Jakarta EE

Leonardo De Boni

December 6, 2022

CONTENTS

1 INTRODUZIONE

Il package che ci interessa é: java.net. Le classi che ci interessano sono:

- InetAddress: rappresenta un indirizzo ip
- Socket: permette di creare un client
- ServerSocket: permette di creare un Server

1.1 Ottenere un indirizzo ip

Per ottenere l'indirizzo ip a partire da un nome possiamo fare cosí:

```
InetAddress address = InetAddress.getByName(String hostName);
```

1.2 Connettersi al server

Per connettersi a un server dopo aver ottenuto l'indirizzo possiamo creare un client cosí:

```
Socket socket = new Socket(InetAddress adress, int port)
```

Dal socket possiamo ottenere un InputStream:

```
InputStream in = socket.getInputStream();
```

2 SERVLET

Classe che mi permette di elaborare una richiesta. Per crearne uno devo estendere la classe HttpServlet. I metodi che posso sovrascrivere che corrispondono alle varie richieste sono i seguenti:

- doGet (HttpServletRequest req, HttpServletResponse res)
- doPost (HttpServletRequest req, HttpServletResponse res)
- doHead (HttpServletRequest req, HttpServletResponse res)
- doOptions (HttpServletRequest req, HttpServletResponse res)
- doTrace (HttpServletRequest req, HttpServletResponse res)
- doPut (HttpServletRequest req, HttpServletResponse res)
- doDelete (HttpServletRequest req, HttpServletResponse res)

2.1 Servlet annotations

3 STRUTTURA DI UN PROGETTO WEB

```
/src/main
├── java/com/example
│   ├── ExampleServlet.java
│   ├── ExampleFilter.java
│   └── ...
├── resources
│   ├── example.pdf
│   ├── example.png
│   └── ...
└── webapp
    ├── WEB-INF
    │   ├── web.xml
    │   └── ...
    ├── index.jsp
    ├── example.jsp
    └── ...
```

4 RESOURCES

5 FILTERS

5.1 Filter annotations

6 JSP (JAKARTA SERVER PAGES)

JSP mi permette di scrivere codice java all'interno di HTML. Per fare ciò posso usare delimitatori dedicati:

- **Scriptlets**: <% ... istruzioni java ... %>
- **Expressions**: <%= ... espressioni java ... %> sará mostrato al client
- **Declarations**: <%! ... espressioni java ... %> posso dichiarare metodi

Nelle pagine JSP ho a disposizione vari oggetti impliciti:

- request HttpServletRequest
- response HttpServletResponse
- out OutputStream (response.getOutputStream())
- page this
- exception Throwable
- config ServletConfig