

Lezione 14: JEE Lab 1

Corso di *Programmazione Distribuita*

Vittorio Scarano

Università di Salerno



Laurea in Informatica

IL PROGRAMMA DELLA LEZIONE

[Introduzione](#)

Introduzione

[Mettiamo tutto insieme](#)

Mettiamo tutto insieme

[Netbeans](#)

Netbeans

[Primo approccio](#)

Primo approccio

[Maven](#)

Maven

[GlassFish](#)

GlassFish

[Gli esempi](#)

Gli esempi

[Conclusioni](#)

Conclusioni

PRESENTAZIONE

14: JEE Lab 1

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio
Maven
GlassFish

Gli esempi

Conclusioni



3/60

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio
Maven
GlassFish

Gli esempi

Conclusioni

PLAN

14: JEE Lab 1

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio
Maven
GlassFish

Gli esempi

Conclusioni



4/60

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio
Maven
GlassFish

Gli esempi

Conclusioni

UN ESEMPIO CON UN *warning*

- ▶ Primo esempio di una applicazione Java Enterprise
- ▶ Focus sulla struttura e l'utilizzo di CDI
- ▶ Esempio però limitato perché non introduciamo **ora** l'utilizzo di un server
- ▶ Struttura dell'esempio: una classe Main che:
 - ▶ crea un proprio container EE (Weld: reference implementation per CDI)
 - ▶ dalla quale chiede una istanza di BookService
 - ▶ su cui fa una semplice richiesta di creazione di un libro
- ▶ Non vedremo la parte di testing



5/60

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

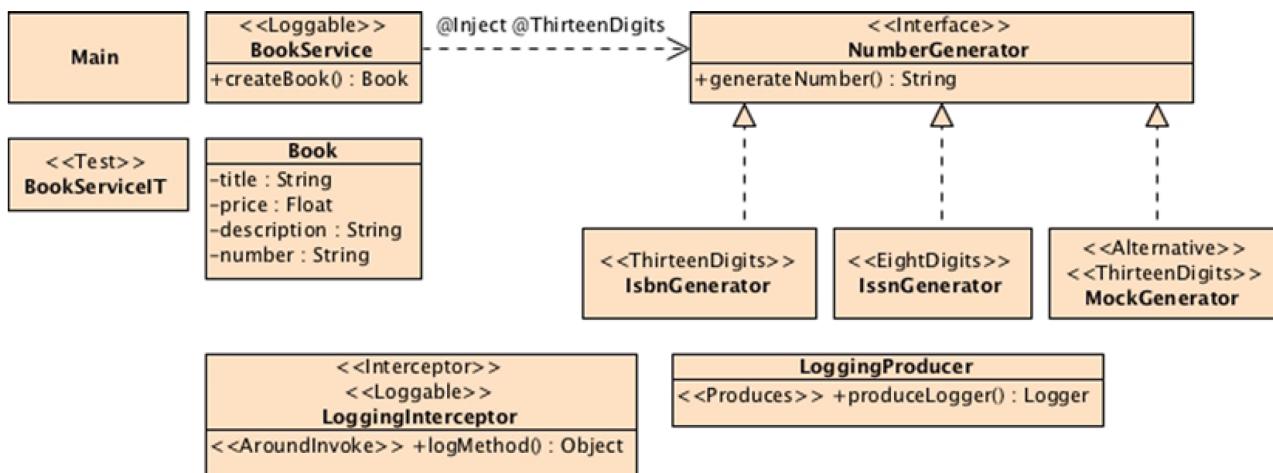
Maven

GlassFish

Gli esempi

Conclusioni

LA STRUTTURA DELL'ESEMPIO



Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



6/60

L'ESEMPIO: IL POJO Book

```
public class Book {
    private String title;
    private Float price;
    private String description;
    private String number;
    // Constructors, getters, setters
}
```

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



7/60

L'ESEMPIO: IL BookService

```
@Loggable
public class BookService {
    @Inject @ThirteenDigits
    private NumberGenerator numberGenerator;
    public Book createBook(String title, Float price, String
        description) {
        Book book = new Book(title, price, description);
        book.setNumber(numberGenerator.generateNumber());
        return book;
    }
}
```

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



8/60

L'ESEMPIO: IL NumberGenerator E SOTTOCLASSI - 1

```
public interface NumberGenerator {
    String generateNumber();
}
```

```
@ThirteenDigits
public class IsbnGenerator implements NumberGenerator {
    @Inject
    private Logger logger;
    @Loggable
    public String generateNumber() {
        String isbn = "13-84356-" + Math.abs(new
            Random().nextInt());
        logger.info("Generated ISBN : " + isbn);
        return isbn;
    }
}
```



9/60

Introduzione

Mettiamo tutto
insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni

L'ESEMPIO: IL NumberGenerator E SOTTOCLASSI - 2

```
@EightDigits
public class IssnGenerator implements NumberGenerator {
    @Inject
    private Logger logger;
    @Loggable
    public String generateNumber() {
        String issn = "8-" + Math.abs(new Random().nextInt());
        logger.info("Generated ISBN : " + issn);
        return issn;
    }
}
```



10/60

Introduzione

Mettiamo tutto
insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni

L'ESEMPIO: I QUALIFICATORI

```
@Qualifier
@Retention(RUNTIME)
@Target({FIELD, TYPE, METHOD})
public @interface ThirteenDigits { }
```

```
@Qualifier
@Retention(RUNTIME)
@Target({FIELD, TYPE, METHOD})
public @interface EightDigits { }
```

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



11/60

L'ESEMPIO: IL LOGGER

```
public class LoggingProducer {
    @Produces
    public Logger produceLogger(InjectionPoint
                                injectionPoint) {
        return Logger.getLogger(injectionPoint.getMember().
                                getDeclaringClass().getName());
    }
}
```

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



12/60

L'ESEMPIO: L'INTERCEPTOR

```

@Interceptor
@Loggable
public class LoggingInterceptor {
    @Inject
    private Logger logger;
    @AroundInvoke
    public Object logMethod(InvocationContext ic) throws Exception {
        logger.entering(ic.getTarget().getClass().getName(),
                        ic.getMethod().getName());
        try {
            return ic.proceed();
        } finally {
            logger.exiting(ic.getTarget().getClass().getName(),
                           ic.getMethod().getName());
        }
    }
}

```

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



13/60

L'ESEMPIO: IL BINDING DI INTERCEPTOR

```

@InterceptorBinding
@Target({METHOD, TYPE})
@Retention(RUNTIME)
public @interface Loggable {
}

```

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



14/60

L'ESEMPIO: LA CLASSE Main

```
public class Main {
    public static void main(String[] args) {
        Weld weld = new Weld();
        WeldContainer container = weld.initialize();

        BookService bookService = container.instance()
            .select(BookService.class).get();

        Book book = bookService.createBook("H2G2", 12.5f,
            "Geeky scifi
            Book");

        System.out.println(book);
    }
}
```

Creazione del Container
(in Java SE!)

Inizializzazione

Restituisce una istanza
costruita e “iniettata” di
BookService

Invochiamo un metodo

Stampiamo il libro creato

Chiudiamo il server

Introduzione

Mettiamo tutto
insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



15/60

PLAN

Introduzione

Introduzione

Mettiamo tutto
insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



16/60

PLAN

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



17/60

NETBEANS - 1: LA STRUTTURA

```

package org.agoncal.book.javaee7.chapter02;
import org.jboss.weld.environment.se.Weld;
import org.jboss.weld.environment.se.WeldContainer;

/**
 * @author Antonio Goncalves
 *         APRESS Book - Beginning Java EE 7 with Glassfish 4
 *         http://www.apress.com/
 *         http://www.antoniogoncalves.org
 */
public class Main {
    public static void main(String[] args) {
        Weld weld = new Weld();
        WeldContainer container = weld.initialize();
        BookService bookService = container.instance().select(BookService.class).get();
        Book book = bookService.createBook("H2G2", 12.5f, "Geeky scifi Book");
        System.out.println(book);
        weld.shutdown();
    }
}

```

Output:

```

ant -f /Users/vitsca/NetBeansProjects/JavaApplication2 clean
init:
deps-clean:
Created dir: /Users/vitsca/NetBeansProjects/JavaApplication2/build
Updating property file: /Users/vitsca/NetBeansProjects/JavaApplication2/build/built-clean.properties
Deleting directory /Users/vitsca/NetBeansProjects/JavaApplication2/build
clean:
BUILD SUCCESSFUL (total time: 0 seconds)

```

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



18/60

NUOVO PROGETTO - 1

New Project

Steps

1. Choose Project
2. ...

Choose Project

Categories:

- Java
- JavaFX
- Java Web
- Java EE
- HTML5
- Maven
- NetBeans Modules
- Samples

Projects:

- Java Application
- Java Class Library
- Java Project with Existing Sources**
- Java Free-Form Project

Description:

Imports an existing Java application into a standard IDE project. The Java application can have multiple source folders. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

