

# Laboratorio di “Sistemi Distribuiti”

## A.A. 2024-2025



## Servlet

Emanuele Petriglia

Slide aggiornate al 15 Maggio 2025

# Indice dei contenuti

1. Maven
2. Jetty
3. Esercizio A
4. Esercizio B
5. Esercizio C

# 1. Maven

# Cos'è Maven?

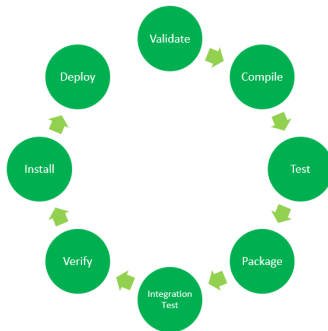
- **Maven** è un software che automatizza la gestione di un progetto Java.



- Perché Maven? Per scrivere ed eseguire delle Servlet è necessario:
  - ① Scaricare le dipendenze necessarie,
  - ② Compilare le classi Java,
  - ③ Impacchettare le classi compilate in un archivio WAR.
- Maven **automatizza**:
  - La **gestione delle dipendenze**,
  - La **costruzione (build) del progetto**.

# Concetti fondamentali

- **Project Object Model (POM):** è un file XML (`pom.xml`) che contiene tutte le informazioni del progetto (nome, elenco delle dipendenze, struttura...).
- **Fasi del ciclo di vita:** sono passi predefiniti eseguiti da Maven. Per esempio i passi del ciclo default:



Fonte

# “Hello World” con Maven (pt. 1)

- ❶ Prerequisiti: JDK e Maven installato (già fatto sulla VM).
- ❷ Aprire VS Code in una cartella vuota, aprire il terminale ed eseguire `mvn --version`.
- ❸ Creare un progetto di esempio con:  

```
> mvn archetype:generate "-DgroupId=it.unimib.sd2025"  
↪ "-DartifactId=hello-world"  
↪ "-DarchetypeArtifactId=maven-archetype-quickstart"  
↪ "-DarchetypeVersion=1.4" "-DinteractiveMode=false"
```

Attenzione agli spazi!

Tra un argomento e l'altro ci deve essere almeno un carattere di spazio!



# “Hello World” con Maven (pt. 2)

## Output del comando:

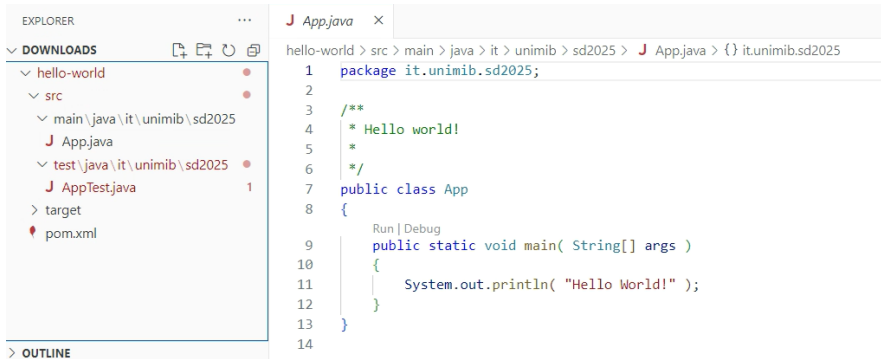
```
PS C:\Users\EP083473\Downloads> mvn archetype:generate "-DgroupId=it.unimib.sd2025" "-DartifactId=hello-world" "-DarchetypeArtifactId=maven-archetype-quickstart" "-DarchetypeVersion=1.4" "-DinteractiveMode=false"
[INFO] Scanning for projects...
[INFO] -----< org.apache.maven:standalone-pom >-----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----[ pom ]-----
[INFO]
[INFO] >>> archetype:3.3.1:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
[INFO] <<< archetype:3.3.1:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO] --- archetype:3.3.1:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[INFO] -----
[INFO] Using following parameters for creating project from archetype: maven-archetype-quickstart:1.4
[INFO] -----
[INFO] Parameter: groupId, Value: it.unimib.sd2025
[INFO] Parameter: artifactId, Value: hello-world
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: it.unimib.sd2025
[INFO] Parameter: packageInPathFormat, Value: it/unimib/sd2025
[INFO] Parameter: package, Value: it.unimib.sd2025
[INFO] Parameter: groupId, Value: it.unimib.sd2025
[INFO] Parameter: artifactId, Value: hello-world
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Project created from Archetype in dir: C:\Users\EP083473\Downloads\hello-world
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.755 s
[INFO] Finished at: 2025-05-14T14:16:47+02:00
[INFO] -----
PS C:\Users\EP083473\Downloads> 
```

