

## Programmazione di Sistemi in Rete (A.A. 2017/18)

Esercitazione n. 9 del 10 novembre 2017

Prof. Eugenio Zimeo

### Esercizio 9.1

Si estenda l'implementazione del server Web realizzata con l'esercizio 8.3 in modo da prevedere la possibilità di scrivere applicazioni web secondo l'approccio **Server Script**.

Quando si intende eseguire codice sul server come azione conseguente alla ricezione di un messaggio HTTP, la risorsa specificata nella request line **non è un file da recuperare ed inviare al browser ma un frammento di codice da mandare in esecuzione**.

Tra le diverse opzioni possibili (processo esterno al server Web o modulo integrato nel server Web) realizzeremo il generatore di risorse come engine di componenti software direttamente integrato nel server Web. Tale engine consentirà di elaborare le richieste HTTP inoltrandole verso la componente software specificata nella request-line del messaggio di richiesta. Allo scopo, sia il server Web che le componenti applicative condivideranno la seguente interfaccia:

```
public interface WebLet {  
    public void service(HttpServletRequest req, HttpServletResponse res);  
}
```

Il parametro di tipo `HttpServletRequest` consente di recuperare il contenuto del messaggio di richiesta HTTP mentre il parametro di tipo `HttpServletResponse` consente di definire il messaggio di risposta HTTP dal codice applicativo.

I tipi `HttpServletRequest` e `HttpServletResponse` sono quelli definiti nella traccia dell'esercizio 8.3. Le implementazioni sono quelle realizzate con l'esercizio 8.3.

L'handler del protocollo HTTP dovrà prevedere sia il comportamento convenzionale di un server HTTP, sia l'attivazione dell'esecuzione di una componente software. Lo specifico comportamento potrà essere attivato, per ogni richiesta, o considerando la particolare estensione dei file che rappresentano le componenti software (es. `.class`), o considerando un particolare percorso nel path dell'URI (es. `/weblet/`).

Nella costruzione del messaggio di risposta si tenga conto che il contenuto non deve essere cachable. Utilizzare la specifica linea di intestazione per forzare l'adeguato comportamento dei proxy e dei browser.

Si completi l'esercizio, scrivendo una componente applicativa di tipo `WebLet` che invii al browser il messaggio "Hello WebLet World".