

781229 Giannini Federico

781266 Manfredelli Mauro

Processo e Sviluppo

Assignment 3

Model View Controller (MVC)

Prima consegna

Sommario

1.	Introduzione	1
2.	Java Resources.....	2
2.1.	Controller.....	2
2.2.	Model.....	2
2.3.	Service	2
3.	WebContent	3
4.	Pom.xml	4

1. Introduzione

Per l'implementazione dell'applicazione è stato utilizzato Spring Framework attraverso un Maven project di Eclipse.

All'interno dell'archivio è presente il workspace che contiene

- Il progetto Maven
- La cartella relativa al server Apache Tomcat 8 sui cui verrà eseguita l'applicazione
- La JDK 1.8 per la compilazione
- L'ambiente di sviluppo Eclipse Mars.

L'applicazione implementa il Design Pattern MVC in cui una Servlet (assignment3-servlet.xml) agisce da Dispatcher intercettando tutte le richieste provenienti dalle diverse view (realizzate tramite pagine jsp). Le richieste intercettate vengono, tramite l'utilizzo di un Handler Mapping, analizzate per l'individuazione del Controller specifico adibito alla risoluzione di tale richiesta.

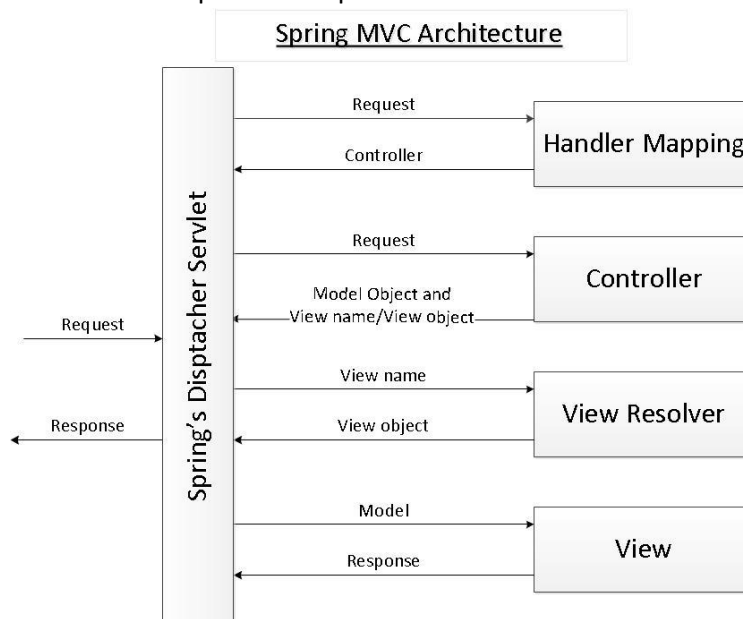
La Servlet trasferisce la richiesta al Controller che la elabora e, attraverso un Model and View, risponde comunicando la nuova vista e i parametri che essa utilizzerà per la presentazione.

A questo punto la Servlet contatterà il View Resolver che, attraverso il nome della vista, recupererà la pagina jsp corrispondente.

Il controllo verrà trasferito alla view che, attraverso i parametri, effettuerà la presentazione.

La view verrà inserita nella Response da inviare al client.

L'immagine seguente sintetizza il processo sopra descritto.



Il progetto è diviso in:

- Java Resources: folder contenente i seguenti package Java:
 - it.assignment3.controller: package contenente i controller che gestiscono le richieste relative agli utenti, gli errori e il primo accesso
 - it.assignment3.model: package che contiene le classi relative al modello dei dati
 - it.assignment3.service: package contenente il servizio che gestisce la logica di business.
- WebContent: folder contenente le risorse locali del progetto, le pagine jsp, la servlet e la configurazione web.
- pom.xml: file contenente la configurazione del progetto.

2. Java Resources

2.1. Controller

Il package `it.assignment3.controller` contiene 3 classi che implementano altrettanti controller.

2.1.1. IndexController

Intercetta la richiesta iniziale relativa al path `"/`, specificato in `web.xml`.

2.1.2. ErrorController

Controlla e gestisce gli errori e le eccezioni generate dall'applicazione.

Per il momento gli unici errori gestiti sono: 403, 404 e 500.