T



# "Hello World" con Maven (pt. 3)

Struttura del progetto:



Sono necessarie due modifiche nel file `pom.xml`:

- Sostituire le righe 17 e 18 con  
`<maven.compiler.release>21</maven.compiler.release>`,
- Eliminare tutte le righe nella sezione `<build>`.





# "Hello World" con Maven (pt. 4)

Il pom.xml risultante:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0"
   ↳ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
   ↳ http://maven.apache.org/xsd/maven-4.0.0.xsd">
4    <modelVersion>4.0.0</modelVersion>
5
6    <groupId>it.unimib.sd2025</groupId>
7    <artifactId>hello-world</artifactId>
8    <version>1.0-SNAPSHOT</version>
9
10   <name>hello-world</name>
11   <!-- FIXME change it to the project's website -->
12   <url>http://www.example.com</url>
13
14   <properties>
15     <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
16     <maven.compiler.release>21</maven.compiler.release>
17   </properties>
18
19   <dependencies>
20     <dependency>
21       <groupId>junit</groupId>
22       <artifactId>junit</artifactId>
23       <version>4.11</version>
24       <scope>test</scope>
25     </dependency>
26   </dependencies>
27 </project>
```



## “Hello World” con Maven (pt. 4)

- Quando si invoca Maven si specifica la fase da raggiungere o un goal da svolgere. Per esempio:
  - `compile`: compila i sorgenti Java,
  - `clean`: elimina tutti i file generati (classi compilate, file compressi...).
  - `package`: crea il file compresso JAR.
- Eseguire, uno alla volta, i seguenti comandi Maven e osservare i log di output e come cambia il contenuto di target:
  - ① `mvn compile`
  - ② `mvn test`
  - ③ `mvn package`
  - ④ `mvn clean`
  - ⑤ `mvn package`

### Automazione di Maven

Niente più compilazioni a mano con `javac` o invocazioni di `java` su un singolo file Java!



# “Hello World” con Maven (pt. 5)

- L'ultimo comando produce l'archivio JAR `hello-world-1.0-SNAPSHOT.jar` nella cartella `target`.
- Eseguire la classe principale dell'archivio con:  

```
> java -cp .\target\hello-world-1.0-SNAPSHOT.jar  
↪ it.unimib.sd2025.App
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ hello-world ---
[INFO] skip non existing resourceDirectory C:\Users\EP083473\Downloads\hello-world\src\test\resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ hello-world ---
[INFO] Recompiling the module because of changed dependency.
[INFO] Compiling 1 source file with javac [debug release 21] to target\test-classes
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ hello-world ---
[INFO] Using auto detected provider org.apache.maven.surefire.junit4.JUnit4Provider
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running it.unimib.sd2025.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.049 s -- in it.unimib.sd2025.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ hello-world ---
[INFO] Building jar: C:\Users\EP083473\Downloads\hello-world\target\hello-world-1.0-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.538 s
[INFO] Finished at: 2025-05-14T14:48:06+02:00
[INFO] -----
PS C:\Users\EP083473\Downloads\hello-world> java -cp .\target\hello-world-1.0-SNAPSHOT.jar it.unimib.sd2025.App
Hello World!
PS C:\Users\EP083473\Downloads\hello-world> 
```



## 2. Jetty

# Cos'è Jetty?

- [Jetty](#) è un Web server Java e contenitore servlet moderno, simile a Apache Tomcat, WildFly o GlassFish.



- Perché Jetty? Perché è facile installare e avviare, inoltre useremo un plugin per Maven per automatizzare il deploy delle Servlet negli esercizi.
- Useremo le API [JakartaEE](#) 10 (ex. JavaEE) per scrivere le Servlet e le eseguiremo con Jetty 12.

## 2. Jetty

Deploy manuale delle Servlet

# Esempio di Servlet

- 1 Scaricare da e-Learning la cartella `hello-world.zip` e decomprimerla.
- 2 Aprire la cartella `hello-world` con Visual Studio Code.
- 3 Aprire un terminale ed eseguire il seguente comando:  

