**1. Browser e Richieste HTTP**

**Concetti Base**

* **Interfacciamento con il browser**  
  Il browser è l’interfaccia principale tra noi e il web, agendo come intermediario tra il client (noi) e le risorse sul server.  
  In passato usavamo il Scanner per input manuale, ora utilizziamo il browser per inviare richieste HTTP e ricevere risposte dai server.
* **Tipi di risorse accessibili tramite il browser**  
  Quando digitiamo un URL, può corrispondere a:
  1. **Risorsa statica**: file HTML, CSS, immagini, serviti direttamente dall’Application Server (AS).
  2. **Risorsa dinamica**: pagina JSP generata lato server dal Servlet Container.
  3. **Risorsa collegata a una classe Java (Servlet)**: una servlet elabora la richiesta e restituisce una risposta dinamica.
* **Gestione URL ambigui**  
  Se il browser non riconosce una risorsa, prova a interpretarla come una pagina HTML per impostazione predefinita.

**2. Application Server (AS) e Servlet Container**

**Concetti Base**

* **Application Server (AS)**  
  Chiamato anche Servlet Container, è responsabile della gestione di risorse dinamiche (JSP, Servlet). Esempi: Apache Tomcat, WildFly, Jetty.
* **Differenza con un Web Server tradizionale**  
  Un Web Server tradizionale (es. Apache HTTPD) non può elaborare JSP o Servlet; richiede un AS che li interpreti e generi una risposta dinamica.
* **Processo generale**
  1. Browser invia una richiesta HTTP.
  2. L’AS interpreta la richiesta, individua la risorsa (statica, JSP o servlet).
  3. L’AS prepara una risposta e la invia al browser.

**3. Processo HTTP e Pacchetti**

**Pacchetti HTTP**

Un pacchetto HTTP è composto da una **richiesta** e una **risposta**, con informazioni organizzate in coppie chiave-valore.

* **Chiavi importanti nella richiesta HTTP**:
  + **Indirizzo IP**: identifica il client.
  + **Accept**: tipi di contenuto accettati (es. text/html, application/json).
  + **Cookie**: usati per tracciare sessioni o personalizzazioni.
* **Flusso di una richiesta HTTP**:
  + Il browser invia una richiesta (es. GET o POST).
  + Il Servlet Container interpreta l’URL e passa la richiesta al gestore appropriato (es. servlet).
  + Il server elabora la richiesta e restituisce una risposta.

**4. Richieste HTTP: GET vs POST**

* **GET**  
  Utilizzato per recuperare dati; invia parametri nell’URL (es. /pagina?parametro=valore). È meno sicuro poiché i dati sono visibili nella barra degli indirizzi.
* **POST**  
  Utilizzato per inviare dati sensibili; i parametri sono inclusi nel corpo della richiesta, rendendolo più sicuro.

**Esempio di richiesta GET**

plaintext

Copia codice

GET /index.html HTTP/1.1

Host: www.esempio.com

Accept: text/html

User-Agent: Mozilla/5.0

Cookie: ID\_sess=1234ABCD

**5. Servlet e JSP**

**Ruoli**

* **JSP (JavaServer Pages)**: generano dinamicamente HTML incorporando Java.
* **Servlet**: gestiscono la logica delle richieste e possono passare dati alle JSP.

**Pattern MVC**

* **Controller**: Servlet
* **View**: JSP  
  La servlet passa dati alla JSP usando un oggetto simile a una HashMap.

**Codice di esempio**

Per includere una JSP in una servlet:

java

Copia codice

request.setAttribute("chiave", valore);

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

**6. Gestione delle Sessioni e dei Cookie**

**Sessione**

* Le **variabili di sessione** sono associate al browser e al server.
* Esempio:

java

Copia codice

HttpSession session = request.getSession();

session.setAttribute("user", user);

**Cookie**

* I cookie sono salvati nel client e possono tracciare l’utente.
* Esempio:

java

Copia codice

Cookie c = new Cookie("email", email);

c.setMaxAge(3600); // Durata in secondi

response.addCookie(c);

**Differenze**

* **Sessione**: salvata lato server, più sicura.
* **Cookie**: salvati lato client, meno sicuri ma utili.

**Domande per lo Studio**

**Generali**

1. Cosa succede quando digiti un URL nel browser?
2. Qual è la differenza tra una risorsa statica e una dinamica?
3. Perché un Servlet Container è necessario per gestire JSP?

**HTTP**

1. Spiega la differenza tra una richiesta GET e POST.
2. Quali sono alcune chiavi importanti di una richiesta HTTP?

**Servlet e JSP**

1. Qual è il ruolo di una servlet nel pattern MVC?
2. Come si passa un dato dalla servlet alla JSP?

**Sessioni e Cookie**

1. Cosa distingue una variabile di sessione da un cookie?
2. Scrivi un esempio di codice per creare un cookie e assegnargli un valore.

**Errori e Debug**

1. Quali sono le cause più comuni di un errore 404?
2. Cosa succede se il browser non riesce a riconoscere una risorsa?