Difference Between Spring DAO vs Spring ORM vs Spring JDBC

priyarajtt

The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE platform.

Spring-DAO

Spring-DAO is not a spring module. It does not provide interfaces or templates to access the data. One important change that needs to be written while using Spring DAO is, that it has to be annotated with **@Repository**. The reason for doing this is the exceptions that may arise in the underlying technology like JDBC, Hibernate, JPA, etc. are consistently translated into their respective DataAccessException subclass. Let us see this with one example of a Student service scenario.

Initially, Hibernate is the persistence mechanism that got used. Suppose let us assume that HibernateException is caught at the service layer. There should be steps available to catch it. But at some point in time, instead of Hibernate, it has been changed to JPA, then no need to change the DAO interfaces.

Instead, if it is annotated with @Repository, then the exceptions related to the current underlying technologies will be directly translated to the spring DataAccessException. Because of this feature, though the underlying technologies are changed from hibernate to JPA or from JPA to hibernate, then the same Spring DataAccessExceptions will still be thrown. According to the underlying technologies, the spring will translate according to their native exceptions.

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Got It !

Limitations in using Spring DAO related to the exceptions

- Should not catch persistence exceptions
- The hierarchy of exceptions is usually richer and more meaningful than the one provided by spring. But there is no mapping from one provider to the other. The reason for adding @Repository in the DAO is the beans are automatically added by the scan procedure. Spring has the tendency to add other useful features to the annotation.

Sample code snippet related to Spring DAO. Service implementation layer has to get annotated with @Repository followed by its corresponding service layer.

Java

```
// Necessary imports
@Repository('<Specify the DAO that is getting accessed')
public class StudentDAOImplementation
    extends HibernateDao<Student, Long>
    implements StudentDAO {

    @Override public boolean remove(Student studentObject)
    {
        // Write necessary steps
        return true;
    }
}
```

Java Arrays Java Strings Java OOPs Java Collection Java 8 Tutorial Courses @SALE Java Multithrea

Spring ORM

Spring-ORM is a very efficient module that plays as an umbrella pattern. The reason for calling this an umbrella is it covers many persistence technologies, namely JPA, JDO,

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Spring. With that integration classes, each technology integrates easily with Spring transaction management. Mainly injection of DataSource is done via SessionFactory or EntityManagerFactory etc. bean. In the case of pure JDBC, apart from JdbcTemplate, there is no need for any integration class as JDBC(Java database connectivity) directly relies on data sources. For each technology, the configuration basically consists in injecting a DataSource bean into some kind. For pure JDBC, there's no need for such integration classes (apart from JdbcTemplate), as JDBC only relies on a DataSource. Spring-JDBC is not required in the case of ORM like JPA or Hibernate but Spring-Data is required. Spring-Data is nothing but an umbrella project and it can provide a common API that defines accessing the DAO and annotations and it covers both SQL and NoSQL data sources. Model classes have to get annotated with @Entity and in that primary key has to get annotated with @Id. The sample code for Student Model Class is given below.

Java

```
// Necessary import statements

// This is much required and here model class
// should match with database table name
@Entity
public class Student {
  @Id
  private int studentId;
  // other necessary attributes like name, address etc.,
  // Corresponding getter and setter methods
}
```

Spring ORM DAO and service class has to get annotated with @Component

Java

```
// Necessary import statements
@Component
public class StudentDAO {
  @PersistenceContext
  private EntityManager em;
  // Rest set of code
}
```

Spring JDBC

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Configuration with a DataSource is mandatory. The sample code is given below.

Java

```
// necessary import statements
int totalStudents = jdbcTemplate.queryForObject("select count(1) from Student", Intege
Student student = jdbcTemplate.queryForObject("select name, address from Student where
            rs -> new Student(rs.getString(1), rs.getString(2)),
            12345);
```

The advantage of using Spring-JDBC is it provides a JdbcDaoSupport, It is useful for extending DAO. It has 2 properties namely a DataSource and a JdbcTemplate. They are helpful for implementing DAO methods. Additionally, there is an exceptions translator available which translates from SQL exceptions to spring DataAccessExceptions.