```
> mvn package
```

Maven costruirà il progetto e in `target` sarà presente il file `hello-world-0.1.war`.

## Attenzione

Il primo avvio del comando Maven impiegherà più tempo perché scaricherà le dipendenze.

## Formato WAR

Un file **WAR** (Web Application Archive) è un archivio usato per distribuire applicazioni Web Java su server Web Java e contenitori servlet, proprio come Jetty!

# Installazione di Jetty (pt. 1)

- 1 Scaricare dalla [pagina download di Jetty](#) la versione 12.0.21 come ZIP ed estrarre l'archivio in Downloads.

La cartella `jetty-home-12.0.21` contiene i file necessari per avviare Jetty. **Non verrà mai modificata.**

- 2 Aprire Windows PowerShell con   :

```
> cd Downloads
> mkdir jetty-base
> cd jetty-base
> java -jar ..\jetty-home-12.0.21\start.jar
↪ --add-modules=server,http,ee10-deploy,ee10-jsp
```

L'ultimo comando inizializza la configurazione di Jetty in `jetty-base`.



## Installazione di Jetty (pt. 2)

- 3 Modificare il file `jetty-base/start.d/ee10-deploy.ini` sostituendo la riga 14 con:

```
jetty.deploy.scanInterval=1
```

Ciò permette di aggiungere e rimuovere le servlet dinamicamente mentre Jetty è in esecuzione.

### Attenzione!

Controllare bene di rimuovere il carattere `#` all'inizio della riga 14!

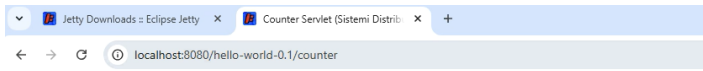
- 4 Dalla cartella `jetty-base`, avviare Jetty con il seguente comando:  
> `java -jar ..\jetty-home-12.0.21\start.jar`

Rimarrà in esecuzione finché non lo si interrompe con `CTRL + C`.



# Deploy della Webapp

- 1 Copiare il file `hello-world-0.1.war` nella cartella `webapps` in `jetty-base`.
- 2 Aprire il browser e raggiungere l'indirizzo <http://localhost:8080/hello-world-0.1/counter>. La servlet verrà creata ed eseguita:



## Counter Servlet (Sistemi Distribuiti 2024-2025)

This is a counter: 8

[Reload](#) to increment the counter.

- 3 Rimuovere il file WAR da `webapps` e notare cosa cambia al ricaricare della pagina Web.



## 2. Jetty

Deploy automatico con Maven

# Perché con Maven?

- **Problema:** durante lo sviluppo delle servlet risulta scomodo spostare manualmente il file WAR e gestire l'avvio e configurazione di Jetty.
  - **Soluzione:** [Jetty Maven Plugin](#)
    - È un plugin per Maven che permette di avviare in automatico un'istanza Jetty e di caricare tutte le servlet del progetto.
- ① Aprire la cartella `hello-world` con Visual Studio Code.
  - ② Aprire un terminale ed eseguire:

```
> mvn jetty:run
```

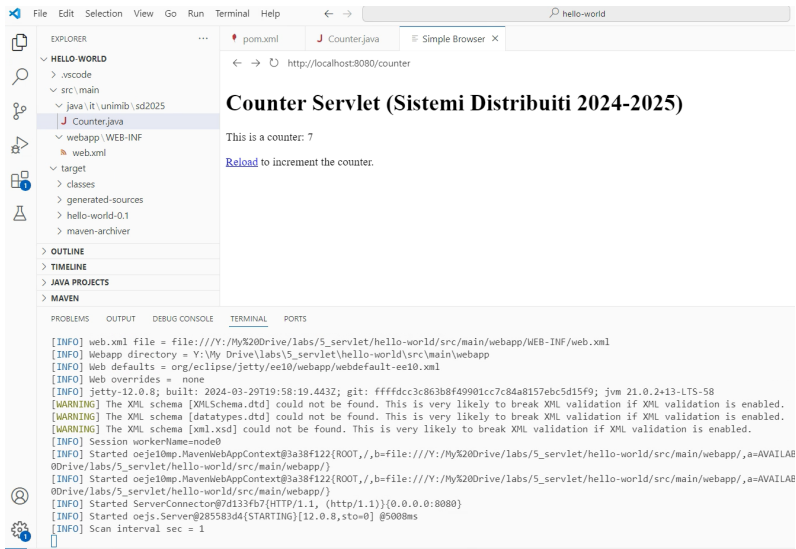
Questo goal compila il codice, avvia Jetty e carica le servlet. Jetty rimane in esecuzione finché non si non lo si interrompe con CTRL + C.

## Browser interno di VS Code

Si accede con `Ctrl` `Shift` `p`, scrivere Simple Browser, premere `Invio` e inserire l'URL <http://localhost:8080/counter>.



# Browser interno in Visual Studio Code



The screenshot displays the Visual Studio Code interface with a web browser integrated into the editor. The browser shows the URL `http://localhost:8080/counter` and displays the title **Counter Servlet (Sistemi Distribuiti 2024-2025)**. The page content includes the text "This is a counter: 7" and a link labeled "Reload" to increment the counter.

The Explorer sidebar on the left shows the project structure for **HELLO-WORLD**, with the following files and folders:

- `.vscode`
- `src/main`
  - `java\it\unimib\sd2025`
    - Counter.java** (selected)
  - `webapp\WEB-INF`
    - `web.xml`
  - `target`
    - `classes`
    - `generated-sources`
    - `hello-world-0.1`
    - `maven-archiver`
- OUTLINE**
- TIMELINE**
- JAVA PROJECTS**
- MAVEN**

The Terminal at the bottom shows the output of the Maven build process, including the following log entries:

```
[INFO] web.xml file = file:///Y:/My%20Drive/labs/5_servlet/hello-world/src/main/webapp/WEB-INF/web.xml
[INFO] Webapp directory = Y:/My Drive/labs/5_servlet/hello-world/src/main/webapp
[INFO] Web defaults = org.eclipse/jetty/ee10/webapp/webdefault-ee10.xml
[INFO] Web overrides = none
[INFO] jetty-12.0.8; built: 2024-03-29T19:58:19.443Z; git: fffffdccc3863b8f49901cc7c84a8157ebc5d15f9; jvm 21.0.2+13-LTS-58
[WARNING] The XML schema [XLSchema.dtd] could not be found. This is very likely to break XML validation if XML validation is enabled.
[WARNING] The XML schema [datatypes.dtd] could not be found. This is very likely to break XML validation if XML validation is enabled.
[WARNING] The XML schema [xml.xsd] could not be found. This is very likely to break XML validation if XML validation is enabled.
[INFO] Session workerName=node0
[INFO] Started oeje10mp.MavenWebAppContext@3a38f122[ROOT,/,b=file:///Y:/My%20Drive/labs/5_servlet/hello-world/src/main/webapp/,a=AVAILABLE/labs/5_servlet/hello-world/src/main/webapp/]
[INFO] Started oeje10mp.MavenWebAppContext@3a38f122[ROOT,/,b=file:///Y:/My%20Drive/labs/5_servlet/hello-world/src/main/webapp/,a=AVAILABLE/labs/5_servlet/hello-world/src/main/webapp/]
[INFO] Started ServerConnector@7d133fb7[HTTP/1.1, (http/1.1)]{0.0.0.0:8080}
[INFO] Started oejs.Server@285583d4[STARTING][12.0.8,sto=0] @5008ms
[INFO] Scan interval sec = 1
```



### 3. Esercizio A

# Consegna

- Implementare la servlet Java Keyboards che risponde alla chiamata GET /keyboards restituendo una pagina HTML con una tabella che elenca le informazioni sulle tastiere salvate nel sistema.

The screenshot displays a web browser window with the address bar showing `localhost:8080/keyboards`. The page title is "Keyboard Manager Panel". The main content is a table with the following data:

Name	Manufacturer	Year	Ergonomic	Backlight	Layout
K55 RGB PRO	Corsair	2023	true	true	QWERTY
K30	Corsair	2020	false	false	QWERTY
G213	Logitech	2023	false	true	Dvorak

Below the table, the browser's developer tools are open to the "Sources" panel, showing the source code of the `keyboards X` file. The code is an HTML document with the following structure:

```
<!DOCTYPE html>
<html>
<head>
<title>Keyboard Manager Panel</title>
</head>
<body>
<h1>Keyboard Manager Panel</h1>
<table>
<thead>
<tr>
<th>Name</th>
<th>Manufacturer</th>
<th>Year</th>
<th>Ergonomic</th>
<th>Backlight</th>
<th>Layout</th>
</tr>
</thead>
<tbody>
<tr>
<td>K55 RGB PRO</td>
<td>Corsair</td>
<td>2023</td>
<td>true</td>
<td>true</td>
<td>QWERTY</td>
</tr>
<tr>
<td>K30</td>
<td>Corsair</td>
<td>2020</td>
<td>false</td>
<td>false</td>
<td>QWERTY</td>
</tr>
<tr>
<td>G213</td>
<td>Logitech</td>
<td>2023</td>
<td>false</td>
<td>true</td>
<td>Dvorak</td>
</tr>
</tbody>
</table>
</body>
</html>
```



# Suggerimenti

- Partire dallo scheletro. Contiene la classe Keyboard e lo scheletro della servlet KeyboardsServlet.
- Come scrivere codice HTML?
  - Nei laboratori va bene usare un `StringBuffer` e scrivere manualmente codice HTML. Nel mondo reale si usano framework e librerie più avanzate.
  - Vedere lo scheletro per il codice di esempio.
- Stringhe su più righe in Java: si può da Java 15. Esempio:

```
1  var i = 5;
2  var str = """
3      <p>
4          Contatore con valore %s
5      </p>""".formatted(i);
```

- In caso di dubbi c'è la [documentazione](#) delle annotazioni, classi e metodi usati per scrivere le servlet.





## 4. Esercizio B

## Consegna (pt. 1)

Estendere l'esercizio precedente:

- Aggiungere due proprietà alla classe Keyboard:
  - `id`: un intero che identifica la singola tastiera,
  - `comments`: commenti (stringhe) sulla tastiera.
- Nella tabella che restituisce GET `/keyboards`, oltre a mostrare l'ID numerico, per la colonna dei commenti c'è un link che porta all'indirizzo GET `/keyboard?id=<id>`.
- Se alla chiamata GET `/keyboards` c'è il parametro `id=<id>`, restituisce una pagina HTML che mostra soltanto le informazioni della tastiera con l'ID fornito.



# Consegna (pt. 2)

## Pagina principale:

The screenshot displays a web browser window with the title 'Keyboard Manager Panel' and the address bar showing 'localhost:8080/keyboards'. The main content area features a table titled 'Keyboard Manager Panel' with the following data:

ID	Name	Manufacturer	Year	Ergonomic	Backlight	Layout	Comments
0	K55 RGB PRO	Corsair	2023	true	true	QWERTY	<a href="#">Click</a>
1	K30	Corsair	2020	false	false	QWERTY	<a href="#">Click</a>
2	G213	Logitech	2023	false	true	Dvorak	<a href="#">Click</a>

On the right side of the browser, the 'Sources' panel is open, showing the HTML source code of the page. The code includes a table structure with columns for ID, Name, Manufacturer, Year, Ergonomic, Backlight, Layout, and Comments. The table body contains three rows of data corresponding to the table shown in the main content area. The source code is as follows:

```
<!DOCTYPE html>
<html>
<head>
  <title>Keyboard Manager Panel</title>
</head>
<body>
  <h1>Keyboard Manager Panel</h1><table>
    <thead>
      <tr>
        <th>ID</th>
        <th>Name</th>
        <th>Manufacturer</th>
        <th>Year</th>
        <th>Ergonomic</th>
        <th>Backlight</th>
        <th>Layout</th>
        <th>Comments</th>
      </tr>
    </thead>
    <tbody><tr>
      <td>0</td>
      <td>K55 RGB PRO</td>
      <td>Corsair</td>
      <td>2023</td>
      <td>true</td>
      <td>true</td>
      <td>QWERTY</td>
      <td><a href="/keyboards?id=0">Click</a></td>
    </tr><tr>
      <td>1</td>
      <td>K30</td>
      <td>Corsair</td>
      <td>2020</td>
      <td>false</td>
      <td>false</td>
      <td>QWERTY</td>
      <td><a href="/keyboards?id=1">Click</a></td>
    </tr><tr>
      <td>2</td>
      <td>G213</td>
      <td>Logitech</td>
      <td>2023</td>
      <td>false</td>
      <td>true</td>
      <td>Dvorak</td>
      <td><a href="/keyboards?id=2">Click</a></td>
    </tr>
  </tbody>
</table>
</body>
</html>
```

Coverage: n/a



# Consegna (pt. 2)

Pagina specifica di una tastiera:

The screenshot shows a web browser window with a single tab titled 'Keyboard 2'. The address bar shows the URL 'localhost:8080/keyboards?id=2'. The page content includes a heading 'Keyboard 2' followed by a bulleted list of specifications: Name: G213, Manufacturer: Logitech, Year: 2023, Ergonomic: false, Backlight: true, Layout: Dvorak, and Comments: Meglio di Corsair, Funzione molto bene, consigliata!. Below the list is a link labeled 'Go back'. The browser's developer tools are open to the 'Sources' panel, displaying the HTML source code for the file 'keyboards?id=2 X'. The code is a standard HTML document with a title 'Keyboard 2' and a body containing a heading and a list of items with their details.

**Keyboard 2**

- **Name:** G213
- **Manufacturer:** Logitech
- **Year:** 2023
- **Ergonomic:** false
- **Backlight:** true
- **Layout:** Dvorak
- **Comments:**
  - Meglio di Corsair
  - Funzione molto bene, consigliata!

[Go back](#)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Keyboard 2</title>
5 </head>
6 <body>
7   <h1>Keyboard 2</h1><ul>
8     <li><b>Name:</b> G213</li>
9     <li><b>Manufacturer:</b> Logitech</li>
10    <li><b>Year:</b> 2023</li>
11    <li><b>Ergonomic:</b> false</li>
12    <li><b>Backlight:</b> true</li>
13    <li><b>Layout:</b> Dvorak</li>
14    <li><b>Comments:</b>
15      <ul><li>Meglio di Corsair</li><li>Funzione molto bene, consigliata!</li></ul>
```



# Suggerimenti

- Per andare alla pagina della singola tastiera, nella colonna dei commenti bisogna aggiungere un collegamento HTML:

```
<a href="/keyboards?id=5">Click</a>
```

- Per ottenere un parametro dall'URL:

```
1 // http://localhost:8080/pagina?X=qualcosa
2 // rawX può essere null se X non è stata impostata
3 String rawX = request.getParameter("X");
```

- Nel metodo che gestisce la chiamata GET /keyboards, separare i due casi (uno con ID e uno senza) creando due metodi specifici per ognuno.
- Cosa fare in caso di errore dell'utente? Restituire un errore:

```
1 var id = 0;
2 try {
3     id = Integer.parseInt(rawId);
4 } catch (NumberFormatException e) {
5     response.setStatus(HttpServletResponse.SC_BAD_REQUEST);
6     response.getWriter().write("400 Bad Request: Invalid ID.");
7     return;
8 }
```



## 5. Esercizio C

# Consegna (pt. 1)

Estendere l'esercizio precedente:

- Alla chiamata GET /keyboards, oltre a restituire la tabella con le tastiere, mostrare un form per aggiungere una nuova tastiera.
- Modificare la servlet aggiungendo la gestione della chiamata POST /keyboards: aggiunge una nuova tastiera nella lista.



