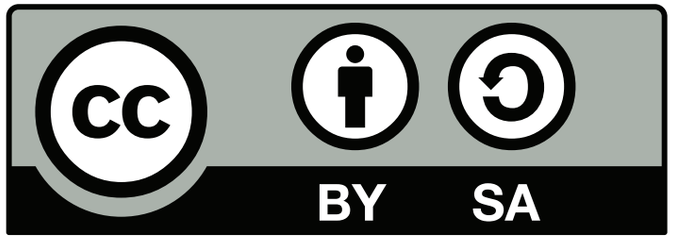
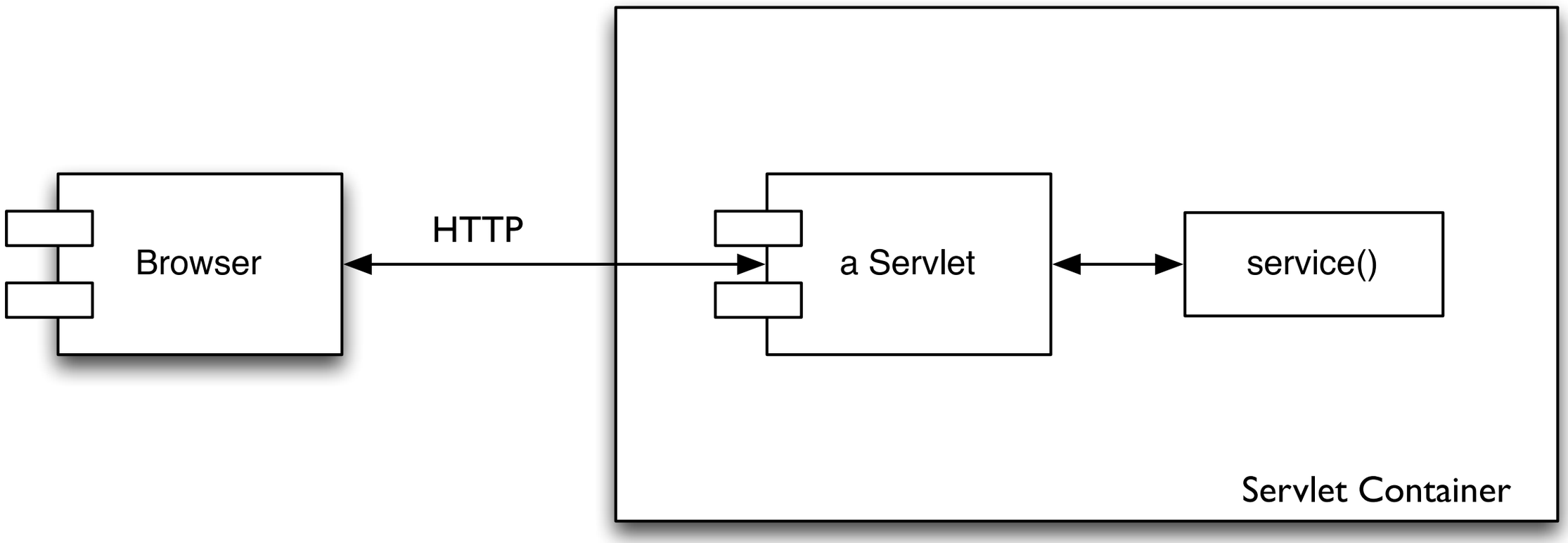
Applicazioni Web 2013/14

Lezione 6 - Servlet API

Matteo Vaccari
http://matteo.vaccari.name/
matteo.vaccari@uninsubria.it



Servlets and container



Simple servlet

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {

  public void service(HttpServletRequest request, HttpServletResponse response)
                                  throws ServletException, IOException {

    String name = request.getParameter("name");
    PrintWriter out = response.getWriter();

    out.println("<html>");
    out.println("<head><title>Hello world</title></head>");
    out.println("<body>");
    out.println("<big>Hello, " + name + "</big>");
    out.println("</body></html>");
  }
}
```

Project structure (JEE container)

```
.
|-- src
|   `-- com
|       `-- example
|           `-- foobar
|               `-- FoobarServlet.java
`-- webapp
    |-- WEB-INF
    |   |-- classes
    |   |   `-- com
    |   |       `-- example
    |   |           `-- foobar
    |   |               `-- FoobarServlet.class
    |   `-- web.xml
    |-- images
    |   `-- logo.png
    |-- index.html
    |-- javascripts
    |   `-- prototype.js
    `-- stylesheets
        `-- style.css
```

Project structure (main program)

```
hangman
├── script
├── src
|   ├── main
|   |   ├── java
|   |   |   └── it
|   |   |       └── xpug
|   |   |           └── hangman
|   |   |               ├── domain
|   |   |               ├── main
|   |   |               └── web
|   |   └── webapp
|   └── test
|       └── java
└── target
```

web.xml

```
<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
    "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">

<web-app>
    <servlet>
        <servlet-name>hello-world</servlet-name>
        <servlet-class>com.example.foobar.HelloWorld</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>hello-world</servlet-name>
        <url-pattern>/hello/*</url-pattern>
    </servlet-mapping>
</web-app>
```

```
/           equivale a /*
/hello      path esatto
/hello/*    tutti i path con quel prefisso
*.png       tutti i path con quella estensione
```

Le servlet sono *singleton*

- Sono accedute da più thread contemporaneamente
- Evitare dunque le variabili di istanza

HttpServletRequest

http://www.example.com/context/hello/world

```
String getParameter(String)
Enumeration getParameterNames()

String getScheme()         //=> http
String getServerName()     //=> www.example.com
getServerPort()            //=> 80
getRequestURI()            //=> /context/hello/world
getContextPath()           //=> /context
getServletPath()           //=> /hello
getPathInfo()              //=> /world

String getRemoteAddr()
String getRemoteHost()

Enumeration getHeaderNames()
String getHeader(String name)

Cookie[] getCookies()
```

HttpServletResponse

```
void setContentType(String type)
void setContentLength(int length)
ServletOutputStream getOutputStream() // output binario
PrintWriter getWriter()               // output testo

void setHeader(String name, String val)
void addCookie(Cookie cookie)

void sendRedirect(String url)
void setStatus(int code)
```

Maven: *build* and *package*

```
<project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/
  <modelVersion>4.0.0</modelVersion>
  <groupId>it.xpug</groupId>
  <artifactId>hangman</artifactId>
  <version>0.1-SNAPSHOT</version>

  <dependencies>
    <dependency>
      <groupId>org.mortbay.jetty</groupId>
      <artifactId>jetty-util</artifactId>
      <version>6.1.26</version>
    </dependency>

    <dependency>
      <groupId>org.mortbay.jetty</groupId>
      <artifactId>jetty</artifactId>
      <version>6.1.26</version>
    </dependency>
  </dependencies>
</project>
```

pom.xml

Using Maven

mvn clean

mvn package

mvn eclipse:eclipse

mvn compile

mvn test