Help < Back **Next >** Finish Cancel



Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni

NUOVO PROGETTO - 2

New Java Project with Existing Sources

**Steps****Existing Sources**



Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

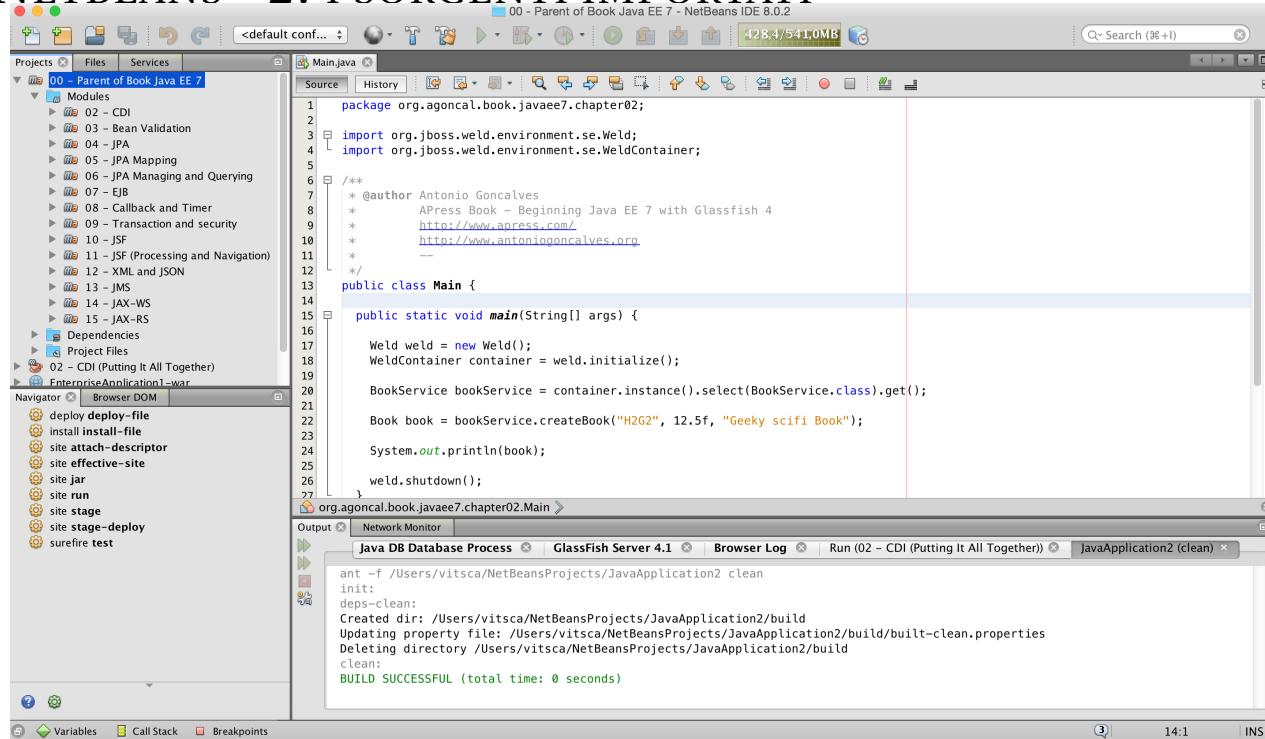
Maven

GlassFish

Gli esempi

Conclusioni

NETBEANS - 2: I SORGENTI IMPORTATI



```

1 package org.agoncal.book.javaee7.chapter02;
2
3 import org.jboss.weld.environment.se.Weld;
4 import org.jboss.weld.environment.se.WeldContainer;
5
6 /**
7  * @author Antonio Goncalves
8  * APress Book - Beginning Java EE 7 with Glassfish 4
9  * http://www.apress.com/
10 * http://www.antoniogoncalves.org
11 */
12
13 public class Main {
14
15     public static void main(String[] args) {
16
17         Weld weld = new Weld();
18         WeldContainer container = weld.initialize();
19
20         BookService bookService = container.instance().select(BookService.class).get();
21
22         Book book = bookService.createBook("H2G2", 12.5f, "Geeky scifi Book");
23
24         System.out.println(book);
25
26         weld.shutdown();
27     }
28 }

```

Output

```

Java DB Database Process GlassFish Server 4.1 Browser Log Run (02 - CDI (Putting It All Together)) JavaApplication2 (clean)
ant -f /Users/vitsca/NetBeansProjects/JavaApplication2 clean
init:
deps-clean:
Created dir: /Users/vitsca/NetBeansProjects/JavaApplication2/build
Updating property file: /Users/vitsca/NetBeansProjects/JavaApplication2/build/built-clean.properties
Deleting directory /Users/vitsca/NetBeansProjects/JavaApplication2/build
clean:
BUILD SUCCESSFUL (total time: 0 seconds)

```

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

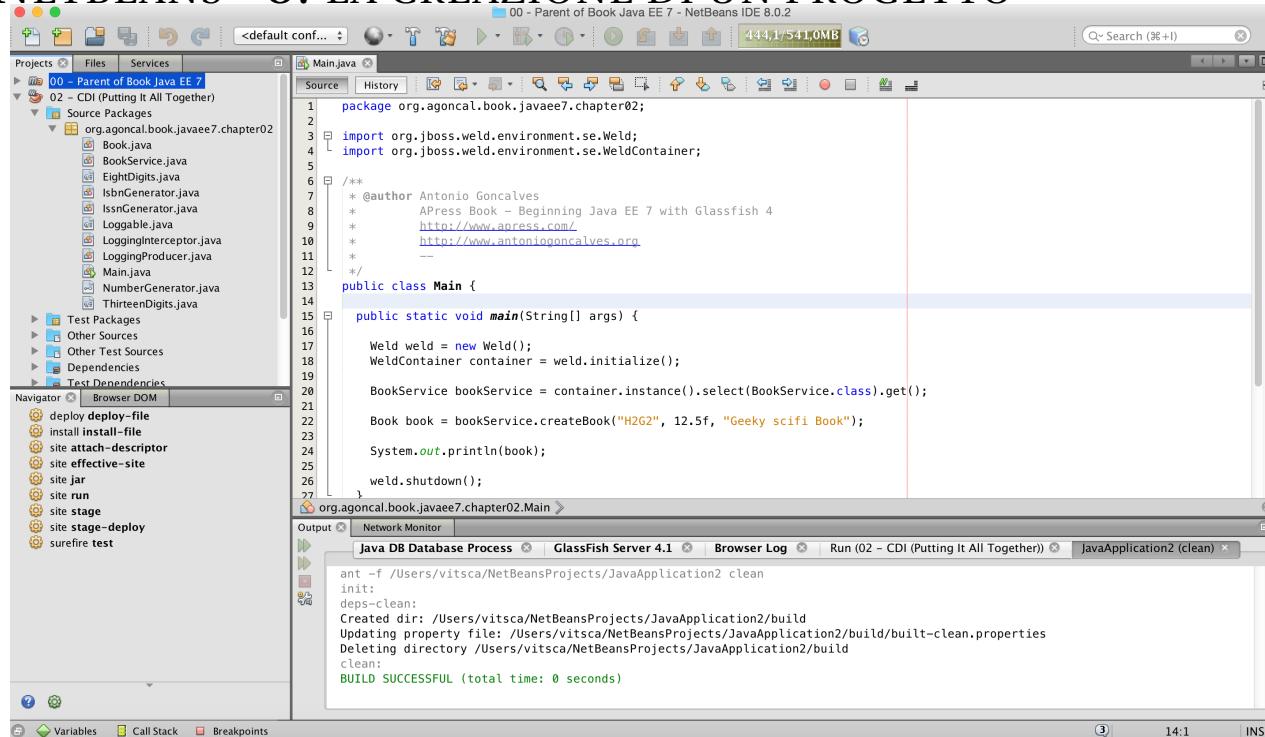
Gli esempi

Conclusioni



21/60

NETBEANS - 3: LA CREAZIONE DI UN PROGETTO



```

1 package org.agoncal.book.javaee7.chapter02;
2
3 import org.jboss.weld.environment.se.Weld;
4 import org.jboss.weld.environment.se.WeldContainer;
5
6 /**
7  * @author Antonio Goncalves
8  * APress Book - Beginning Java EE 7 with Glassfish 4
9  * http://www.apress.com/
10 * http://www.antoniogoncalves.org
11 */
12
13 public class Main {
14
15     public static void main(String[] args) {
16
17         Weld weld = new Weld();
18         WeldContainer container = weld.initialize();
19
20         BookService bookService = container.instance().select(BookService.class).get();
21
22         Book book = bookService.createBook("H2G2", 12.5f, "Geeky scifi Book");
23
24         System.out.println(book);
25
26         weld.shutdown();
27     }
28 }

```

Output

```

Java DB Database Process GlassFish Server 4.1 Browser Log Run (02 - CDI (Putting It All Together)) JavaApplication2 (clean)
ant -f /Users/vitsca/NetBeansProjects/JavaApplication2 clean
init:
deps-clean:
Created dir: /Users/vitsca/NetBeansProjects/JavaApplication2/build
Updating property file: /Users/vitsca/NetBeansProjects/JavaApplication2/build/built-clean.properties
Deleting directory /Users/vitsca/NetBeansProjects/JavaApplication2/build
clean:
BUILD SUCCESSFUL (total time: 0 seconds)

```

14: JEE Lab 1

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



22/60

NETBEANS - 4: LA GESTIONE DEI SERVER

```

1 package org.agoncal.book.javaee7.chapter02;
2
3 import org.jboss.weld.environment.se.Weld;
4 import org.jboss.weld.environment.se.WeldContainer;
5
6 author Antonio Goncalves
7 APRESS Book - Beginning Java EE 7 with Glassfish 4
8 http://www.apress.com/
9 http://www.antoniogoncalves.org
10
11
12 public class Main {
13
14     public static void main(String[] args) {
15         Weld weld = new Weld();
16         WeldContainer container = weld.initialize();
17
18         BookService bookService = container.instance().select(BookService.class).get();
19
20         Book book = bookService.createBook("H2G2", 12.5f, "Geeky scifi Book");
21
22         System.out.println(book);
23
24         weld.shutdown();
25
26     }
27 }

```

Output:

```

ant -f /Users/vitsca/NetBeansProjects/JavaApplication2 clean
Created dir: /Users/vitsca/NetBeansProjects/JavaApplication2/build
init:
deps-clean:
Updating property file: /Users/vitsca/NetBeansProjects/JavaApplication2/build/built-clean.properties
Deleting directory /Users/vitsca/NetBeansProjects/JavaApplication2/build
clean:
BUILD SUCCESSFUL (total time: 0 seconds)

