**BHARGAV PAMULAPATI**

**Software Developer**

**Email:** [brookspamulapati@gmail.com](mailto:brookspamulapati@gmail.com) **Phone:** 469-581-8483 **LinkedIn:** <https://www.linkedin.com/in/pamulapat/>

**PROFESSIONAL SUMMARY:**

* Around 12 years of experience in Java/J2EE, Software Development Life Cycle, Object Oriented programming, developing, and testing of Client/Server, Enterprise, and Web Applications.
* Led and managed projects through all phases of the Software Development Life Cycle (SDLC), including requirement analysis, design, development, testing, deployment, and maintenance, ensuring high-quality deliverables and adherence to timelines.
* Implemented database solutions tailored to the project's specific needs, using SQL (MySQL, Oracle, PostgreSQL, SQL Server) and NoSQL (MongoDB, Cassandra) technologies.
* Utilized Java 8's functional programming features, such as Lambda expressions and method references, to write concise and expressive API endpoints.
* Developed custom ETL scripts with Python to integrate and synchronize data across heterogeneous systems, utilizing APIs, flat files (CSV, JSON), and databases (SQL/NoSQL), enabling seamless data flow between applications.
* Extensively used the J2EE design patterns like Business Delegate, Service Locator, Value Object Data Access Object leveraged Spring Boot's automated REST API documentation using Swagger and API annotations.
* Proficiency in Java backend development, with expertise in Core Java (collections, multithreading, design patterns) Java 8 features (lambda expressions, streams, functional programming) Multithreading (concurrent programming, synchronization, thread-safe data structures).
* Integrated Spring JPA with Kafka and Spring Boot, utilizing Spring JPA's object-relational mapping (ORM) capabilities, Kafka's model for decoupling applications, and Spring Boot's automated configuration and setup.
* Utilized Java 11 features such as Local-Variable Type Inference, Sealed Classes, New String Methods, etc. Demonstrated proficiency in RDBMS concepts, SQL queries, and database caching strategies.
* Designed and developed RESTful APIs using Spring Boot framework to facilitate seamless integration between financial applications and client systems, adhering to industry standards and best practices.
* Implemented continuous integration and delivery (CI/CD) pipelines using Jenkins and Azure DevOps. Utilized Docker, Spring Boot, JBOSS, Azure, and Cassandra in various development projects.
* Implemented automated testing for web applications using Selenium and Spring Boot and Used Mockito's mocking and stubbing capabilities to isolate dependencies in tests.
* Deployed and managed AWS services such as EC2, S3, RDS, Lambda, and ECS for hosting and scaling trading platforms, risk management systems, and regulatory compliance applications.
* Implemented security best practices and compliance controls on AWS, including encryption, identity and access management (IAM), and network security, to protect sensitive financial data.
* Utilized Spring Security to implement authentication and authorization for microservices, ensuring regulatory compliance and data privacy and Configured and deployed web applications on JBoss, and Tomcat.
* Developed a distributed systems architecture using Spring Cloud's Netflix stack, incorporating components like Eureka, Ribbon, and Hystrix.
* Leveraged DynamoDB's auto-scaling capabilities and global tables to seamlessly handle fluctuations in workload and ensure high availability and fault tolerance for financial services across different regions.
* Implemented a microservices-based architecture using Spring Boot, Docker, and Kubernetes on AWS, with automated testing using JUnit and Selenium deployed EAR and JAR files using Jenkins and Maven.
* Utilized Spring Boot's auto-configuration and starter POMs to streamline application development Conducted performance testing with JMeter and applied Hibernate optimization techniques.

**TECHNICAL SKILL:**

|  |  |
| --- | --- |
| Languages | Java, Python, JS, SQL, PL/SQL, C |
| Java/J2EE Technologies | Servlets, JSP, JDBC, JSF, JavaBeans, MVC, ANT, MAVEN, Log4j, XML, JAXP, DOM, JNDI, JUNIT, Web services, Multithreading, JTA, Custom Tags, Java 6,7,8,11,17. |
| Web Technologies | HTML, JavaScript, PHP, Node.js, React.js, CSS, DOJO, XML, (SOAP, REST), Angular. |
| Frameworks and IDE’s | Struts, Spring version 4,5, Hibernate, JPA, Spring MVC, Spring AOP, Spring Boot. |
| Web/Application Servers | Web Logic, Web Sphere, Apache Tomcat., Jetty, JBoss. |
| Other tools | Jenkins, Docker, Kubernetes, AWS, Azure, GCP, Agile, GIT, Bitbucket, SOAP, REST, Selenium, Cucumber, Eclipse, NetBeans, IntelliJ, STS |
| Database Environments | Oracle 10g/9i, SQL Server, H2, MySQL, PostgreSQL, MongoDB, Cosmos DB, Cassandra. |
| Operating Systems | Windows, (Mac & Linux) UNIX |

**EDUCATION DETAILS**

* Bachelor of Technology from Universal College of Engineering &Technology, India (2012).