Il file `web.xml` configura il forward, nel caso di generazione di errore, a questo controller (con un path contenente il codice dell'errore).

Tutte le eccezioni sono mappate attraverso l'annotazione `@ExceptionHandler` nel metodo `handleAllException` (della classe `ErrorController`) che restituisce una pagina di errore.

2.1.3. UserController

Gestisce le richieste relative a:

- log in di un utente
- log out di un utente
- visualizzazione della lista di utenti da parte di un utente admin
- registrazione di un utente.

Ogni richiesta viene gestita da un metodo specifico prima del quale viene indicato il path che il Request Handler utilizzerà per l'individuazione del Controller (attraverso l'annotazione `@RequestMapping`).

I metodi che richiedono operazioni sulla lista di utenti interagiranno con la classe `UserService` che implementa la logica di business (contenuta nel package `it.assignment3.service`).

Ogni metodo ritorna un oggetto di classe `ModelAndView` che contiene la nuova vista selezionata e i parametri che essa utilizzerà.

2.2. Model

Il package `it.assignment3.model` contiene le classi che modellano il dominio degli utenti:

- `User`: contiene i dati dell'utente
- `SessionUser`: contiene i dati dell'utente loggato salvati nella sessione
- `Role`: modella i ruoli possibili di un utente: `user` o `admin`.

2.3. Service

Il package `it.assignment3.service` contiene la classe che implementa la logica di business relative alle operazioni sugli utenti.

Tale classe manterrà memorizzata la lista di tutti gli utenti e effettuerà su di essa le operazioni di aggiornamento e di login.

3. WebContent

Questa cartella implementa la gestione della parte web:

- WebContent/resources: path utilizzato per le risorse locali:
 - immagini
 - file JavaScript
 - file CSS
 - librerie: bootstrap, jquery, datatables, sweetalert2.
- WebContent/WEB-INF/classes: path contenente i file di properties per la configurazione di alcune componenti del progetto:
 - log4j.properties: definisce la stampa dei log dell'applicazione e la scrittura su un file esterno contenuto nella cartella logs del Tomcat
 - messages.properties: gestisce le label presenti nelle viste
- WebContent/WEB-INF/views: path che si riferisce alla cartella contenente le view.
Ogni view è realizzata attraverso una pagina dinamica jsp che, in base ai parametri contenuti nel model e ai tag jstl, permette la presentazione.
Ogni pagina jsp ha un suo file .js di scripting.
I tag jstl vengono utilizzati per la realizzazione dei costrutti relativi alla logica di programmazione: cicli e selezioni.

Sono presenti:

- tre pagine jsp per la gestione del flusso dell'applicazione:
 - user/index.jsp: pagina iniziale
 - user/login.jsp: pagina personale dell'utente
 - user/usersList.jsp: pagina che permette la visualizzazione dell'elenco degli utenti ad un utente admin
- tre pagine per la gestione degli errori (error/error403.jsp, error/error404.jsp, error/error500.jsp), ognuna relativa ad un errore diverso
- una pagina di layout (decorator/layout.jsp) che definisce il template, importa le librerie JavaScript e CSS e importa le funzioni generali (generic).
- WebContent/WEB-INF/assignment3-servlet.xml: file xml che configura il dispatcher (dispatcher servlet).

Tale file indica i bean che verranno utilizzati per:

- risorse locali
- viste
- adapter
- mapping
- validazione lato server
- label dell'applicazione
- view resolver.
- WebContent/WEB-INF/sitemesh3.xml: file xml che il file che gestisce il layout grafico di ogni pagina (layout.jsp).
- WebContent/WEB-INF/web.xml: file xml che indica:
 - nome dell'applicazione
 - nome della dispatcher servlet
 - url pattern iniziale
 - pagine di errore
 - presenza di sitemesh.

4. Pom.xml

Questo file xml contiene la configurazione del progetto relativamente a:

- build: compilazione e installazione del progetto attraverso i comandi Maven clean e install.
- dependencies: riferimenti esterni che il progetto utilizza e che l'applicazione dovrà scaricare al momento della build.

Nello specifico sono indicati:

- Spring Framework
- Validation-API e Hibernate-Validator
- Sitemesh
- JSTL
- Servlet-API.
- repositories: repository contenenti le risorse del framework.
- profiles: profili per building e definizione di properties specifiche.