Library DAO Tutorial

Step 2: JdbcTemplate Version – Creating the Model

SOFTWARE GUILD

The Learning House

427 S 4th Street #300

Louisville KY 40202

Step 2: JDBCTemplate Version – Creating the Model

Overview

In this step, we will create the model objects for the JdbcTemplate version of our application. The approach taken for this version of the DAO is a bit more database-oriented than object-oriented. This can be seen in the Book object, where we have an array of author ids and one publisher id rather than an array of Author objects and a Publisher object.

Author

Author is a simple Java object with a set of private fields and getters/setters for each. Note that the Author object contains no object or id references to other model objects in the system.

```java
package com.swcguild.library.model;

public class Author {
    private int authorId;
    private String firstName;
    private String lastName;
    private String street;
    private String city;
    private String state;
    private String zip;
    private String phone;

    public int getAuthorId() {
        return authorId;
    }
    public void setAuthorId(int authorId) {
        this.authorId = authorId;
    }
    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
    public String getStreet() {
        return street;
    }
    public void setStreet(String street) {
        this.street = street;
    }
    public String getCity() {
        return city;
    }
    public void setCity(String city) {
        this.city = city;
    }
    public String getState() {
        return state;
    }
    public void setState(String state) {
        this.state = state;
    }
    public String getZip() {
        return zip;
    }
    public void setZip(String zip) {
        this.zip = zip;
    }
    public String getPhone() {
        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }
}
```

Publisher

Publisher is also a simple Java object consisting of private fields and getters/setters. Like Author, Publisher has no object or id references to other model objects in the system.

```
package com.swcguild.library.model;

public class Publisher {
    private int publisherId;
    private String name;
    private String street;
    private String city;
    private String state;
    private String zip;
    private String phone;

    public int getPublisherId() {
        return publisherId;
    }
    public void setPublisherId(int publisherId) {
        this.publisherId = publisherId;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getStreet() {
        return street;
    }
    public void setStreet(String street) {
        this.street = street;
    }
    public String getCity() {
        return city;
    }
    public void setCity(String city) {
        this.city = city;
    }
    public String getState() {
        return state;
    }
    public void setState(String state) {
        this.state = state;
    }
    public String getZip() {
        return zip;
    }
    public void setZip(String zip) {
        this.zip = zip;
    }
    public String getPhone() {
        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }
}
```

Book

Book is similar to Author and Publisher in that it consists of private fields and getters/setters. Book differs from the other model objects in that it contains database id references to the authors and the publisher of the book. The array of author ids in the Book object represent the many-to-many relationship between books and authors. The publisher id in the Book object represents the many-to-one relationship between publishers and books.

Ids (rather than Author and Publisher objects) are used here because they map to the database and can easily be used to drive dropdown menus in the JSPs. This is a design choice for this implementation — with a little more work in the DAO we could have used objects instead of ids.

```java
package com.swcguild.library.model;

import java.math.BigDecimal;
import java.time.LocalDate;

public class Book {
    private int bookId;
    private String isbn;
    private String title;
    private int publisherId;
    private int[] authorIds;
    private BigDecimal price;
    private LocalDate publishDate;

    public int getBookId() {
        return bookId;
    }
    public void setBookId(int bookId) {
        this.bookId = bookId;
    }
    public String getIsbn() {
        return isbn;
    }
    public void setIsbn(String isbn) {
        this.isbn = isbn;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public int getPublisherId() {
        return publisherId;
    }
    public void setPublisherId(int publisherId) {
        this.publisherId = publisherId;
    }
    public int[] getAuthorIds() {
        return authorIds;
    }
    public void setAuthorIds(int[] authorIds) {
        this.authorIds = authorIds;
    }
    public BigDecimal getPrice() {
        return price;
    }
    public void setPrice(BigDecimal price) {
        this.price = price;
    }
    public LocalDate getPublishDate() {
        return publishDate;
    }
    public void setPublishDate(LocalDate publishDate) {
        this.publishDate = publishDate;
    }
}
```

Wrap-up

In this step, we did the following:

1. Created the model objects for our DAO.
2. Showed how one-to-many (Publishers to Books) relationships can be represented in our model.
3. Showed how many-to-many (Authors to Books) relationships can be represented in our model.
4. Implemented our models in a more database-oriented manner (rather than purely object-oriented).

In the next step, we will design the interface for the JdbcTemplate version of the DAO.