**WORK EXPERIENCE DETAILS**

**Title: Lead Java Developer ||Wells Fargo, New Jersey, NJ. Feb 2022 – Current**

Responsibilities:

* Implemented Spring Boot microservices for core mortgage processing functions, including application intake, credit checks, document verification, and automated underwriting, achieving a modular and scalable system.
* Acquired extensive knowledge of the Spring framework, including Spring AOP, Spring core, Spring JDBC, Spring batch, and Spring MVC, as well as dependency injection and legacy application migration.
* Developed scalable and fault-tolerant microservices using Kafka's distributed architecture and replication, Spring Boot's lightweight and modular design, and Spring JPA's persistence capabilities.
* Developed and maintained data migration processes using MySQL data import/export utilities and ETL tools, ensuring smooth transitions between different database environments and formats.
* Utilized Java Streams API to perform complex data processing tasks, including filtering, mapping, and reducing large data sets efficiently and with concise code.
* Designed and developed RESTful APIs using Java with Spring Boot, applying principles of SOLID design and best practices for creating maintainable and scalable service architectures.
* Applied design patterns like Circuit Breaker, Fallback, and Bulkhead to ensure resilience and fault tolerance in microservices and implemented event-driven architecture (EDA) using Kafka and Spring Boot, featuring real-time event processing and reaction, loose coupling, and high scalability.
* Configured Kafka brokers, topics, and partitions to handle high-throughput mortgage application events, ensuring efficient data streaming and real-time processing across microservices.
* Developed and maintained robust ETL pipelines using Python, facilitating data extraction from multiple sources, transformation, and loading into centralized data warehouses Redshift and Snowflake.
* Automated data workflows and batch processes using Python scripts integrated with scheduling tools like Apache Airflow or AWS Step Functions.
* Implemented ETL monitoring and logging solutions using custom Python logging frameworks and integrated with cloud services (AWS CloudWatch, Azure Monitor) for real-time alerts.
* Designed and implemented a distributed caching layer using Spring Boot’s Cache Abstraction and an in-memory data store like Redis, significantly reducing database load and improving application performance.
* Developed RESTful APIs with Spring Boot for service-to-service communication and exposed secure endpoints for external integrations with credit bureaus, appraisal services, and regulatory agencies.
* Implemented Spring Security and OAuth2 for robust authentication and authorization mechanisms, protecting sensitive mortgage application data and ensuring secure access control across microservices.
* Utilized MySQL’s backup and recovery tools to create and manage database backups, implementing strategies for point-in-time recovery and disaster recovery planning.
* Designed and implemented ETL pipelines using Python to extract data from multiple sources, transform it through data cleaning, normalization, and enrichment processes, and load it into target databases, ensuring data integrity and consistency.
* Leveraged Kafka Streams API to build stream processing applications for real-time analytics and transformations of mortgage application data, improving decision-making and operational efficiency.
* Utilized Docker to create container images for Spring Boot microservices, ensuring consistent and isolated execution environments across development, testing, and production stages.
* Developed Kafka producers and consumers within Spring Boot microservices to enable real-time event-driven communication and processing of mortgage application data, enhancing responsiveness and system efficiency.
* Integrated Spring Boot Actuator for monitoring and managing microservices health and metrics, allowing for proactive maintenance and performance optimization.
* Implemented CI/CD pipelines using Jenkins or GitHub Actions to automate the build, test, and deployment processes of microservices, enhancing development agility and reducing time-to-market.
* Designed and maintained a comprehensive logging and monitoring solution using ELK stack (Elasticsearch, Logstash, Kibana) , providing deep insights into system behavior and troubleshooting capabilities.