```



23/60

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni

PLAN

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



24/60

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

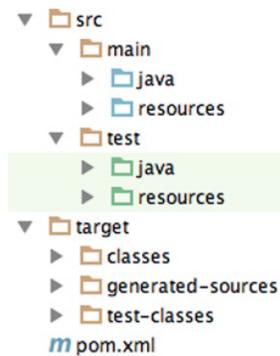
GlassFish

Gli esempi

Conclusioni

UNO STRUMENTO IMPORTANTE

- ▶ Netbeans permette di usare (integrato) Maven
- ▶ Uno strumento che permette di automatizzare compiti che vengono realizzati di solito nello sviluppo
 - ▶ quali generare codice, risorse, compilare classi e classi di test, eseguire unit test, test di unità e fare il packaging di applicazioni in archivi di vario tipo
- ▶ Maven è un approccio indipendente a questi compiti (non dipendente da linguaggio, da Sistema Operativo e da Integrated Development Environment)
- ▶ Basato su Apache Ant
- ▶ L'idea: "*convention over configuration*" detta la struttura del progetto (indipendente)



Introduzione
Mettiamo tutto
insieme
Netbeans
Apache
Ant
GlassFish
Gli esempi
Conclusioni



25/60

PLAN

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni

Introduzione
Mettiamo tutto
insieme
Netbeans
Primo approccio
Maven
GlassFish
Gli esempi
Conclusioni



26/60

APPLICATION SERVER DI RIFERIMENTO

- ▶ *reference Implementation*
- ▶ Non solo il server “di default” (scaricato con Java EE)
- ▶ ... ma anche un server che può essere usato per applicazioni industriali
- ▶ Origini che vanno al server Tomcat, nel 2005 (primo shipping nel 2006)
- ▶ Comunità viva e vitale, con documentazione di buona qualità
- ▶ Versione “community” (=free) e “supported” (con tool aggiuntivi di monitoraggio, ma con lo stesso core)

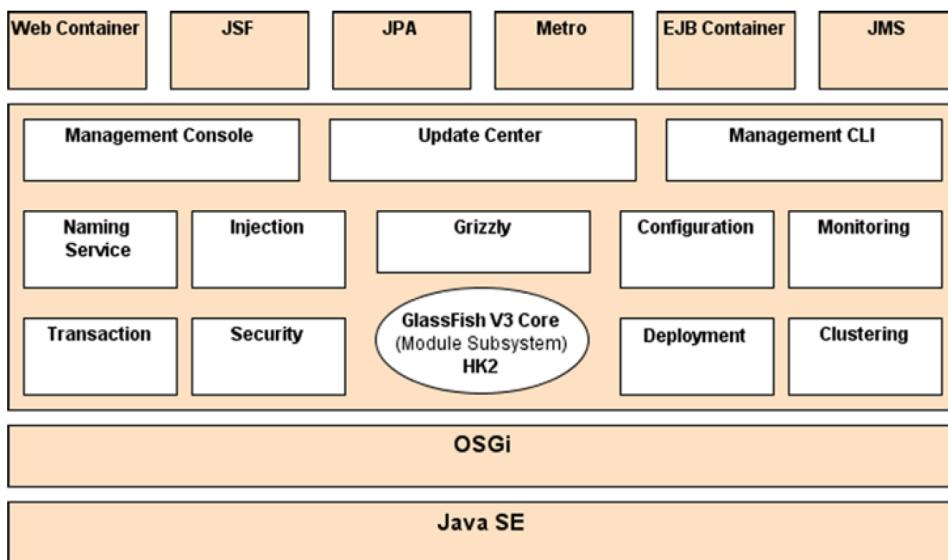
Introduzione
Mettiamo tutto insieme
Netbeans
Primo approccio
Maven
GlassFish
Gli esempi
Conclusioni

LA ARCHITETTURA DI GLASSFISH

- ▶ Come programmatore, abbiamo bisogno di capire la architettura a “grana grossa” del server (dettagli non strettamente influenti suMe funzionalità fornite)
- ▶ Costruito su un kernel modulare basato su standard OSGi
- ▶ OSGi è uno standard (commerciale Ericsson, IBM, Oracle) per la gestione dinamica di componenti nei sistemi, e la loro scoperta e utilizzo senza reboot
 - ▶ componenti che riconoscono l’aggiunta/rimozione di altre componenti a run-time
- ▶ Diversi sistemi sono costruiti su standard OSGi (Apache Felix implementazione free, Equinox (Eclipse))
- ▶ Caratteristiche: leggerezza (utile per testing in-site)
- ▶ Installazione standard: 8080 server, 4848 console di admin

Introduzione
Mettiamo tutto insieme
Netbeans
Primo approccio
Maven
GlassFish
Gli esempi
Conclusioni

LA ARCHITETTURA

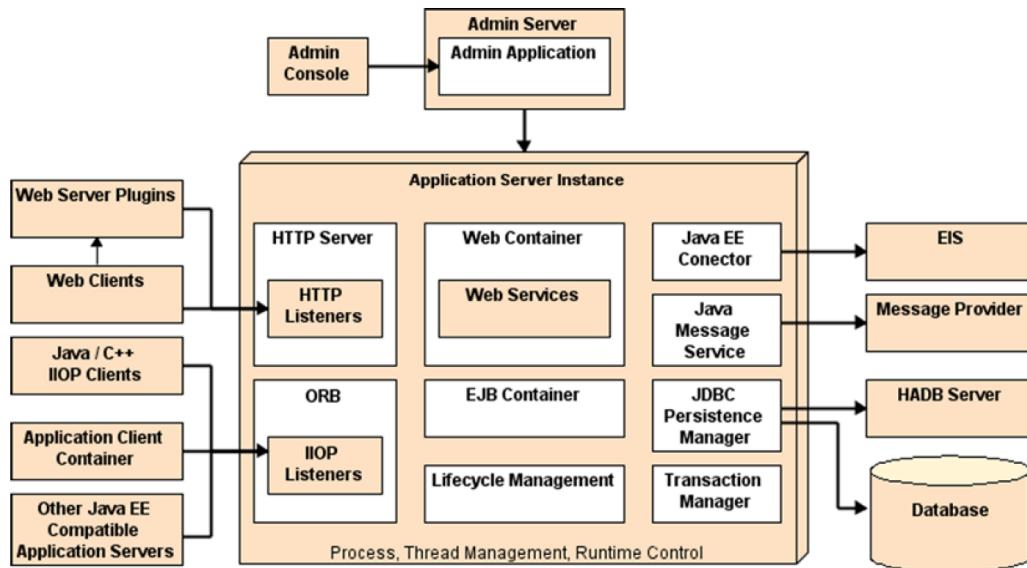


Introduzione
Mettiamo tutto insieme
Netbeans
Primo approccio
Maven
GlassFish
Gli esempi
Conclusioni



29/60

LE FUNZIONALITÀ



Introduzione
Mettiamo tutto insieme
Netbeans
Primo approccio
Maven
GlassFish
Gli esempi
Conclusioni



30/60

GLASSFISH - 1: LA CONSOLE

The screenshot shows the GlassFish Server Open Source Edition console interface. At the top, there's a header bar with 'Home', 'About...', 'User: admin | Domain: domain1 | Server: localhost', and a 'Help' link. Below the header is the title 'GlassFish™ Server Open Source Edition'. The left sidebar contains a tree view of server components: Common Tasks, Domain, Clusters, Standalone Instances, Nodes, Applications, Lifecycle Modules, Monitoring Data, Resources (with sub-options like Concurrent Resources, Connectors, JDBC, JMS Resources, JNDI, JavaMail Sessions, Resource Adapter Configs), Configurations, and Update Tool. The main content area is titled 'GlassFish Console - Common Tasks' and includes sections for GlassFish News, Deployment (List Deployed Applications, Deploy an Application), Administration (Change Administrator Password, List Password Aliases), Monitoring (Monitoring Data), Documentation (Open Source Edition Documentation Set, Quick Start Guide, Administration Guide, Application Development Guide, Application Deployment Guide), Update Center (Installed Components, Available Updates, Available Add-Ons), and Resources (Create New JDBC Resource, Create New JDBC Connection Pool).



31/60

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni

GLASSFISH - 2: LE APPLICAZIONI

This screenshot is similar to the previous one but shows the 'Applications' section in the sidebar expanded, revealing three deployed applications: EnterpriseApplication1-war, SoloWebApplicationCDIHelloWorld, and WebApplication1. The rest of the interface is identical to the first screenshot, displaying the 'Common Tasks' dashboard with its various sections and links.



32/60

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni

GLASSFISH - 3: I DETTAGLI

Common Tasks

- Domain
- server (Admin Server)
- Clusters
- Standalone Instances
- Nodes
- Applications
 - EnterpriseApplication1.war
 - SoloWebApplicationCDIHelloWorld
- Lifecycle Modules
- Monitoring Data
- Resources
 - Concurrent Resources
 - Connectors
 - JDBC
 - JMS Resources
 - JNDI
 - JavaMail Sessions
 - Resource Adapter Configs
- Configurations
 - default-config
 - server-config
 - Update Tool

Modify an existing application or module.

