
Corso di "Sviluppo di applicazioni Web"

Docente: Giovanni Grasso

- Dao in Spring

Un po' di terminologia

- I componenti della persistenza sono detti “Data Access Object” (DAO)
 - hanno il compito di effettuare tutti gli accessi alla base di dati
 - I componenti del modello sono detti “Data Transfer Object” (DTO)
 - hanno il compito di “trasportare” i dati caricati dalla base di dati fino al controllo e poi di nuovo fino alla base di dati
-

Comunicazione DAO/DTO

- Regole di comunicazione
 - i componenti di controllo creano e manipolano i componenti del modello (DTO)
 - i componenti di controllo usano i componenti di persistenza (DAO) per effettuare operazioni sulla strato di persistenza (DB)
 - i componenti del modello non conoscono i DAO
 - i componenti del controllo sono isolati rispetto alla tecnologia della persistenza (es: tipo di DBMS)
-

Caso d'uso

- Esempio: inserimento di un utente
 - il controllo acquisisce attraverso la vista i dati di un nuovo utente
 - il controllo crea un oggetto della classe Utente (DTO) e ne imposta le proprietà
 - Nome, nickname, *servizi usufruibili*, ecc.
 - il controllo chiede al DAOUtente di effettuare l'inserimento, fornendo il DTO
-

Distribuzione dei componenti

- Package della persistenza
 - un DAO ed un DTO per ogni tabella della base di dati
 - eventuali DAO aggiuntivi per effettuare operazioni non specifiche di una tabella particolare (es: interrogazioni o statistiche che coinvolgono più tabelle)
 - componenti specializzati (DataSource e DAOException)
-

DAO

- DAO
 - componente specializzato in operazioni sulla persistenza
 - tipicamente associato ad una tabella
 - Nel caso di una base di dati relazionale
 - ciascun DAO implementa le quattro operazioni fondamentali sulla tabella, le cosiddette operazioni “CRUD”
-

CRUD

- Operazioni CRUD su una tabella
 - “Create”: inserimento di una tupla
 - “Retrieve”: ricerca di tuple nella tabella sulla base di un criterio di selezione
 - “Update”: aggiornamento di una tupla
 - “Delete”: eliminazione di una tupla
 - le operazioni di tipo “retrieve” possono essere varie; es: ricerca per nickname, ricerca per nome, ecc.
-

CRUD: esempio

- DAOUtente

- ❑ `public void doInsert(Utente u)`
 - ❑ `public void doUpdate(Utente u)`
 - ❑ `public void doDelete(Utente u)`
 - ❑ `public Utente doSelectNickname(String nickname)`
 - ❑ `public List doSelectName(String name)`
-

DTO

- Si tratta tipicamente di JavaBeans
 - JavaBean
 - ❑ classe Java con caratteristiche particolari
 - ❑ costruttore senza argomenti
 - ❑ proprietà tutte private
 - ❑ metodi get e/o set che rispettano le convenzioni di stile per l'accesso e la modifica delle proprietà
 - ❑ eventuali altri metodi di logica applicativa
-

Concetti di Spring DAO

- Template

pattern che permette di definire la struttura di un algoritmo lasciando alle sottoclassi il compito di implementarne alcuni passi come preferiscono. In questo modo si può ridefinire e personalizzare parte del comportamento nelle varie sottoclassi senza dover riscrivere più volte il codice in comune.

Spring DAO Templates

- Built in code templates that support JDBC, Hibernate, JDO, and iBatis SQL Maps
 - Simplifies data access coding by reducing redundant code and helps avoid common errors.
 - Alleviates opening and closing connections in your DAO code.
 - Transaction management is handled by a wired bean
 - Code only needs to be implemented in callback methods.
 - `doInXXX(Object)`
-

Spring Dao

Spring supporta diversi sistemi di accesso ai dati tra cui:

JDBC (JDBCTemplate)

Hibernate (HibernateTemplate)

JDO (JDOTemplate)

Riduce drasticamente il codice da scrivere

Classi astratte di Spring per il supporto DAO

- Estendere i DAO utente dalla opportuna classe xxxDAOSupport, in base al meccanismo di persistenza scelto.

 - JdbcDaoSupport
 - Super class for JDBC data access objects.
 - Requires a DataSource to be set, providing a JdbcTemplate based on it to subclasses.

 - HibernateDaoSupport
 - Super class for Hibernate data access objects.
 - Requires a SessionFactory to be set, providing a HibernateTemplate based on it to subclasses.

 - JdoDaoSupport
-

Ex: Code without a template

```
public class OrderHibernateDAO implements IOrderDAO {
    public Order saveOrder(Order order) throws OrderException{
        Session s = null;
        Transaction tx = null;
        try{
            s = ... // get a new Session object
            tx = s.beginTransaction();
            s.save(order);
            tx.commit();
        } catch (HibernateException he){
            // log, rollback, and convert to OrderException
        } catch (SQLException sqle){
            // log, rollback, and convert to OrderException
        } finally {
            s.close(); // needs a try/catch block
        }
        return order;
    }
}
```

Ex: Spring DAO Template Example

```
public class OrderHibernateDAO extends HibernateDaoSupport
    implements IOrderDAO {
    ...
    public Order saveOrder(final Order order) {
        return (Order) getHibernateTemplate().execute(new
        HibernateCallback() {
            public Object doInHibernate(Session session)
                throws HibernateException, SQLException {

                session.save(order);
                return order;
            }
        });
    }
    ...
}
```

Spring – Hibernate, configuration how to

- *Definire un bean dataSource della classe DriverManagerDataSource di Spring, che consente la connessione al database (url, driver, user, psw)*
 - *Definire un bean per la sessionFactory di hibernate utilizzando la classe AnnotationSessionFactoryBean.*
 - *Nella proprietà “annotatedClasses” elencare le classi entità del modello.*
 - *Settare il riferimento al bean datasource*
 - *Settare la proprietà “hibernateProperties” con le proprietà desiderate e.g. hibernate.dialect = org.hibernate.dialect.HSQLDialect*
 - *Definire i bean utente, settando la dipendenza dalla sessionFactory*
-

Esercizio

- Dato lo schema
 - User(userName, name, password, email)
 - Post(author,title, text, data) , Comment(author, post, text, data)

Un utente può creare diversi post, e diversi commenti su un post esistente. La cancellazione di un utente provoca la cancellazione di tutti i suoi post, mentre la cancellazione di un post provoca la cancellazione di tutti i commenti associati

- Si implementino le entità hibernate necessarie per il modello, con i relativi DAO. Ogni DAO deve consentire almeno di:
 - Save di oggetti transienti, delete di oggetti persistenti, update di oggetti dirty
 - Ottenere la lista di tutti gli oggetti di una entità – findAll()
 - Ottenere un oggetto con il dato id - find(id)
 - Ottenere un oggetto by example - find(example)
 - Si utilizzi il supporto di Spring per i DAO di Hibernate
 - Se eseguano i test Junit per i metodi di ognuno dei DAO realizzati
-