

Introduzione a JavaServer Faces

Maurizio Cozzetto

16 Settembre 2009

Cosa è JavaServer Faces (JSF)

JSF è un framework per lo sviluppo di applicazioni web Java *EE* (*Enterprise Edition*) che semplifica lo sviluppo dell'*interfaccia* web di un'applicazione e fornisce strumenti per la gestione del *flusso* di navigazione.

JSF è strutturato secondo il design pattern *MVC* (*Model-View-Controller*)¹ proponendo un modello di sviluppo dell'applicazione a *componenti* e a *eventi* simile a *Swing*, il package della Standard Edition per lo sviluppo di interfacce grafiche. Molti componenti sono pronti all'uso e rispondono alle necessità più comuni per lo sviluppo di un'interfaccia web ma il programmatore può anche realizzarne di propri (*custom components*, componenti personalizzati).

I servizi di JSF

Gli strumenti messi a disposizione da JSF sono i servizi tipici di cui ha bisogno uno sviluppatore web. In particolare, abbiamo:

- *Conversioni di tipo*. Sappiamo che un client può inviare a un server web solo ed esclusivamente valori di tipo stringa che poi vanno trasformati nei tipi corrispondenti. JSF semplifica e automatizza questa conversione.
- *Validazione*. La validazione dei campi è uno dei compiti più comuni e ripetitivi nello sviluppo di un'interfaccia web. JSF fornisce una serie di *validatori standard* (ma consente di realizzarne anche altri personalizzati) che sollevano lo sviluppatore da questa attività senza valore aggiunto (ad esempio la validazione di una data o di un campo numerico ecc).
- *Binding* (corrispondenza) tra *campi* dell'interfaccia e *attributi* di un *JavaBean*². I dati immessi in un modulo web, ad esempio, opportunamente validati, devono poi essere convertiti e *mappati* negli attributi di un Javabean.
- *Gestione del flusso di navigazione*. La navigazione tra le pagine web è semplificata e di solito gestita mediante file *xml* (file *web.xml*).
- *Rendering dei componenti*. L'aspetto grafico dei componenti è a carico di una classe distinta detta *Renderer* ma è possibile sceglierne altre (per default, il renderer è l'*HTML-RenderKit*).
- *Gestione degli errori*. JSF dispone dei meccanismi necessari per la gestione e la *segnalazione* degli errori a livello di interfaccia.

¹<http://it.wikipedia.org/wiki/Model-View-Controller>

²Per *JavaBean* solitamente intendiamo una normale *classe* pubblica, con attributi *privati* forniti dei corrispondenti metodi pubblici *getter* e *setter* e dotata almeno del *costruttore* di default. Se si vogliono poi *serializzare* gli oggetti ottenuti a partire da questa classe, allora la classe deve implementare anche l'interfaccia `java.io.Serializable`.

- *Internazionalizzazione (i18n)* dell'applicazione³.

Coming soon

³<http://it.wikipedia.org/wiki/I18n>