Name: SoloWebApplicationCDIHelloWorld
Status: Enabled
Virtual Servers: server
 Associates an Internet domain name with a physical server.
Context Root: /SoloWebApplicationCDIHelloWorld
 Path relative to server's base URL.
Implicit CDI Enabled
 Implicit discovery of CDI beans
Location: file:/Users/vitsca/NetBeansProjects/SoloWebApplicationCDIHelloWorld/build/web/
Deployment Order: 100
 A number that determines the loading order of the application at server startup. Lower numbers are loaded first. The default is 100.
Libraries:
Description:

Modules and Components (5)						
Module Name	Engines	Component Name	Type	Action		
SoloWebApplicationCDIHelloWorld	[web, weld]	-----	-----	Launch		
SoloWebApplicationCDIHelloWorld		default	Servlet			
SoloWebApplicationCDIHelloWorld		jsp	Servlet			
SoloWebApplicationCDIHelloWorld		FacesServlet	JSP			
SoloWebApplicationCDIHelloWorld		NewServlet	Servlet			



33/60

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni

PLAN

14: JEE Lab 1

Introduzione

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni

Gli esempi

Conclusioni



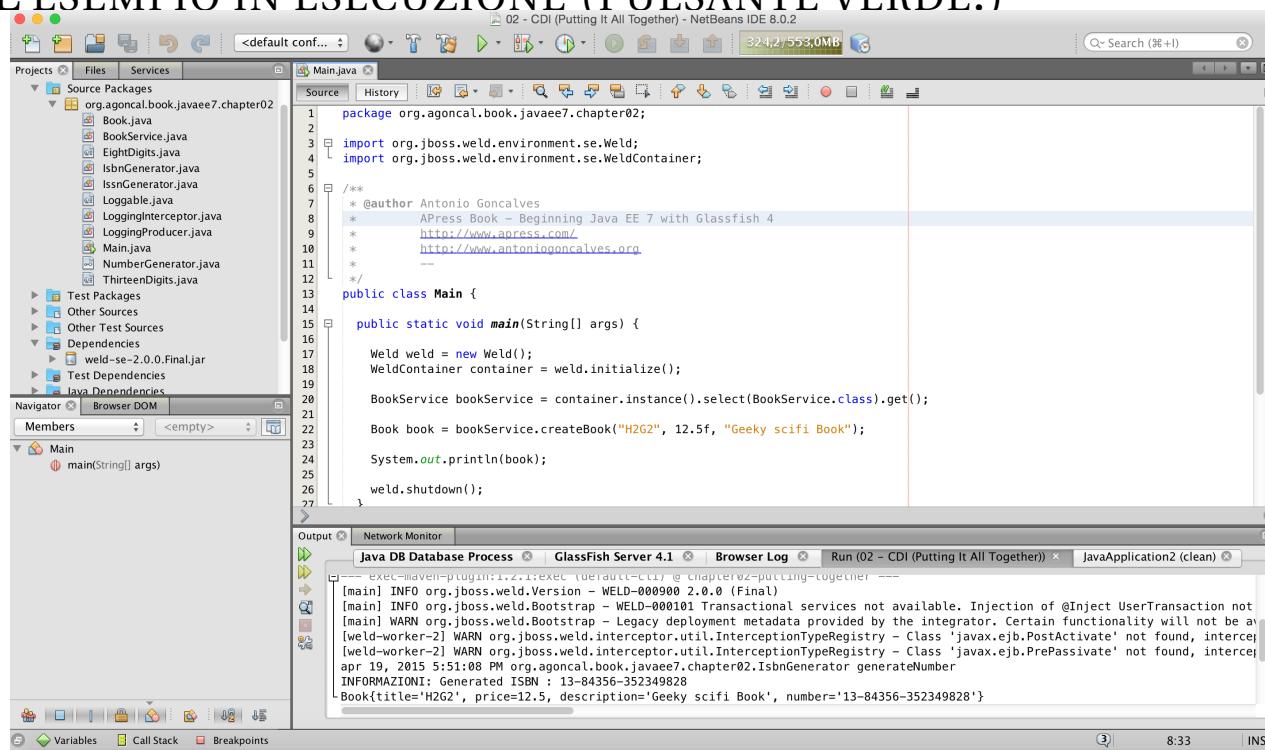
34/60

ESECUZIONE IN WELD

- ▶ La esecuzione dell'esempio prevede l'utilizzo di Weld
- ▶ La classe Main provvede a creare una istanza di BookService
- ▶ ... con tutte le risorse "iniettate" al punto giusto
- ▶ Maven provvede a scaricare Weld (se non presente tra le risorse) la prima volta
- ▶ L'esecuzione era semplicemente la stampa a video del libro creato

Introduzione
Mettiamo tutto insieme
Netbeans
Primo approccio
Maven
GlassFish
Gli esempi
Conclusioni

L'ESEMPIO IN ESECUZIONE (PULSANTE VERDE!)



```

1 package org.agoncal.book.javae7.chapter02;
2
3 import org.jboss.weld.environment.se.Weld;
4 import org.jboss.weld.environment.se.WeldContainer;
5
6 /**
7  * @author Antonio Goncalves
8  *         APress Book - Beginning Java EE 7 with Glassfish 4
9  *         http://www.apress.com/
10 *         http://www.antonionogoncalves.org
11 *
12 */
13 public class Main {
14
15     public static void main(String[] args) {
16
17         Weld weld = new Weld();
18         WeldContainer container = weld.initialize();
19
20         BookService bookService = container.instance().select(BookService.class).get();
21
22         Book book = bookService.createBook("H2G2", 12.5f, "Geeky scifi Book");
23
24         System.out.println(book);
25
26         weld.shutdown();
27     }
28 }

```

The Output window shows the following log:

```

--- exec-maven-plugin:1.2.1:exec (default-cli) @ Chapter02 Putting It All Together ---
[main] INFO org.jboss.weld.Version - WELD-000900 2.0.0 (Final)
[main] INFO org.jboss.weld.Bootstrap - WELD-000101 Transactional services not available. Injection of @Inject UserTransaction not available.
[main] WARN org.jboss.weld.Bootstrap - Legacy deployment metadata provided by the integrator. Certain functionality will not be available.
[weld-worker-2] WARN org.jboss.weld.interceptor.util.InterceptionTypeRegistry - Class 'javax.ejb.PostActivate' not found, interception for PostActivate events disabled.
[weld-worker-2] WARN org.jboss.weld.interceptor.util.InterceptionTypeRegistry - Class 'javax.ejb.PrePassivate' not found, interception for PrePassivate events disabled.
apr 19, 2015 5:51:08 PM org.agoncal.book.javae7.chapter02.IssnGenerator generateNumber
INFORMAZIONI: Generated ISBN : 13-84356-352349828
Book{title='H2G2', price=12.5, description='Geeky scifi Book', number='13-84356-352349828'}

```

Introduzione
Mettiamo tutto insieme
Netbeans
Primo approccio
Maven
GlassFish
Gli esempi
Conclusioni

COMPUTAZIONE LATO SERVER HTTP

- ▶ Componenti di una Web Application che rispondono ad una richiesta HTTP
- ▶ Ricevono la richiesta, con tutti i parametri del protocollo HTTP
 - ▶ header, metodo (GET, POST, ...), parametri passati, etc.
- ▶ Costruiscono una risposta in HTML che viene restituita al client
- ▶ Modello di computazione “antico” (un tempo si usavano le applicazioni esterne: Common Gateway Interface)
- ▶ Possibile inserirle in diversi ambienti server, da quelli più semplici (semplici Web container) ai veri e propri application server (GlassFish, JBoss, etc.)
- ▶ Vedremo un esempio di una servlet costruita con NetBeans per accedere ad una risorsa iniettata



WEB APP IN NETBEANS - 1

Introduzione
Mettiamo tutto insieme
Netbeans
Primo approccio
Maven
GlassFish
Gli esempi
Conclusioni



Help

< Back

Next >

Finish

Cancel

Choose Project

New Project

Steps

1. Choose Project
2. ...

Categories:

- Java
- JavaFX
- Java Web**
- Java EE
- HTML5
- Maven
- NetBeans Modules
- Samples

Projects:

- Web Application**
- Web Application with Existing Sources
- Web Free-Form Application

Description:

Creates an empty Web application in a standard IDE project. A standard project uses an **IDE-generated build script** to build, run, and debug your project.

WEB APP IN NETBEANS - 2

The screenshot shows the NetBeans IDE interface for a Java Web Application named "WebApplication1". The code editor displays the Facelet XML for "index.xhtml", which contains the following code:

```

<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
    <h:head>
        <title>Facelet Title</title>
    </h:head>
    <h:body>
        Hello from Facelets
    </h:body>
</html>

```

The Navigator panel shows a single entry: "html". The Output panel displays logs from GlassFish Server 4.1, including messages about registering OSGi services and loading the Mojarra library.

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

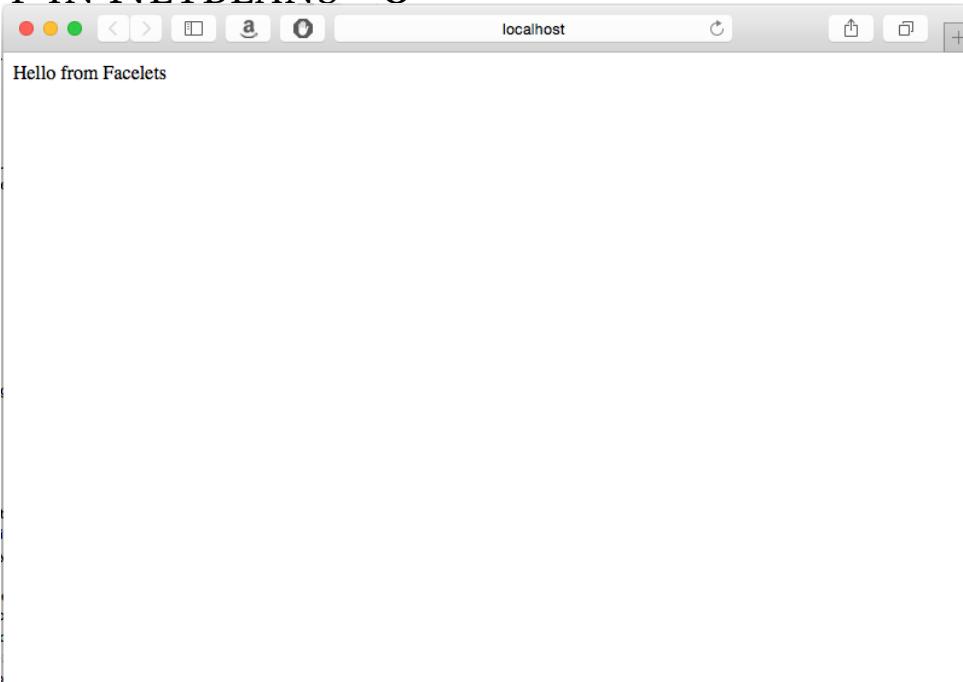
Gli esempi

Conclusioni



39/60

WEB APP IN NETBEANS - 3



Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



40/60

WEB APP IN NETBEANS - 4

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <context-param>
        <param-name>javax.faces.PROJECT_STAGE</param-name>
        <param-value>Development</param-value>
    </context-param>
    <servlet>
        <servlet-name>Faces Servlet</servlet-name>
        <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>Faces Servlet</servlet-name>
        <url-pattern>/faces/*</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <welcome-file-list>
        <welcome-file>faces/index.xhtml</welcome-file>
    </welcome-file-list>
</web-app>

```

Output:

```

compile-jsp:
Incrementally deploying WebApplication1
Completed incremental distribution of WebApplication1
run-deploy:
Browsing: http://localhost:8080/WebApplication1
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 0 seconds)

```

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



41/60

COSA VOGLIAMO FARE

- ▶ Scrivere una servlet che risponde a <http://localhost:8080/WebApplication1/NewServlet>
- ▶ Che riceve, iniettata, una istanza di un BookService
- ▶ ... dal quale stampa (nella risposta HTML) il libro
- ▶ Partiamo dall'includere l'esempio del libro
 - ▶ basta fare drag and drop
 dal progetto del libro nel package nel sorgente del nuovo progetto
- ▶ Risolvendo alcuni problemi: serve una libreria da includere per risolvere alcune dipendenze (right-click su progetto)
- ▶ Ovviamente è da cancellare la classe Main

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

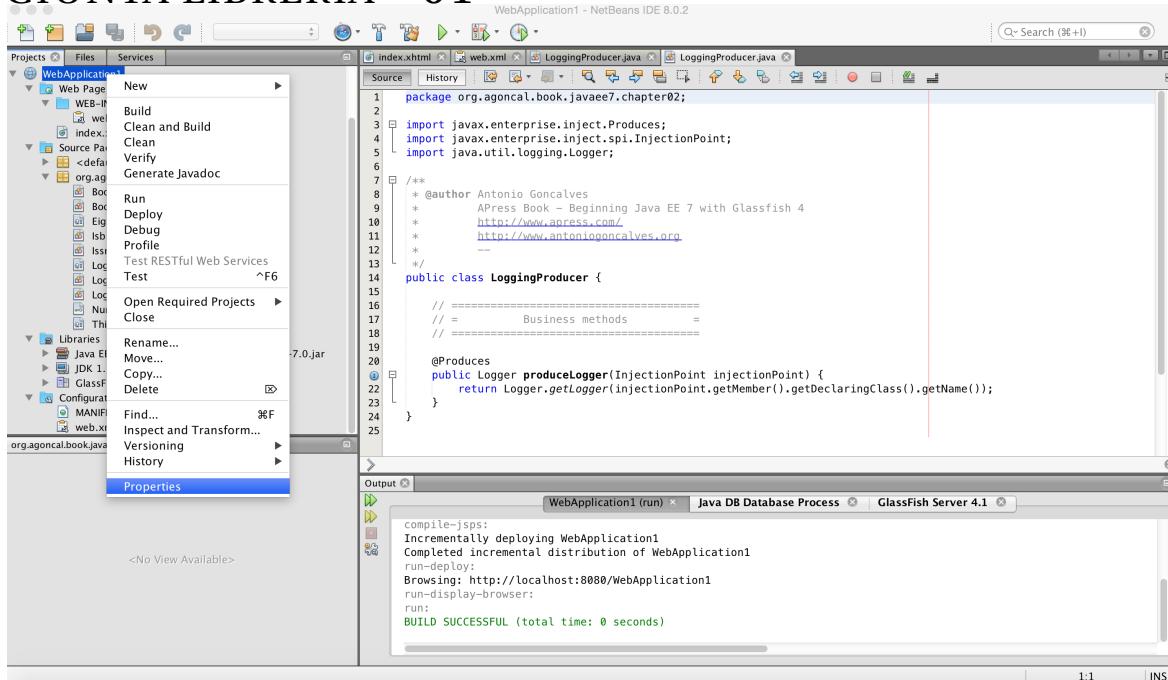
Gli esempi

Conclusioni



42/60

AGGIUNTA LIBRERIA - 01



43/60

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio
Maven
GlassFish

Gli esempi

Conclusioni

AGGIUNTA LIBRERIA - 02

Project Properties - WebApplication1

Introduzione

Mettiamo tutto insieme

Netbeans

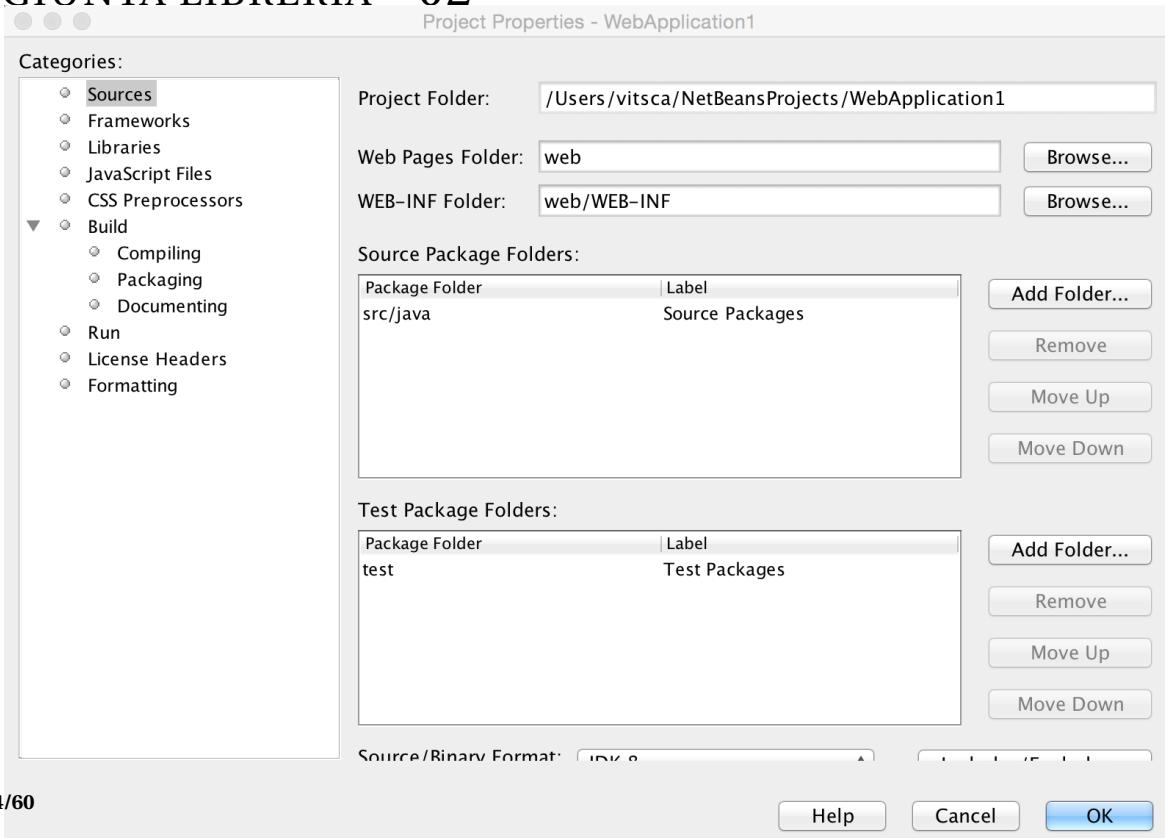
Primo approccio
Maven
GlassFish

Gli esempi

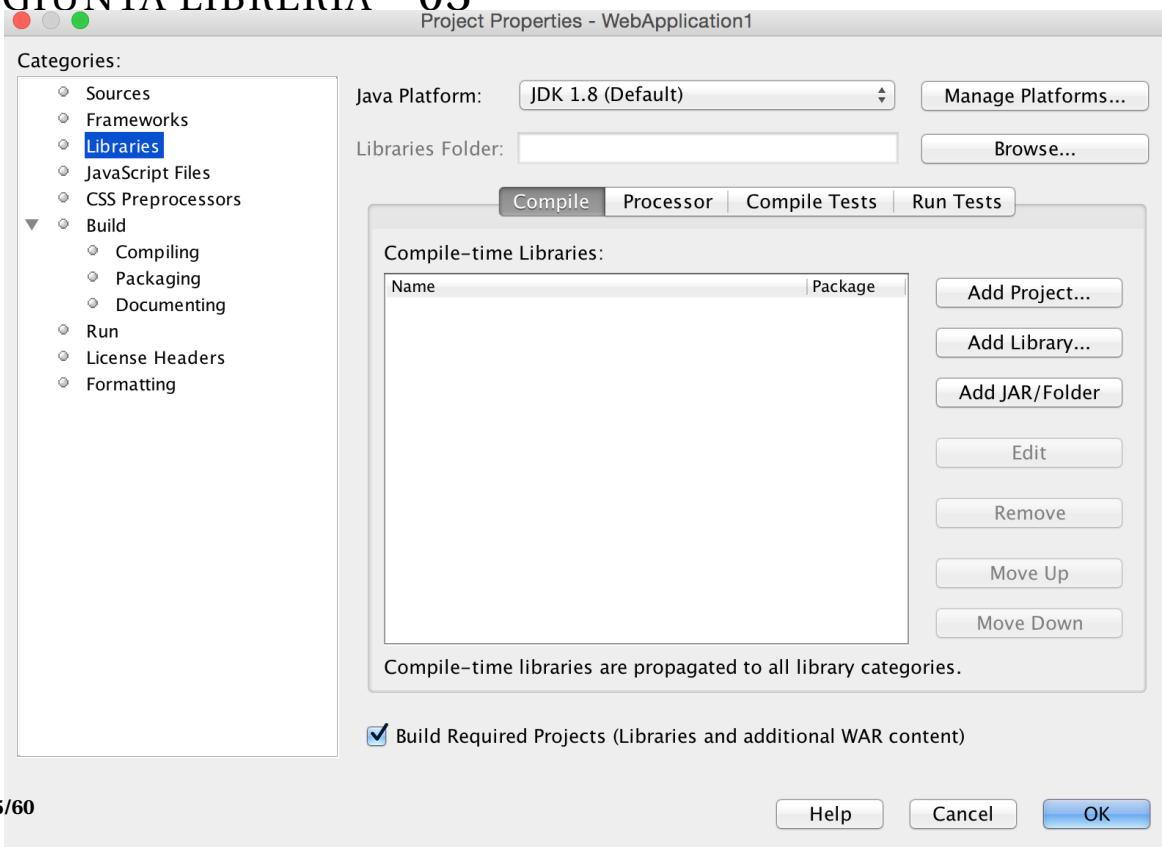
Conclusioni



44/60



AGGIUNTA LIBRERIA - 03



Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

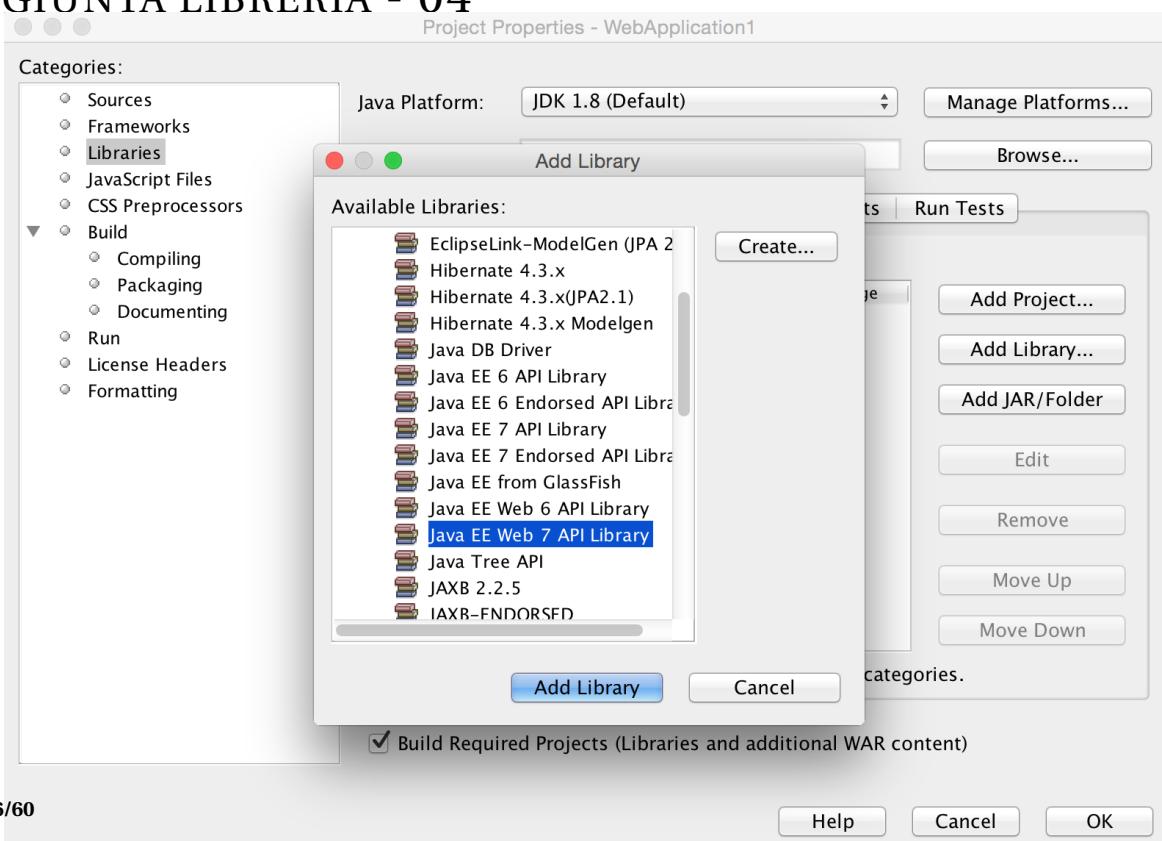
Maven

GlassFish

Gli esempi

Conclusioni

AGGIUNTA LIBRERIA - 04



Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

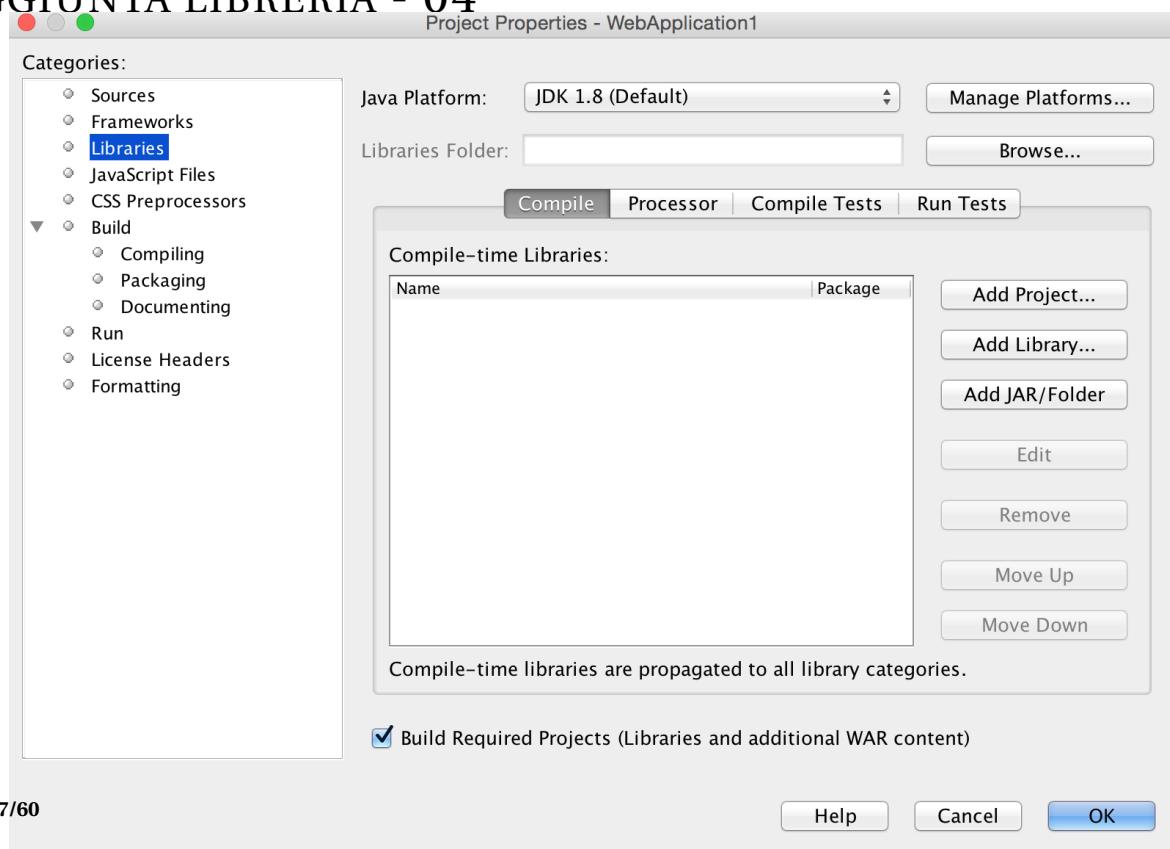
Maven

GlassFish

Gli esempi

Conclusioni

AGGIUNTA LIBRERIA - 04



Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

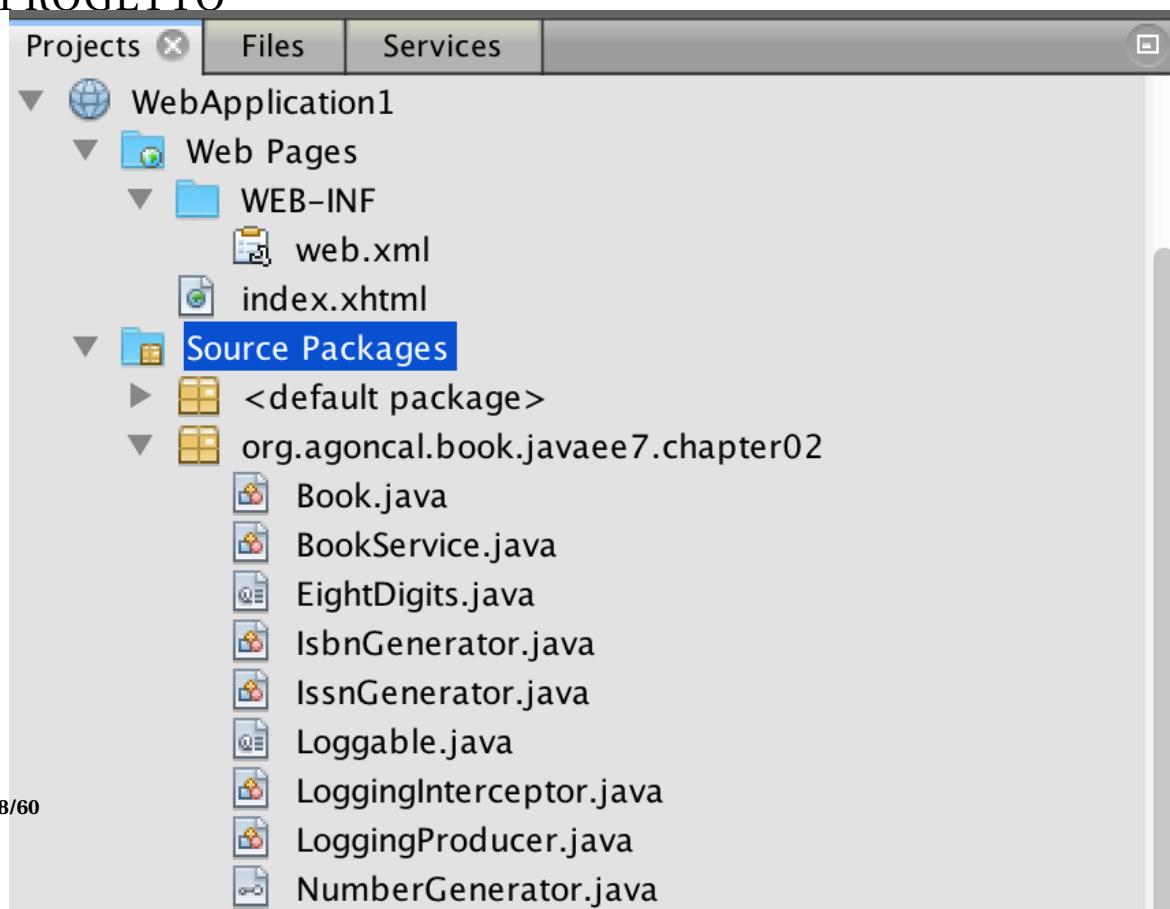
Maven

GlassFish

Gli esempi

Conclusioni

IL PROGETTO



Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni

CREAZIONE DELLA SERVLET - 1

14: JEE Lab 1

The screenshot shows the NetBeans IDE interface for a Java EE 7 application named "WebApplication1".

- Project Tree:** Shows the project structure with modules like "00 - Parent of Book Java EE 7", "02 - CDI (Putting It All Together)", "EnterpriseApplication1-war", "SoloWebApplicationCDIHelloworld", and "WebApplication1". Under "WebApplication1", there are "WEB-INF" and "Source" folders.
- Code Editor:** The main window displays the file "index.xhtml" with JavaServer Faces (JSF) code. A context menu is open over the code, showing options like "New", "Find...", "Paste ⌘V", "Tools", and "Properties".
- Output Window:** The bottom window shows the deployment log:

```
compile-jsp:
Incrementally deploying WebApplication1
Completed incremental distribution of WebApplication1
run-deploy:
Browsing: http://localhost:8080/WebApplication1
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 0 seconds)
```



49/60

CREAZIONE DELLA SERVLET - 2

14: JEE Lab 1

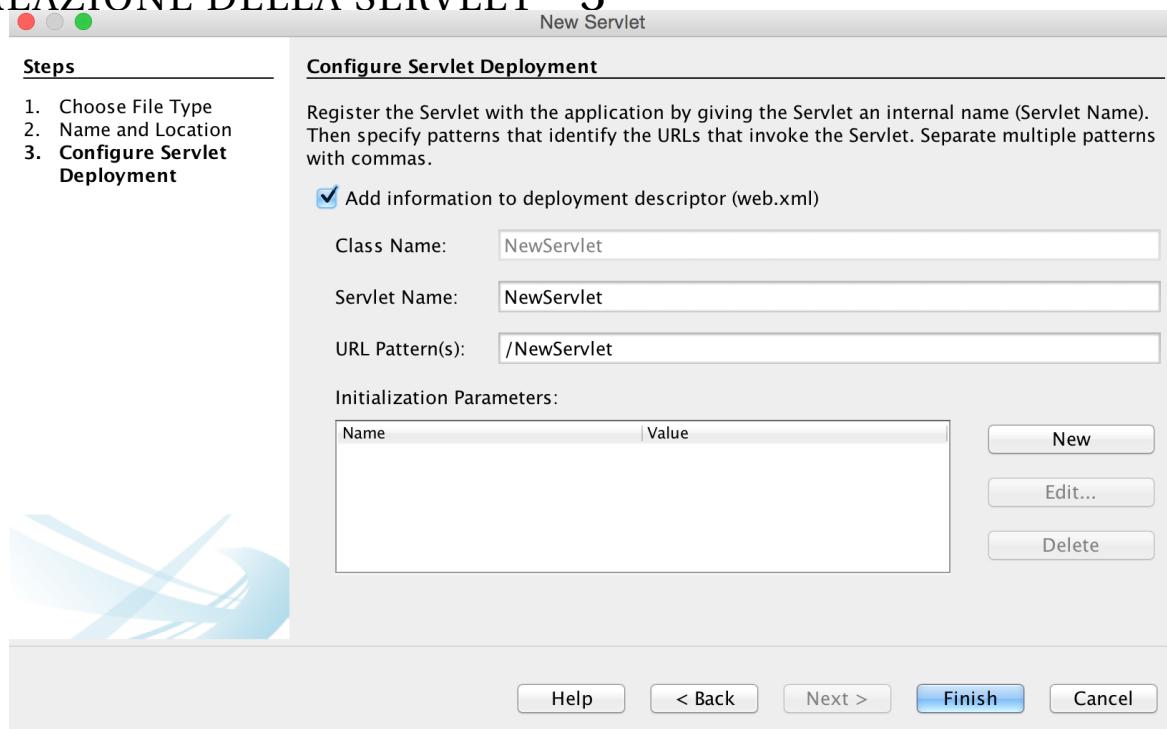
The screenshot shows the NetBeans IDE interface with a title bar "New Servlet". On the left, there's a sidebar titled "Steps" with three items: "Choose File Type", "Name and Location", and "Configure Servlet Deployment". The main area is titled "Name and Location" and contains fields for "Class Name" (set to "NewServlet"), "Project" (set to "WebApplication1"), "Location" (set to "Source Packages"), and "Package" (empty). Below these, the "Created File" path is shown as "/Users/vitsca/NetBeansProjects/WebApplication1/src/java/NewServlet.java". At the bottom, a warning message reads "Warning: It is highly recommended that you do not place Java classes in the default package." with an exclamation mark icon. Navigation buttons at the bottom include "Help", "< Back", "Next >", "Finish", and "Cancel".



50/60

Gli esempi

CREAZIONE DELLA SERVLET - 3



51/60

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

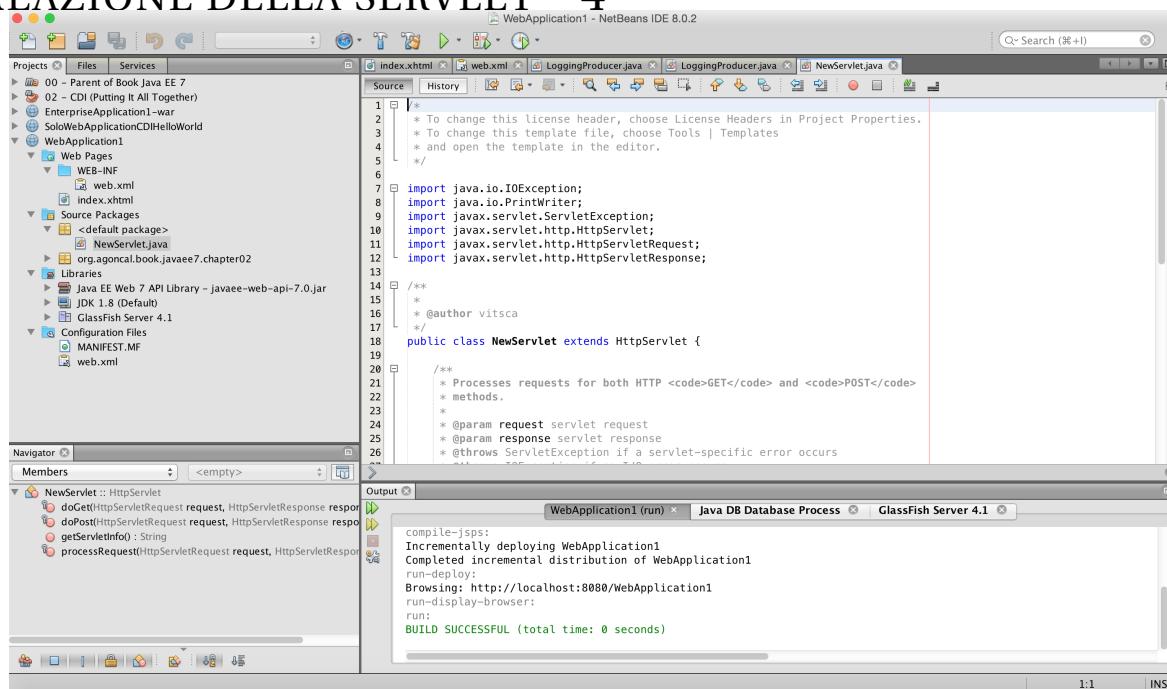
Maven

GlassFish

Gli esempi

Conclusioni

CREAZIONE DELLA SERVLET - 4



52/60

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

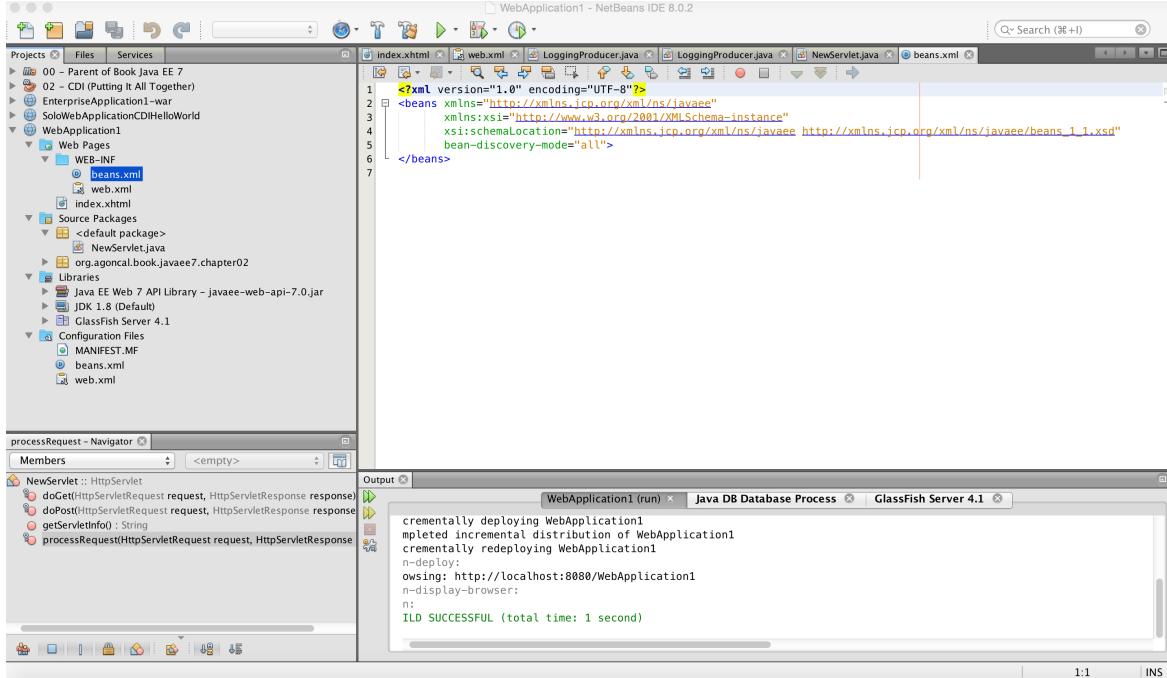
Maven

GlassFish

Gli esempi

Conclusioni

FILE BEANS.XML



53/60

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni

COME È FATTA LA SERVLET - 1

