

Spring and Hibernate Frameworks



معرفی دوره:

Spring محبوب ترین چارچوب کاری جهت توسعه نرم افزارهای تجاری و سازمانی به زبان جاوا می باشد. میلیون ها برنامه نویس در سراسر جهان ، از چارچوب کاری Spring برای تولید نرم افزارهایی با کارایی بالا ، ساده جهت تست و قابل استفاده مجدد از کدهای آن ، استفاده می کنند . تعدد ماژولها و راهکارهای مختلف چارچوب Spring برای لایه های مختلف نرم افزار، تمایل به استفاده از این چارچوب را روز افزون نموده است . در کنار چارچوب Spring یادگیری چارچوب Hibernate بمنظور فراهم سازی لایه دسترسی و مدیریت داده به زبان جاوا از اهمیت بالایی برخوردار است . Hibernate یک نگاهت ما بین مفاهیم و موجودیت های شی گرا با موجودیت های بانک اطلاعات می باشد . همچنین ابزاری را برای بازیابی دادهها فراهم میآورد . Hibernate . کمک میکند یک برنامه نویس بدون استفاده مستقیم از دستورات مربوط به پایگاه داده با آن رابطه برقرار و شی ها را بازیابی، ذخیره یا به روزرسانی کند . اگر قصد دارید که از چارچوب Spring در پروژه خود استفاده کنید، Spring راه کارهایی برای Hibernate نیز ارائه کرده است تا بتوانید داده های موجود در پایگاه داده را نیز مدیریت کنید.

پیش نیاز:

آشنایی کامل و سابقه کار بر روی JSE

آشنایی کامل و سابقه کار بر روی JEE

تعداد ساعات:

۲۵ ساعت

Spring 5.0 Introduction

- Motivation behind Spring
- What is Spring
- The Spring Framework
- The Spring Core
- Where you can use Spring
- Why use Spring
- Why not use Spring

Spring Beans

- Spring Beans
- Spring Container
- A Simple Spring Application
- Spring Context config files
- Spring test Application
- What happens in a Spring application?

Spring Configuration

- Bean configurations
- Bean Scoping
- Spring Annotations
- Java Config approach
- Lambda approach
- Mixing XML and annotations

Dependency Injection

- Inversion of Control and Dependency Injection
- Spring Dependency Injection
- Wiring Beans together using XML
- Wiring Beans together using Annotations
- Wiring Beans together using a mixture of XML and Annotations

Further Configuration and Bean Lifecycle

- Wiring Collections
- Auto-wiring in XML files
- Compound Names
- Look at Bean Lifecycle

- Explore interface and annotations
- Understand PostProcessors
- POJO *init* and *destroy* methods
- Lifecycle interfaces

SpEL and Validation

- Introduce SpEL
- How SpEL is used
- Some examples
- Spring Validation
- Enabled validation
- Constraints available

Externalizing Properties

- Introduce Property values
- Property files
- Properties Resolver
- XML configuration
- Java Config configuration
- Use of @Value

Spring Environment and Profiles

- Introduce need for Environments
- Spring Environments
- Spring Profiles
- Defined in XML and Java Config
- Setting the active Profile
- Multiple active Profiles
- Default Profile

Bean Factories

- Factory Classes
- Static Factory classes
- Non-Static Factory objects
- Annotated Factory classes

Spring Test

- Introduce Testing with Spring
- Highlight Test Support in Spring
- Support for both JUnit 4.x and JUnit 5.x
- JUnit 4.x
- Spring and JUnit 4.x
- JUnit 5.x
- Spring and JUnit 5.x

Spring JMS

- Basic idea of message queues
- point-to-point
- publish and subscribe (pub-sub)
- ActiveMQ message server
- JMS (Java Message Service)
- Spring JMS Development
- Spring POJO bean as a message consumer

Spring Data

- Data Access objects
- Spring DAO
- Spring and JDBC
- Spring DAO Exception Hierarchy
- Spring Template
- Spring DAO Templates
- DAO Architecture
- Using Spring JDBC Templates
- Using Object Mappers

Spring AOP

- Motivation for Aspect Oriented Programming
- AOP in general
- How AOP can be achieved
- Spring AOP
- AOP Terminology
- Building in AOP interceptors
- Creating your own Aspect
- Applying a custom aspect in an application
- Challenges of AOP

Spring Transaction Management

- Spring Transaction management
- Declarative Transaction Management
- Transaction Managers

- Transaction Propagation
- Isolation levels
- Transaction Management
- Spring configuration
- Programmatically controlling a Transaction

Spring Web MVC

- Introduce Spring Web MVC
- Present flow through MVC
- Creating a simple Web MVC application
- Defining a Controller
- Handling view resolution
- Building and Deploying
- Java Config web applications
- Web AppInitializer class
- Spring Java Config class

Spring Rest

- Web Services
- What is REST
- Spring REST
- Building a Spring REST application

Spring Rest Client

- Introduce Spring REST Clients
- Spring REST Template
- REST client with XML and Java Config

Spring Boot

- Spring Boot Library
- Motivation
- Key Features Auto-configuration, Standalone, Opinionated
- Maven configuration

Introduction to Spring Security

- Configure Spring Security for HTTP BASIC authentication.
- Implement form-based authentication.
- Configure authentication features including remember-me, anonymous users, and logout.
- Apply authorization constraints to URLs and URL patterns.
- Method and Instance Authorization.

Introduction to Hibernate

- Overview of Hibernate Architecture
- Object Persistence
- O/R mismatch
- Object Relational Mapping (ORM)
- Java ORM/Persistent Frameworks
- A simple Hibernate application

Hibernate Query Language (HQL)

- Fetching options
- Hibernate queries
- HQL parameters
- Externalizing queries
- Using the Criteria API

Persistent Objects

- Using Hibernate with Spring
- Entity Lifecycle
- Transient, Persistent, and Detached States
- Persistent Object Updates

Mapping using Annotations

- Hibernate and JPA annotations
- Class-to-table mappings
- Property Mapping
- Identifiers - Generators and Keys
- Mapping an Entity to multiple tables

Hibernate Mapping Associations

Fetching strategies

- Many-to-one
- Bidirectional many-to-one
- One-to-one
- Bidirectional one-to-one
- Many-to-many
- Bidirectional many-to-many
- Components
- Cascading

Mapping Inheritance

- Inheritance Mapping Strategies
- Table per Concrete Class
- Table per Subclass
- Table per Hierarchy
- Unions
- Polymorphism

Spring and Hibernate Transaction Management

- Hibernate transaction configuration
- Hibernate Transaction API
- Transaction Propagation

Performance Tuning and Optimization

- 2nd-Level Cache
- Batching
- DML-style operations

Industrial-strength Hibernate

- Optimistic locking and versioning
- Pessimistic locking
- Mixing Hibernate and JDBC

Interceptors and Events

- Defining interceptors
- Defining events
- Recommendation