سرفصل آموزشی

پک حرفه ای متخصص برنامه نویسی جاوا

فهرست سرفصل‌های دوره‌های آموزشی

سرفصل‌های دوره آموزشی

JavaSpring

Part 1: Spring Core Technology:

Instantiation and Configuration

- Configuring Through Properties
- Configuration Namespaces
- The p: Notation
- Bean (Configuration) Inheritance
- Configuring Through Constructors
- Bean Post-Processors
- Lifecycle Hooks
- Integrating Existing Factory Code
- Awareness Interfaces

The IoC Container

- Introduction to the Spring IoC Container and Beans
- Container Overview
- Bean Overview
- Dependencies
- Bean Scopes
- Customizing the Nature of a Bean
- Bean Definition Inheritance
- Container Extension Points
- Annotation-based Container Configuration
- Class path Scanning and Managed Components
- Using JSR 330 Standard Annotations

- Java-based Container Configuration
- Environment Abstraction
- Registering a LoadTimeWeaver
- Additional Capabilities of the Application Context
- The BeanFactory

Resources

- Introduction
- The Resource Interface
- Built-in Resource Implementations
- The Resource Loader
- The ResourceLoaderAware interface
- Resources as Dependencies
- Application Contexts and Resource Paths

Dependency Injection

- Assembling Object Graphs
- Dependency Injection
- Single and Multiple Relationships
- The Utility Schema
- Using Spring Expression Language (SpEL)
- Inner Beans
- Auto wiring
- @Component, @Service, & Company
- @Autowired Properties
- Best Practices with Spring Annotations
- Java Classes as @Configurations
- Annotation Config Application Context
- Capabilities and Limitations
- Mixing and Importing XML and Java Configurations

Assembling Object Models

- Collections and Maps
- Support for Generics
- The Spring Utility Schema (util:)
- Autowiring to Multiple Beans
- Order of Instantiation
- Bean Factory vs. Application Context

Validation, Data Binding, and Type Conversion

- Validation by Using Spring’s Validator Interface
- Resolving Codes to Error Messages
- Bean Manipulation and the BeanWrapper
- Spring Type Conversion
- Spring Field Formatting
- Configuring a Global Date and Time Format
- Spring Validation

Spring Expression Language (SpEL)

- Evaluation
- Expressions in Bean Definitions
- Language Reference
- Classes Used in the Examples

Aspect Oriented Programming with Spring

- AOP Concepts
- Spring AOP Capabilities and Goals
- AOP Proxies
- @AspectJ support

- Schema-based AOP Support
- Choosing which AOP Declaration Style to Use
- Mixing Aspect Types
- Proxying Mechanisms
- Programmatic Creation of @AspectJ Proxies
- Using AspectJ with Spring Applications
- Further Resources

Data Buffers and Codecs

- DataBufferFactory
- DataBuffer
- PooledDataBuffer
- DataBufferUtils
- Codecs
- Using DataBuffer

Part 2: Data Access:

- Transaction Management
-
- Advantages of the Spring Framework’s Transaction Support Model
- Understanding the Spring Framework Transaction Abstraction
- Synchronizing Resources with Transactions
- Declarative transaction management
- Programmatic Transaction Management
- Choosing Between Programmatic and Declarative Transaction Management
- Transaction-bound Events
- Application server-specific integration
- Solutions to Common Problems
- Further Resources

DAO Support

- Consistent Exception Hierarchy
- Annotations Used to Configure DAO or Repository Classes

Persistence with JDBC

- Reducing Code Complexity
- The DataAccessException Hierarchy
- JdbcTemplate
- RowMapper<T> and ResultSetExtractor<T>
- The DaoSupport Hierarchy
- Capturing Generated Keys
- Transaction Control
- Platform Transaction Manager
- Transaction Template
- Isolation Levels
- Transaction Propagation

Persistence with JPA

- Object/Relational Mapping
- The Java Persistence API
- JpaDaoSupport and JpaTemplate
- @PersistenceUnit and @PersistenceContext
- Shared Entity Managers
- Using <tx:annotation-driven>
- The @Transactional Annotation
- Isolation and Propagation
- A Limitation of @Transactional

- Understanding Entity States
- Configuring JPA Without persistence.xml
- Bean Validation in JPA
- Optimistic Locking

Marshalling XML by Using Object-XML Mappers

- Introduction
- Ease of configuration
- Consistent Interfaces
- Consistent Exception Hierarchy
- Marshaller and Unmarshaller
- Using Marshaller and Unmarshaller
- XML Configuration Namespace
- JAXB
- JiBX
- XStream

Part 3: Web on Servlet Stack:

Spring Web MVC

- DispatcherServlet
- Filters
- Annotated Controllers
- Functional Endpoints
- URI Links
- Asynchronous Requests
- CORS
- Web Security
- HTTP Caching
- View Technologies

- MVC Config
- HTTP/2

REST Clients

- Rest Template
- Web Client

WebSockets

- Introduction to WebSocket
- HTTP Versus WebSocket
- When to Use WebSockets
- WebSocket API
- SockJS Fallback
- STOMP

Assembling Object Models

- Collections and Maps
- Support for Generics
- The Spring Utility Schema (util:)
- Auto wiring to Multiple Beans
- Order of Instantiation
- Bean Factory vs. Application Context

Handling Requests

- Matching URLs
- Identifying Views
- Request Parameters
- Injectable Parameters
- Command Objects
- Return Types
- HTTP Methods
- Path Variables
- Scope and Granularity of Command Objects
- Headers and Cookies
- RESTful Web Services