* Developed scalable and high-performance applications using Java 11, leveraging its latest features such as local-variable type inference and enhanced APIs to write clean and efficient code.
* Implemented multi-threading and concurrency solutions in Java to enhance application performance and responsiveness, utilizing Java’s ExecutorService, ForkJoinPool, and other concurrency utilities.
* Implemented custom exception handling and logging frameworks using Java’s built-in logging API and third-party libraries like SLF4J and Logback, improving application observability and error tracking.
* Leveraged Java Collections Framework to design and implement efficient data structures and algorithms, optimizing application performance and resource utilization.
* Engineered a MySQL database schema to support complex loan servicing workflows, including payment schedules, account management, and delinquency tracking.
* Utilized MongoDB for flexible storage of loan-related documents and records, improving accessibility and management of borrower information.
* Implemented infrastructure as code (IaC) using tools like Terraform and AWS CloudFormation to provision Dockerized applications on AWS ECS (Elastic Container Service) and EKS (Elastic Kubernetes Service).
* Developed a centralized API Gateway using Spring Boot, serving as the entry point for external requests and routing to respective microservices, and implemented API security using Spring Boot's security features.
* Utilized Spring Boot's automated testing features, such as @SpringBootTest and @TestPropertySource, to write efficient unit tests and implemented JUnit tests for Spring Boot applications.
* AGILE methodology for efficient project management, including requirements gathering, analysis, design, development, and testing, and Implemented data access layers using Spring Data JPA and Hibernate.
* Integrated AWS services with third-party financial systems and APIs using AWS SDKs and RESTful APIs, enabling seamless data exchange and interoperability between different platforms and applications.
* Designed and implemented microservices architecture using Spring Cloud components (e.g., Eureka for service discovery, Zuul for API gateway, Ribbon for client-side load balancing) to build scalable and resilient financial systems capable of handling varying workloads and traffic patterns.
* Implemented automated testing for RESTful APIs using Spring Boot's REST client and Selenium and Used JUnit's assertions and assumptions to write robust test cases.
* Implemented monitoring and logging solutions using Prometheus, Grafana, and AWS CloudWatch to monitor the health and performance of Docker containers and Kubernetes clusters.
* Streamlined CRUD operations by integrating MongoDB with Spring Data MongoDB and utilized MongoDB to store and manage non-relational data efficiently.
* Leveraged Apache Spark for batch processing and data analytics tasks, optimizing performance and scalability for processing terabytes of historical data.
* Utilized Spring Boot’s @KafkaListener annotations to build asynchronous message processing capabilities, ensuring timely handling of application statuses and updates.
* Integrated Apache Kafka with Spring Boot applications to achieve event-driven architecture for seamless communication and data synchronization across distributed systems.
* Implemented disaster recovery and business continuity solutions on AWS, utilizing services such as AWS Backup and AWS Disaster Recovery to ensure resilience and data integrity in the event of system failures.
* Implemented service discovery and load balancing using Netflix Eureka and Ribbon to achieve high availability.
* Integrated distributed tracing with Spring Cloud Sleuth and Zipkin to troubleshoot performance bottlenecks and monitor request flows, and Optimized query performance and data processing using MongoDB indexes.
* Conducted load testing with tools like Apache JMeter to evaluate API performance under various traffic conditions and optimize scalability. Developed the ability to unit test Java 17 applications using JUnit or TestNG and followed Agile/Scrum methodology to ensure effective project management and delivery.
* Became proficient in Jenkins, Git, and Maven for continuous integration and version control, along with JIRA Issue Tracker for efficient project tracking and collaboration.