```

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException; ←
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class NewServlet extends HttpServlet {
    protected void processRequest(HttpServletRequest
        request, ←
        HttpServletResponse response) ←
        throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) { ←
        /* TODO output your page here. */
        out.println("<!DOCTYPE html>"); ←
        out.println("<html>");
        out.println("  <head>"); ←
        out.println("    <title>Servlet NewServlet</title>"); ←
        out.println("  </head>"); ←
        out.println("  <body>"); ←
        out.println("    <h1>Servlet NewServlet at " + ←
                    request.getContextPath() + "</h1>"); ←
        out.println("  </body>"); ←
        out.println("  </html>"); ←
    }
}
// ...

```

Import

Estende classe astratta che fornisce metodi già implementati

Metodo interno, unico per gestire sia GET che POST

Parametro risposta

Stream bufferizzato di output

Si scrive il file HTML ...

... fino alla fine

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



54/60

COME È FATTA LA SERVLET - 2

```
// ...
@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);
}
```

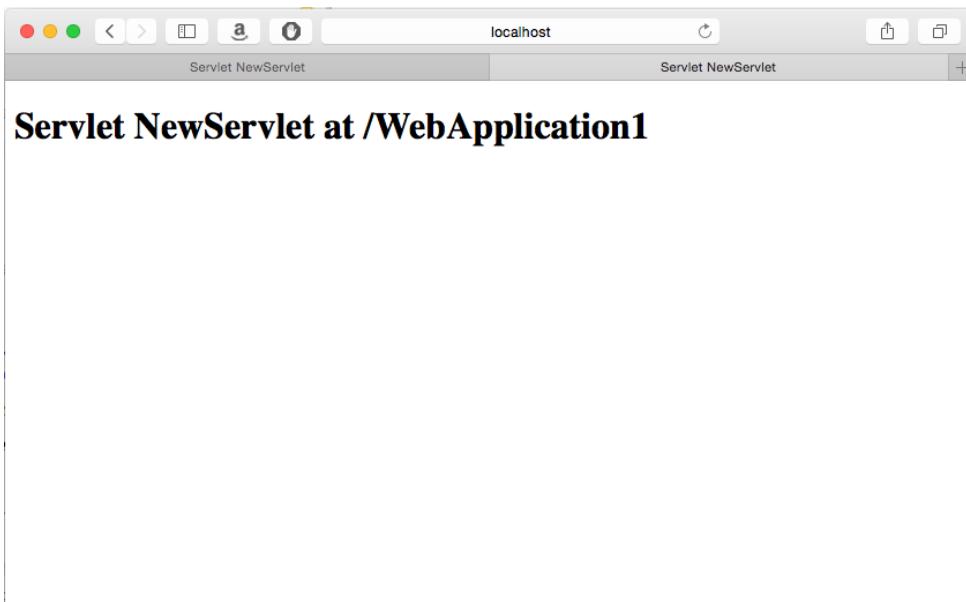
Metodo per trattare GET
... con i parametri
... passati a metodo interno unico
Stesso per POST

Introduzione
Mettiamo tutto insieme
Netbeans
Primo approccio
Maven
GlassFish
Gli esempi
Conclusioni



55/60

LA RISPOSTA DELLA SERVLET



Introduzione
Mettiamo tutto insieme
Netbeans
Primo approccio
Maven
GlassFish
Gli esempi
Conclusioni



56/60

METTIAMO INSIEME IL TUTTO!

```
// ... altri import
import org.agoncal.book.javaee7.chapter02.*;
public class NewServlet extends HttpServlet {
    @Inject
    BookService b;
    protected void processRequest(HttpServletRequest request,
                                   HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html; charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet NewServlet</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet NewServlet at " +
                       request.getContextPath() + "</h1>");
            Book book = b.createBook("H2G2", 12.5f, "Geeky scifi
Book");
            out.println("<h3>Libro creato:" + book + "</h3>");
            out.println("</body>");
            out.println("</html>");
        }
    }
// ...
}
```

Import classi esempio

Servlet

Risorse iniettata dal container

Inizio output

Creiamo un libro dal servizio offerto dal container

Lo stampiamo a video

Fine dell'output

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

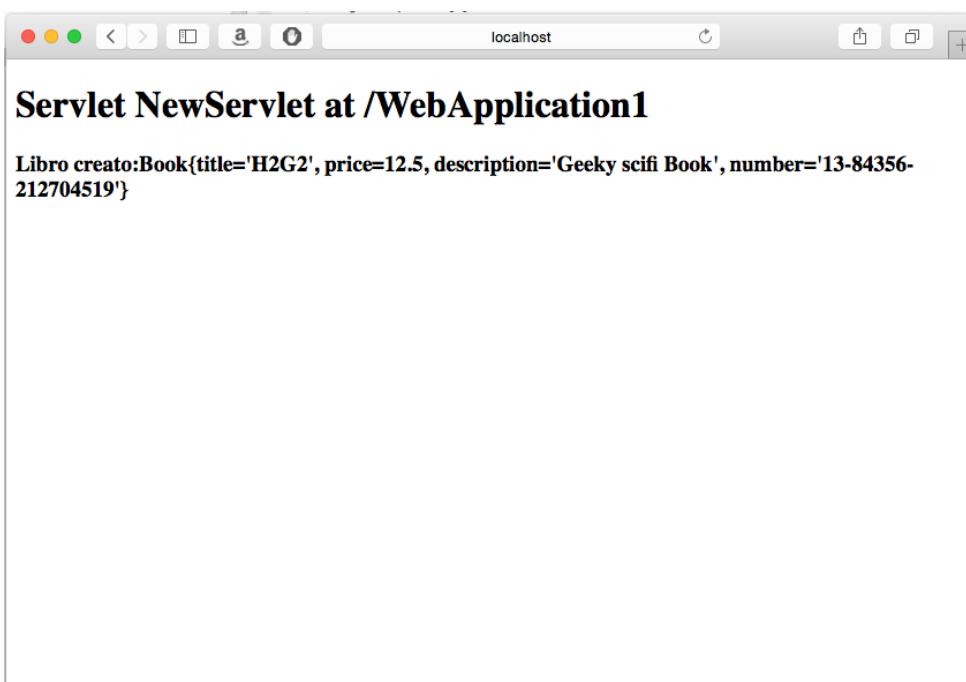
Gli esempi

Conclusioni



57/60

IL RISULTATO



Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



58/60

PLAN

14: JEE Lab 1

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



59/60

QUELLO CHE ABBIAMO FATTO...

14: JEE Lab 1

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni

Introduzione

Mettiamo tutto insieme

Netbeans

Primo approccio

Maven

GlassFish

Gli esempi

Conclusioni



60/60