Working with Forms

- Form Processing in Spring MVC
- Command Objects in Request Methods
- Spring Custom Tags
- <form:form> and Friends
- Text Fields, Check Boxes, and Buttons
- Radio Buttons and Select/Option Lists
- Command objects at Session Scope
- Limitations of @SessionAttributes

Configuring Spring MVC

- Configuring Message Sources
- Resolving Error Codes
- Codes for Bean Validation

- HandlerExceptionResolver
- @ExceptionHandler
- @ControllerAdvice for Global Error Handling

Interceptors

- Interceptors
- Configuring Interceptors
- Filters in the Request-Handling Cycle

Part 4: Thymeleaf Template Engine:

- Introducing Thymeleaf
- The Good Thymes Virtual Grocery
- Using Texts
- Standard Expression Syntax
- Setting Attribute Values
- Iteration
- Conditional Evaluation
- Template Layout
- Local Variables
- Attribute Precedence
- Comments and Blocks
- Inlining
- Textual template modes
- Some more pages for our grocery
- More on Configuration
- Template Cache
- Decoupled Template Logic

Part 5: Testing:

- Introduction to Spring Testing
- Unit Testing
- Mock Objects
- Unit Testing Support Classes
- Integration Testing
- Goals of Integration Testing
- JDBC Testing Support
- Annotations
- Spring TestContext Framework
- Spring MVC Test Framework
- WebTestClient
- Testability of Spring Applications
- Dependency Injection
- SpringJUnit4ClassRunner
- @ContextConfiguration
- Preserving Test Isolation
- @DirtiesContext
- Mocking Spring MVC
- Building Requests
- Checking the ModelAndView
- Profiles
- Testing Persistence Components

Part 6: Web on Reactive Stack:

Spring WebFlux

- Overview
- Reactive Core
- DispatcherHandler
- Annotated Controllers

- Functional Endpoints
- URI Links
- CORS
- Web Security
- View Technologies
- HTTP Caching
- WebFlux Config
- HTTP/2

WebClient

- Configuration
- retrieve()
- exchange()
- Request Body
- Client Filters
- Synchronous Use
- Testing

WebSockets

- Introduction to WebSocket
- WebSocket API

RSocket

- Overview
- RSocketRequester
- Annotated Responders
- MetadataExtractor

Part 7: Integration:

- Remoting and Web Services
-
- RMI
- Using Hessian to Remotely Call Services through HTTP
- Spring HTTP Invoker
- Java Web Services
- JMS
- AMQP
- Considerations when Choosing a Technology
- REST Endpoints

Enterprise JavaBeans (EJB) Integration

Accessing EJBs

JMS (Java Message Service)

- Using Spring JMS
- Sending a Message
- Receiving a Message
- Support for JCA Message Endpoints
- Annotation-driven Listener Endpoints
- JMS Namespace Support

JMX

- Exporting Your Beans to JMX
- Controlling the Management Interface of Your Beans
- Controlling Object Name Instances for Your Beans

- Using JSR-160 Connectors
- Accessing MBeans through Proxies
- Notifications
- Further Resources

JCA CCI

- Configuring CCI
- Using Spring’s CCI Access Support
- Modeling CCI Access as Operation Objects
- Transactions

Email

- Usage
- Using the Java Mail Mime Message Helper

Task Execution and Scheduling

- The Spring Task Executor Abstraction
- The Spring Task Scheduler Abstraction
- Annotation Support for Scheduling and Asynchronous Execution
- The task Namespace
- Using the Quartz Scheduler

Cache Abstraction

- Understanding the Cache Abstraction
- Declarative Annotation-based Caching
- JCache (JSR-107) Annotations
- Declarative XML-based Caching
- Configuring the Cache Storage

- Plugging-in Different Back-end Caches
- How can I Set the TTL/TTI/Eviction policy/XXX feature?

سرفصل‌های دوره آموزشی

Microservices

& Distributed Systems

Microservices Architecture (Spring + docker + kubernetes)

Architecture-Big picture and Concepts:

- What is Microservices
- What is monolithic
- Pros and cons of Microservices Architecture
- Why Microservices
- When Microservices
- Steps to design

Design of Microservices:

- Domain driven design
- Clean code
- Maven
- Structure
- Modularity

Spring boot:

- Spring Web MVC
- REST Clients

Spring cloud:

- Service Discovery
- Distributed Tracing, …
- Monitoring and log
- Test
- Feign client
- RestTemplate

Spring data jpa:

Structure and principles

Rabbitmq:

- Architecture and fundamentals
- Install and configuration
- Producers and consumers
- Spring boot integration

Kafka:

- Architecture and fundamentals
- Install and configuration

- Producers and consumers
- Spring boot integration
- Why and where kafka

Docker and dockerize spring application:

- What is Docker?
- Docker Image
- Docker Image Architecture
- Docker Container
- Docker Reistry
- Install Docker
- Docker compose
- Spring Tools

Kubernetes:

- Understand the Kubernetes API primitives
- Understand the Kubernetes cluster architecture
- Understand Services and other network primitives
- Run docker image on k8s

Microservices security:

- Architecture and big picture
- Basic Authentication
- Authentication server

Microservices design patterns:

- Saga
- Service locator
- Api gateway
- QRS
- Event-driven
- Circuit breaker
- Transactional Outbox
- Api Composition