**Environment:** Java 17, JSP, Core Java, Tomcat, Servlets, Spring, Angular 8/11, Hibernate, XML Web service, SOAP, JAX-WS, Unix, XML, XSLT, Eclipse IDE, AJAX, JavaScript, JDBC, JUnit4, Log4j, SQL, PL/SQL, MongoDB, JSON, Elastic Search, H2 and Oracle Database, AWS lambda, Junit, Mockito, UNIX, WebSphere, Jetty, Jenkins, GIT.

**Title: Sr Java Developer || TD Bank, New Jersey, NJ Nov 2020 – Jan 2022**

Responsibilities:

* Developed reactive, non-blocking API endpoints using Spring Boot's Web Flux and Java 8's Reactive Streams and Utilized Java 8's Optional and Spring Boot's @Nullable annotations to handle nullability.
* Automated infrastructure provisioning and configuration management using Terraform and Ansible, enabling infrastructure as code (IaC) practices for reproducible and scalable deployments on GCP.
* Integrated Tomcat with various databases such as Oracle to store and retrieve data for web applications and Implemented Kafka Streams to trigger real-time alerts and notifications to customers.
* Implemented OAuth 2.0 authentication and JWT-based authorization mechanisms within the API gateway to ensure secure communication and access control across microservices.
* Developed RESTful APIs for automated generation and submission of regulatory compliance reports, ensuring adherence to financial regulations and standards without manual intervention.
* Leveraged Docker containers and Kubernetes orchestration for deploying and managing microservices, enabling seamless scaling and resource utilization optimization.
* Utilized Kafka Streams' declarative API to build and deploy stream processing applications, leveraging its concise and expressive syntax, and Utilized various debugging tools and techniques to troubleshoot.
* Implemented authentication and authorization mechanisms using OAuth 2.0 and JSON Web Tokens (JWT) to ensure secure API access. Implemented Java 17's new concurrency features, such as the Flow API.
* Leveraged Spring Data JPA to simplify the implementation of data access layers, reducing boilerplate code.
* Implemented CI/CD pipelines using tools GitLab to automate testing, deployment, and rollback processes for microservices updates, ensuring rapid and reliable delivery of new features.
* Automated data ingestion workflows using Python and libraries like Pandas and SQLAlchemy, optimizing the ETL process for handling large datasets efficiently, reducing manual intervention, and improving processing speed.
* Used Kafka to orchestrate and optimize workflows in payment processing systems, reducing transaction times, and improving the customer experience in digital payment platforms.
* Implemented parameterized tests and test suites using JUnit to automate testing of multiple input scenarios, improving test coverage and identifying edge cases in the codebase.
* Designed and implemented design patterns such as Singleton, Factory, Observer, and Strategy to address common software design problems and improve application architecture.
* Utilized Java Reflection API to build dynamic and flexible components, enabling runtime inspection and modification of classes and methods.
* Developed and maintained microservices using Java 11 and Spring Boot, implementing RESTful APIs to support scalable and modular application architecture.
* Designed and implemented Kafka-based messaging systems for real-time data processing and event-driven communication between microservices, enhancing system responsiveness and scalability.
* Utilized Spring Boot’s microservices capabilities to create and deploy independent services, facilitating seamless integration and communication through standardized REST APIs and Kafka topics.
* Implemented Kafka producers and consumers within Spring Boot microservices to handle asynchronous messaging and ensure reliable event processing across distributed systems.
* Applied Java’s concurrency utilities to develop thread-safe and synchronized components, ensuring data consistency and integrity in multi-threaded applications.
* Optimized application performance through Java profiling and tuning techniques, including JVM garbage collection settings, heap management, and performance monitoring.
* Automated key financial workflows using Hibernate and JPA, including loan processing, interest calculation, and account reconciliation, enhancing operational efficiency, and reducing manual errors.
* Developed automated test suites using JUnit and Mockito for unit testing and integration testing of Spring Boot applications, improving code quality and reliability of deployments on GCP.
* Implemented Kafka Connect to integrate disparate banking systems and databases, enabling unified data streams for customer relationship management, risk assessment, and operational efficiency.
* Implemented CI/CD pipelines to automate the build, testing, and deployment of Java web applications on Tomcat and Gained strong expertise in Data-Driven Automation Testing using tools like Selenium WebDriver.
* Gained experience in Exception Handling and deployment of applications using EAR, WAR, and JAR files.
* Became familiar with build tools like Maven, application servers like WebLogic and Apache Tomcat, and version control systems like Git and GitHub.
* Designed and implemented RESTful APIs with Spring Boot, using controllers and service layers to handle complex business logic and enable seamless client-server communication.
* Utilized Spring Boot’s dependency injection and inversion of control to create maintainable and testable components, enhancing code modularity and facilitating easier unit and integration testing.
* Implemented Spring Boot’s data access layer using Spring Data JPA/Hibernate, creating efficient repository interfaces for managing relational database interactions and optimizing query performance.
* Configured and managed application properties with Spring Boot’s externalized configuration support, using profiles and configuration files to handle different environments and settings.
* Developed microservices with Spring Boot and Spring Cloud, utilizing service discovery with Eureka, API gateways with Zuul or Spring Cloud Gateway, and distributed configuration with Spring Cloud Config.
* Implemented Docker-based containerization for Spring Boot microservices, facilitating consistency and portability of applications across development, testing, and production environments on Google Cloud Platform (GCP) and proficient in Apache JMeter for performance testing and Log4J for application logging.

