
try {
```

session = request.getSession(); String mail = request.getParameter("mail"); String password = request.getParameter("password"); Database.getInstance().getConnection("test");

if (Database.getInstance().checkUser(mail, password) == 1) {
 Map<String, String> info_user = Database.getInstance().getInfo("users", "mail", mail);
 session.setAttribute("info_user", info_user);
 getServletContext().getRequestDispatcher("/profile.jsp").forward(request, response);

else{

session.setAttribute("flash_message", "o");

getServletContext().getRequestDispatcher("/first_page.jsp").forward(request, response);

} catch (SQLException ex) {

Logger.getLogger(CheckUser.class.getName()).log(Level.SEVERE, null, ex);

@Override protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException { processRequest(request, response);

@Override protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException { processRequest(request, response);

JSP per Profile

session = request.getSession();

HashMap<String, String> info_user = (HashMap<String, String>) session.getAttribute("info_user");

%>

<%

</head>

</div>

<div class="container">

...

<div class="row"> <div class="col-sm-6 col-md-4"> </div><div class="col-sm-6 col-md-8"> <h4><% out.print(info_user.get("name") + " " + info_user.get("surname")); %> </h4> <small><cite title="San Francisco, USA"> <%out.print(info_user.get("city"));%> ITA <i class="glyphicon glyphicon-mapmarker"></i></cite></small> <i class="glyphicon glyphicon-phone-alt"></i></out.print(info_user.get("phone"));%>
br/><i class="glyphicon glyphicon-home"></i></out.print(info_user.get("university"));%>
br/><i class="glyphicon glyphicon-envelope"></i></out.print(info_user.get("mail"));%>
 <!-- Split button --> <div class="btn-group"> <button type="button" class="btn btn-primary">View all users</button> </div> </div></div>

JSP per user_table_page

<%

ArrayList<Map<String, String>> data = Database.getInstance().getAllRows("users");

%>

</head>

<div id="wrapper"></div>	
<h1>User Table</h1>	
<table <="" id="keywords" td=""><td>" cellspacing="o" cellpadding="o"></td></table>	" cellspacing="o" cellpadding="o">
<pre><thcode< pre=""></thcode<></pre>	
<t< td=""><td></td></t<>	
	<pre><tb><cpap>Name</cpap></tb></pre>
	<t< td=""></t<>
	City
	University
	Phone
	Mail
<%	
	<pre>for (int i = 0; i < data.size(); i++) {</pre>
	out.print(""):
	out print("" + data get(i) get("name") + "
	out print("=tdiss=langn>" adda.get(i).get(hanne) = ;
	out print("+ data.get(i).get("solutine") + "
	out print($+ data.get(i).get(-tity) + $
	out.print(+ data.get().get(university) +
	out.print(<ta> + data.get(i).get(phone) + </ta> ");
	out.print("" + data.get(i).get("mail") + "");
	out.print("");

%>

/table>

Esercizio

- Implementare un classe Servlet per la gestione del LogOut di un utente
 - Rimozione della sessione
 - Redirezione alla pagina di login

WEB-INF: file di configurazione

web.xml: descrittore di distribuzione della WA.

- Descrive le servlets e gli altri componenti che compongono la WA.
- Utilizzato per l'impostazione dei parametri d'inizializzazione e per i vincoli di sicurezza che si vuole fare rispettare alla WA.

Note: nel caso in cui tale file non sia disponibile nella cartella WEB-INF al path: **web/WEB-INF** del progetto, e' possibile reperirlo dal materiale messo a disposizione sulla piattaforma modificandolo appropriatemente.

classes: cartella contenente i file delle classi Java richieste dalla WA, che include entrambe le classi Servlet e non-Servlet.

lib: cartella dei file JAR, contenenti i file delle classi JAVA richieste per la WA.Librerie di classi di terze parti e/o driver JDBC.

WEB-INF: file di configurazione (cont.)

<?xml version="1.0" encoding="UTF-8"?>

<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/webapp_3_0.xsd">

<display-name>MyFirstWebApp</display-name></display-name>

<welcome-file>first_page.jsp</welcome-file>
</welcome-file-list>

<servlet>

<servlet-name>SignUp</servlet-name> <servlet-class>servlets.SignUp</servlet-class> </servlet> <servlet-mapping> <servlet-name>SignUp</servlet-name> <url-pattern>/SignUp</url-pattern> </servlet-mapping> <session-config> <session-timeout>10</session-timeout> </session-config>

Tomcat Web Application Manager

- Path: URL della WA
- Sessions: numero di sessioni della WA create
- **Commands:** sezione per la gesione della WA

Applications									
Path	Version	Display Name	Running	Sessions	Commands				
Ĺ	None specified	Welcome to Tomcat	true	<u>0</u>	Start Stop Reload Undeploy				
					Expire sessions with idle \geq 30 minutes				
/docs	None specified	Tomcat Documentation	true	<u>0</u>	Start Stop Reload Undeploy				
					Expire sessions with idle ≥ 30 minutes				
/examples	None specified	Servlet and JSP Examples	true	<u>0</u>	Start Stop Reload Undeploy				
					Expire sessions with idle \geq 30 minutes				
/host-manager	None specified	Tomcat Host Manager Application	true	<u>0</u>	Start Stop Reload Undeploy				
					Expire sessions with idle ≥ 30 minutes				
/manager	None specified	Tomcat Manager Application	true	<u>3</u>	Start Stop Reload Undeploy				
					Expire sessions with idle ≥ 30 minutes				
/myfirstwebapp	None specified	MyFirstWebApp	true	1	Start Stop Reload Undeploy				
					Expire sessions with idle ≥ 10 minutes				

Tomcat Web Application Manager

• WAR file to deploy: sezione per l'upload della WA

List the configured ciphers for each connector

Connector ciphers

ploy							
ploy directory or WAR file located on server							
Context Path (required): XML Configuration file URL: WAR or Directory URL: Deploy							
WAR file to deploy							
Select WAR file to upload Scegli file Nessun file selezionato Deploy							
agnostics							
Check to see if a web application has caused a memory leak on stop, reload or undeploy							
Find leaks This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems.							
SSL connector configuration diagnostics							

Server Information											
Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address				
Apache Tomcat/8.0.27	1.8.0_111-b14	Oracle Corporation	Windows 10	10.0	amd64	DESKTOP-GCI99R8	192.168.123.128				

Riferimenti

- Apache Tomcat, JavaBeans e JDBC
 - 1. <u>https://www.ntu.edu.sg/home/ehchua/programming/howto/Tomcat_HowTo.html</u> (Windows and Unix)
 - 2. <u>https://wolfpaulus.com/journal/mac/tomcat8/</u> (OS X)
 - 3. <u>https://www.mulesoft.com/tcat/tomcat-linux</u> (Linux)
 - 4. <u>https://www.tutorialspoint.com/jsp/jsp_java_beans.htm</u> (JavaBeans)
 - 5. <u>http://www.java-samples.com/showtutorial.php?tutorialid=552</u> (JavaBeans)
 - 6. <u>http://theopentutorials.com/tutorials/java/jdbc/jdbc-examples-introduction/</u> (JDBC example)
 - 7. <u>https://netbeans.org/kb/docs/ide/java-db.html</u> (Java DB Derby)

Contatti:

davide.nardone@studenti.uniparthenope.it

"Laboratorio di Architettura e Sistemi" IV piano lato NORD, stanza 432.