# Consegna (pt. 2)

## Pagina principale:

The screenshot shows a web browser displaying the 'Keyboard Manager Panel' application. The application has a table of keyboard configurations and a form to add a new one. The table lists three keyboards: K55 RGB PRO Corsair, K30 Corsair, and G213 Logitech. The form includes fields for Name, Manufacturer, Year, Ergonomic, Backlight, and Layout, with a Submit button.

ID	Name	Manufacturer	Year	Ergonomic	Backlight	Layout	Comments
0	K55 RGB PRO	Corsair	2023	true	true	QWERTY	<a href="#">Click</a>
1	K30	Corsair	2020	false	false	QWERTY	<a href="#">Click</a>
2	G213	Logitech	2023	false	true	Dvorak	<a href="#">Click</a>

Name:

Manufacturer:

Year:

Ergonomic: ☐

Backlight: ☒

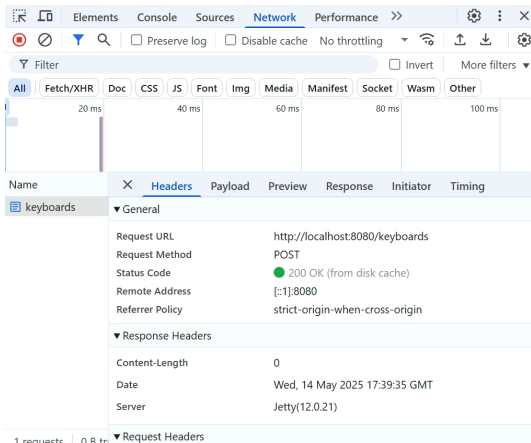
Layout:

The browser's developer tools show a network request to 'localhost:8080/keyboards'. The request is a GET method with a status code of 200 OK. The response headers indicate a content length of 2102 and a content type of 'text/html; charset=utf-8'.



# Consegna (pt. 3)

Invio del modulo HTML:



The screenshot shows the Chrome DevTools Network tab. The 'Network' tab is selected, and the 'Fetch/XHR' filter is applied. A single request is visible, which has been expanded to show its details. The request is a POST to 'http://localhost:8080/keyboards' with a status of '200 OK (from disk cache)'. The response headers show 'Content-Length: 0', 'Date: Wed, 14 May 2025 17:39:35 GMT', and 'Server: Jetty(12.0.21)'. The request headers section is also visible at the bottom.

Name	Value
Request URL	http://localhost:8080/keyboards
Request Method	POST
Status Code	200 OK (from disk cache)
Remote Address	::1:8080
Referrer Policy	strict-origin-when-cross-origin
▼ Response Headers	
Content-Length	0
Date	Wed, 14 May 2025 17:39:35 GMT
Server	Jetty(12.0.21)
▼ Request Headers	

# Consegna (pt. 4)

Pagina principale dopo l'invio del modulo:



## Keyboard Manager Panel

ID	Name	Manufacturer	Year	Ergonomic	Backlight	Layout	Comments
0	K55 RGB PRO	Corsair	2023	true	true	QWERTY	<a href="#">Click</a>
1	K30	Corsair	2020	false	false	QWERTY	<a href="#">Click</a>
2	G213	Logitech	2023	false	true	Dvorak	<a href="#">Click</a>
3	GXT 835	Trust	2024	false	false	QWERTY	<a href="#">Click</a>

Name:

Manufacturer:

Year:

Ergonomic:

☐

Backlight:

☒

Layout:



# Suggerimenti

- Per gestire una chiamata POST bisogna aggiungere il metodo:

```
1  @Override
2  protected void doPost(HttpServletRequest request, HttpServletResponse
   ↪ response)
3      throws ServletException, IOException {
4      // ...
5  }
```

- Generare l'ID della nuova tastiera lato server, e attenzione all'accesso concorrente alla lista!
- Per il form HTML, si può leggere la documentazione su [MDN Web Docs](#). Esempio:

```
1  <form action="/keyboards" method="POST">
2      <p>
3          <label for="name">Name:</label>
4          <br>
5          <input type="text" id="name" name="name" value="" required />
6          </p>
7
8          <p>
9              <input type="submit" value="Submit" />
10             </p>
11 </form>
```