**Environment:** Java 11, Java 8 Spring, Hibernate, Web Services, REST, WSDL, SOAP, XML, XSD, AJAX, JSON, Oracle 11g, Oracle SQL developer, JBOSS Application server, Eclipse, Maven, Mockito, Jasmine/Karma, Power Mock, Selenium, Tortoise SVN, Git, JIRA, Splunk, SonarQube/SonarLint, Azure, Docker, Postman, Kafka, Jenkins.

**Title: Sr Java Developer || Siemens Healthineers, Atlanta, GA. Jul 2019 – Oct 2020**

**Responsibilities:**

* Worked with Agile methodology (SCRUM), practicing Test-driven development (TDD), and Pair Programming for efficient and collaborative software development.
* Created deployment packages and used AWS SAM or the Serverless Framework to manage Lambda deployments and Utilized Spring Boot's testing frameworks like JUnit to write unit and integration tests.
* Optimized API gateway configurations and microservices interactions to enhance performance and reduce latency, resulting in improved response times for critical financial transactions.
* Implemented RESTful APIs with Spring MVC for seamless integration between microservices, facilitating efficient communication and data exchange within the retail ecosystem.
* Leveraged Spring Cloud Netflix for service discovery, load balancing, and circuit breaking, ensuring fault tolerance and resilience in distributed systems.
* Successfully migrated data from existing data sources to MongoDB, ensuring seamless data transfer and compatibility with the new data storage solution.
* Conducted code reviews and pair programming sessions to promote TDD and BDD best practices among team members, fostering a culture of collaboration and continuous improvement.
* Deployed microservices on AWS Elastic Container Service (ECS) and AWS Lambda for containerized and serverless execution, optimizing resource utilization and reducing operational overhead.
* Utilized Docker to containerize Spring Boot microservices, ensuring consistent deployment and scaling across different environments, and managed orchestration with Kubernetes for automated scaling and load balancing.
* Integrated Spring Boot microservices with external APIs and databases, using Spring Data JPA or MongoDB to manage data persistence and provide seamless access to backend resources.
* Developed and applied custom Spring Boot starters and auto-configuration to simplify setup and configuration for common microservice patterns and requirements, enhancing code reusability and maintainability.
* Implemented health checks and metrics collection with Spring Boot Actuator, enabling proactive monitoring and alerting for microservices’ health and performance status.
* Collaborated with DevOps teams to build and maintain CI/CD pipelines for Spring Boot microservices using Jenkins, GitHub Actions, or GitLab CI, automating deployment and ensuring continuous integration.
* Implemented inter-service communication with Spring Boot and RabbitMQ, facilitating asynchronous messaging and event-driven architecture to handle high-throughput data processing.
* Configured and managed distributed tracing and monitoring for microservices using Spring Boot Actuator, integrating with tools like Zipkin, Sleuth, or Prometheus to track and analyze service interactions.
* Applied Spring Security with OAuth2 and JWT for securing microservices, implementing authentication and authorization mechanisms to protect service endpoints and ensure secure access control.
* Integrated with AWS services such as Amazon DynamoDB and Amazon RDS for data storage and retrieval, ensuring data consistency and durability in high-traffic retail environments.
* Utilized lambda expressions and functional interfaces to simplify asynchronous programming tasks, enhancing concurrency and responsiveness in multi-threaded applications.
* Leveraged Java 8 streams API for efficient processing of large datasets, reducing boilerplate code and enhancing productivity in data manipulation and transformation tasks.
* Utilized Java’s CompletableFuture API for asynchronous programming and parallel data processing, enhancing responsiveness and allowing for non-blocking operations.
* Implemented Java’s Future and CountDownLatch mechanisms for coordinating asynchronous tasks and managing complex workflows in multi-threaded environments.
* Implemented event-driven architecture using Amazon SNS for real-time processing of customer orders, inventory updates, and shipping notifications, enhancing the responsiveness of the retail platform.
* Configured monitoring and logging using AWS CloudWatch and AWS X-Ray for visibility into application performance and troubleshooting of production issues, ensuring high availability and reliability.
* Utilized Kafka's connectors and Spring Boot's integration features to connect with external systems, such as databases and messaging platforms. Implemented dependency injection and AOP using Spring Boot's IoC container, Optimized database performance using Oracle's performance tuning tools and techniques.
* Utilized Oracle's advanced features, such as partitioning, clustering, and replication, to improve performance and optimize complex SQL queries, stored procedures, and functions using Oracle's.
* Utilized AWS Code Pipeline to automate build, test, and deployment processes for Dockerized applications and implemented continuous integration and delivery (CI/CD) pipelines using Git, Docker, and Kubernetes on AWS and Used SVN (Subversion) for version control and source code management.