Difference Table

Spring DAO	Spring ORM	Spring JDBC
Generalized concept and @Repository annotation are mandatory.	Easy integration with Spring with the following • SessionFactory for Hibernate • EntityManagerFactory for JPA • SqlSessionFactory for MyBatis	For plain JDBC calls.
Data access implementation is totally separated and hence it is independent of the database.	Multi-technology implementation is possible by integrating with the required tools.	If the application is not complex and diversified and lies on a single database, we can use this and it is efficient.
An additional layer and its dependencies need to be specified. Hence it may take some time to start if the application is	An additional layer and its dependencies need to be specified. Hence it may take some time to start if the application is complex	As this is straightforward, no complex dependencies are required but portability will become

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Spring DAO	Spring ORM	Spring JDBC
Maintenance issues will be there because of the complexity of the additional layer.	Maintenance issues will be there because of the complexity of the additional layer.	Here less maintenance only.
Design patterns like Factory classes, and Data Transfer Object(DTO) are needed to get implemented along with DAO.	Got the support for multiple technologies like Hibernate, JPA, and iBatis.	Implementation is simple. If relying on a single database and direct query purpose means can depend on this.

Last Updated : 28 Nov, 2022

7

Similar Reads

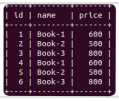
Spring Boot - Spring JDBC vs Spring Data JDBC

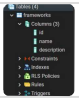
Hibernate - Difference Between ORM and JDBC

Spring - ORM Framework

Spring ORM Example using Hibernate


What is Spring Framework and Hibernate ORM?


Spring - Using SQL Scripts with Spring JDBC + JPA + HSQLDB


Spring - Prepared Statement JDBC Template


Spring Boot - JDBC

Spring JDBC Example

Advantages of Spring Boot JDBC

Related Tutorials


Java AWT Tutorial

Spring MVC Tutorial

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Spring Tutorial

Java 8 Features - Complete Tutorial

Spring Boot Tutorial

Previous Next

Article Contributed By :

priyarajtt

priyarajtt

Follow

Vote for difficulty

Easy | Normal | Medium | Hard | Expert

Improved By : modalaashwin41

Article Tags : Java-Spring, Picked, Difference Between, Java

Practice Tags : Java

Improve Article | Report Issue

GeeksforGeeks
A-143, 9th Floor, Sovereign Corporate Tower, Sector-136, Noida, Uttar Pradesh - 201305

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Company

About Us

Legal

Careers

In Media

Contact Us

Advertise with us

Explore

Job-A-Thon Hiring Challenge

Hack-A-Thon

GfG Weekly Contest

Offline Classes (Delhi/NCR)

DSA in JAVA/C++

Master System Design

Master CP

Languages

Python

Java

C++

PHP

GoLang

SQL

R Language

Android Tutorial

DSA Concepts

Data Structures

Arrays

Strings

Linked List

Algorithms

Searching

Sorting

Mathematical

Dynamic Programming

DSA Roadmaps

DSA for Beginners

Basic DSA Coding Problems

Complete Roadmap To Learn DSA

DSA for FrontEnd Developers

DSA with JavaScript

Top 100 DSA Interview Problems

All Cheat Sheets

DSA Roadmap by Sandeep Jain

Web Development

HTML

CSS

JavaScript

Bootstrap

ReactJS

AngularJS

NodeJS

Express.js

Lodash

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Computer Science

GATE CS Notes

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Python

Python Programming Examples

Django Tutorial

Python Projects

Python Tkinter

OpenCV Python Tutorial

Python Interview Question

Data Science & ML

Data Science With Python

Data Science For Beginner

Machine Learning Tutorial

Maths For Machine Learning

Pandas Tutorial

NumPy Tutorial

NLP Tutorial

Deep Learning Tutorial

DevOps

Git

AWS

Docker

Kubernetes

Azure

GCP

Competitive Programming

Top DSA for CP

Top 50 Tree Problems

Top 50 Graph Problems

Top 50 Array Problems

Top 50 String Problems

Top 50 DP Problems

Top 15 Websites for CP

System Design

What is System Design

Monolithic and Distributed SD

Scalability in SD

Databases in SD

High Level Design or HLD

Low Level Design or LLD

Top SD Interview Questions

Interview Corner

Company Wise Preparation

Preparation for SDE

Experienced Interviews

Internship Interviews

Competitive Programming

Aptitude Preparation

GfG School

CBSE Notes for Class 8

CBSE Notes for Class 9

CBSE Notes for Class 10

CBSE Notes for Class 11

CBSE Notes for Class 12

English Grammar

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Accountancy
Business Studies
Economics
Management
Income Tax
Finance
Statistics for Economics

Polity Notes
Geography Notes
History Notes
Science and Technology Notes
Economics Notes
Important Topics in Ethics
UPSC Previous Year Papers

SSC/ BANKING

SSC CGL Syllabus
SBI PO Syllabus
SBI Clerk Syllabus
IBPS PO Syllabus
IBPS Clerk Syllabus
Aptitude Questions
SSC CGL Practice Papers

Write & Earn

Write an Article
Improve an Article
Pick Topics to Write
Write Interview Experience
Internships

@geeksforgeeks , Some rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy