

Le Servlet

una breve introduzione

di

Giuseppe Sottile

♠ Introduzione

Una Servlet “(serv)let” altro non rappresenta se non un programma o applicativo realizzato allo scopo di estendere le capacità effettive di un server, più in particolare nell'ambito della programmazione web, viene impiegata per la realizzazione delle cosiddette web-app (applicazioni Java che possono essere eseguite direttamente dal browser).

♠ Il ciclo di vita di una servlet

Nel modello J2EE che prevede una stratificazione in tre livelli (client, server, model) il modulo che costituisce la servlet viene realizzato nel livello server, all'interno di un oggetto adibito all'interfacciamento con il server stesso chiamato “**container**”. In ogni istanza di un container, “vive”, per così dire una istanza di servlet insieme a tutte le risorse necessarie al corretto funzionamento (file, sockets.. etc), ogni qualvolta avviene una richiesta al servlet, in realtà è il container che fa da tramite lungo il percorso delle chiamate. Esso si preoccupa di creare il servlet se esse non sono mai state create e di invocare le relative procedure interne.

Ciò che accade può essere riassunto attraverso 3 soli metodi:

init service destroy

○ Se nel momento di una chiamata implicita e/o esplicita al server da parte del client non esiste nessuna istanza del modulo servlet, è compito del container creare una istanza di un servlet mediante una chiamata al metodo:

`init(ServletConfig servletConfig)`

Questo metodo prende come parametro un oggetto di tipo ServletConfig che contiene un riferimento al file di configurazione XML più propriamente detto in gergo **deployment descriptor** e cioè (descrittore di sviluppo). Cio che contiene questo file sono i parametri di configurazione della servlet ad esempio il nome

```

<web-app>

...

<servlet>
  <servlet-name>MiaServlet</servlet-name>
  <servlet-class>MiaServlet</servlet-class>
  <init-param>
    <param-name>parametro1</param-name>
    <param-value>Ciao, benvenuto sul mio sito.</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>

...

<servlet-mapping>
  <servlet-name>MiaServlet</servlet-name>
  <url-pattern>/miaservlet</url-pattern>
</servlet-mapping>

...

</web-app>

```

○ Se invece, è già attiva una istanza di servlet nel container e ciò significa che sicuramente almeno una volta il metodo `init()` è stato invocato e quindi la servlet è già in funzione pronta a ricevere richieste dall'esterno, esso (il container), si preoccupa solo di invocare un metodo speciale per rispondere alle richieste:

`service(ServletRequest sReq, ServletResponse sRes)`

In cui il primo parametro contiene informazioni sulla richiesta, il secondo sulla risposta. Attraverso l'uso dei parametri si possono invocare diversi metodi per settare alcuni degli attributi relativi agli oggetti richiesta e risposta.

○ Per ultimo il container si preoccupa di liberare le risorse e di arrestare il servlet mediante una chiamata implicita al metodo

`destroy()`

Ovviamente nel caso in cui il servlet lancia più threads il container deve attendere che essi abbiano esaurito il loro tempo o siano stati eliminati.