**Environment:** Java 8, JSP, Servlets, Spring, Microservices, Hibernate 3, AJAX, JavaScript, XML Spring Boot, Web service, SOAP/REST, JAX-WS, Spring batch, XSLT, Eclipse IDE, JUnit 3.8, Log4j, SQL, PL/SQL, HTML5, JSON, Elastic Search, H2 and Oracle Database, Mockito, Grunt, UNIX, WebLogic, JMeter, AWS, Splunk, Redis, Jenkins1.6/2.0.

**Title: Full Stack Developer || State of Delaware, Wilmington DE. Nov 2017 – Jun 2019**

Responsibilities:

* Strong knowledge of Core Java concepts including Generics, Collections, Exception Handling, Multi-Threading, and Serialization. Integrated MongoDB for efficient non-relational data storage.
* Experienced in applying design patterns like Singleton for creating efficient and reusable code structures.
* Skilled in Spring Web MVC and Spring Web Flow for building robust and scalable web applications in the Laboratory Equipment Biotechnology Pharmaceutical Healthcare domain.
* Proficient in Spring IOC/Dependency Injection and Spring AOP for implementing modular and loosely coupled application architectures. Experienced in using Jira for project management and issue tracking.
* Skilled in Hibernate ORM for object-relational mapping and database integration in healthcare-related projects and Knowledgeable in using Confluence for collaborative documentation and knowledge sharing.
* Employed servlet-based caching strategies, including in-memory caching with EHCache and distributed caching with Redis, to improve application performance and reduce database load.
* Developed and maintained detailed servlet documentation, including API documentation with tools like Swagger and Javadoc for comprehensive project documentation.
* Enhanced Struts action execution by implementing pre-action and post-action logic using interceptors, allowing for reusable and modular code components.
* Skilled in Browser Compatibility Testing to ensure seamless user experience across different web browsers.

**Environment:** JAVA 1.8, J2EE, Core Java, J2EE, JSP, Spring 4.0 Hibernate ORM 4.3, EJB-Session Beans, Entity Beans, JMS, XML, JDBC, Log4J, CVS, GIT/stash, Drupal, Maven, Jenkins, Jira.

**Title: Java Developer || ITC Infotech, India. July 2012 – Aug 2017**

Responsibilities:

* Strong knowledge of Core Java concepts including Generics, Collections, Exception Handling, Multi-Threading, and Serialization. Utilized Spring AOP's @AspectJ annotation to define and apply aspects.
* Utilized Core Java coding and Java APIs such as Collections, Multithreading, Exception Handling, and Java I/O to implement business logic. Experienced in using Jira for project management and issue tracking.
* Implemented asynchronous servlet processing using AsyncContext to improve scalability and performance, and leveraged non-blocking I/O (NIO) capabilities for handling large volumes of concurrent client requests.
* Developed servlet filters for request/response filtering and modification, such as logging, authentication, and compression, utilizing FilterChain to chain multiple filters and create reusable processing components.
* Configured servlet listeners (e.g., ServletContextListener, HttpSessionListener) for monitoring application events, and managing application startup and shutdown routines using context initialization.
* Utilized Struts Tiles for reusable layout definitions and consistent look and feel across web pages, configuring Tiles definitions in tiles.xml for dynamic composition of views.
* Developed custom Struts interceptors for cross-cutting concerns like logging, security, and transaction management, and implemented custom result types for handling various response formats (e.g., JSON, XML).

**Environment:** JAVA 1.8, J2EE, Core Java, J2EE, JSP, Spring 4.0 Hibernate ORM 4.3, EJB-Session Beans, Entity Beans, JMS, XML, JDBC, Log4J, CVS, GIT/stash, Drupal, Maven, Jenkins, Jira, JBoss, Net Beans IDE.