**GEORGE CURINGTON**

**Montclair, New Jersey**

**908-612-4531**

**PROFESSIONAL SUMMARY**

* Senior Core Java developer.
* Proven technical design and programming skills spanning over 15 years of programming experience within several different team-oriented environments.

**Education**

* Temple University, Philadelphia, PA - Bachelor Degree in Computer Science
* University of Illinois, Champaign Urbana, ILL - Liberal Arts Major, Music Minor

**TECHNICAL KNOWLEDGE SUMMARY**

**Main Focus Area**

* Object Oriented Design
* Core Java/Server Side Development
* Linux
* Design patterns
* Generics and Collections,Lambda Functional Programming
* Java Concurrency(multi-threading)
* JSON, YAML
* JaxB Eclipse Plugin, Soap Webservices
* JDK 6,7,8
* Eclipse IDE
* Jersey V2 RESTful Service using JAX-RS, Jboss RestEasy
* Core Spring Framework, JPA, Hibernate, Spring Boot, Spring Data, Spring Profiles, Transaction Management, Spring Rest Client and Services

**Areas of special interest**

* Scala Programming imperative/functional / Scalatest
* AKKA Actors using Scala
* JDK 8 Functional Programming
* Algorithms/Data structures
* MongoDb, Robomongo 3T, ElasticSearch

**Software Configuration Area / Testing**

* Git/Bitbucket/Github
* Maven
* Gradle ( with Groovy )
* Jenkins Continuous Integration
* Junit
* JFrog Artifactory, Nexus

**Frameworks/Middleware and other API**

* JDBC
* ActiveMQ ( JMS )
* Tomcat Servlet Engine , JBoss/Wildfly, WebLogic
* Swagger
* Elastic Search, Kibana
* Redis-Distributed Data Grid

**Database**

* Oracle/MySQL/MSSQL
* MongoDb 3.x

**Operating System and Scripting Languages**

* Linux ( fluent in Shell, Perl, Awk, Sed and most Linux command tools )
* Scala Functional Scripts

**Currently Learning**

* Docker/Openshift
* RabbitMQ
* Junit tests with Mockito
* Development using Java on Mac Os X

**Legacy Knowledge No Longer Used**

* Sybase, C, C++, Ant, Subversion, Cvs

**PROFESSIONAL EXPERIENCE**

**United Health Group**

**Jul 2017 – Present**

**Consultant/Core Java Developer**

**Group Mentor**

Helped and taught other members of my team how to set up and configure MongoDb , Mongo Connector and Elastic Search. Also helped members understand Linux commands.

**ElasticSearch POC**

Connected MongoDB to ElasticSearch using Mongo-Connector to allow updates to mongo to be seamlessly applied to Elastic search engine. Served as lead and mentor to other developers on this effort.

**SpringBoot/Swagger**

Created Swagger based api using SpringBoot to apply Elastic DSL queries to the ElasticSearch engine.

**Spring/JAXB Webservices utility**

Created a generic utility that allowed java clients to execute SOAP calls without the need to use the underlying JAXB calls. This utilized Spring's JAXB plugin.

**Redis-Distributed Data Grid**

Implemented POC for Redis vesion 3 using Radis Sentinel, Slaves and Master Configuration. Demonstrated use of Redisson java Framework in a clustered environment for failover and replication of data.

**Mulesoft/Anypoint ESB**

Responsible for creating various connectors within the Mulesoft ESB framework. Initial project is to implement a data aggregator for generating files from split incoming data streams.

**Bank of America**

**Apr 2015 – Apr 2017**

**Consultant/Core Java Developer**

**Responsible for java development / Database as a Service / Cloud Services.**

**Key accomplishments:**

**Jenkins-JFrog Artifactory Automation**

Created automated deployment tool used in conjunction with Jenkins CI and JFrog to check for check-sum changes (md5, sha1) in deployed artifacts on linux servers and automatically deploy war files into Tomcat application containers. This was written in Java utilizing executor scheduling service, Jersey Restful services and JSON processing.

**Data Layer Abstraction**

Created a Spring-based data layer abstraction that allowed Angular-2 ui components to interact with SQLsevers and MySql servers using Jersey Rest as the underlying communication layer.

**Jersey Rest Services Abstraction**

Created a JSON-based configurable framework that allowed clients of Jersey Restful Services to send to and receive from Rest Resources with minimal coding. All configuration is encapsulated in JSON files.

**Scala/Akka Smoke Testing**

Created scala application to spray our jersey java-based restful servers with request as a means of stress testing the web applications. This was a multi-threaded application that used AKKA to send multiple simultaneous requests to Restful Services endpoints.

**Intermediate Level Scala Consulting**

Gain further knowledge of Scala and started to impart that knowledge to fellow co-developers in my group and other groups on my current assignment.

**Jersey Restful Services**

Creating Jersey Restful Services to implement services required to create Virtual Machines containing Database Servers. The services make use of a sophisticated loosely coupled framework distributed over Linux machines. Hazelcast distributed cache management system is used for the bulk of database access. All inter process communication is via Jersey Restful Services. A thin web layer also utilizes direct database access using Spring JPA-DATA within the Jersey Restful Services. Gradle, Groovy, Svn, Git, Maven and Jenkins are used in the build and deploy aspects of the system.

**Frameworks**

Creating a data driven framework that allows Jersey Restful clients to automatically call service endpoints requiring minimal programming. Endpoints are configured via configuration files allowing the framework to generically process the call and the response.

**Team Mentoring**

Helping other developers in areas involving multi-threading, Unix scripting and Design Pattern usage.

**Research**

Apache Spark, Scala , Jersey Rest, Datastructures, Search Technology, multithreading

**Bank of America**

**Feb 2011 – Jun 2014**

**AVP / Core Java Developer for Bank of America**

* Responsible for helping the development team carry out regression analysis, system debugging, Autosys monitoring, java development and minor Perl development.

**Key accomplishments:**

* Note: All accomplishments utilize pure java concurrent facilities of the java.util.concurrent package.
* **Multi-threading.**
* Due to inter-continental lag time in the ftp process, created a highly flexible multi-threaded ftp-client for Bank of America’s business process between their main Risk Analysis partner (Risk Matrix Group). This ftp-client successfully allowed Bank of America’s business of ftping millions of files between east and west coast in the time required by the SLA (Service level agreement). Prior to this implementation, the time required to process all the files was failing the SLA daily. This project made heavy use of Java’s concurrent API.
* **Multi-threading.**
* Converted a legacy Securities database refresh application to a multi-threaded application gaining about a 700% increase in throughput on a multi-core Linux system.
* **Multi-threading.**
* Created a multi-threaded regression tool that automated a previously manual process of executing SQL queries against different database environments and comparing the results; the resultant output was in the form of an Html page and Excel spreadsheet. This tool reads an XML document containing multiple SQL expressions, database config info and other required configuration and uses threads to process the SQL expressions against multiple databases, comparing the results and placing the output on HTML and Excel forms. This is then automatically emailed to Managers, Quality Assurance and Developers every configurable time minutes.
* **Multi-threading.**
* Created a multi-threaded log file analyzer utilizing regular expressions. This tool uncovered several important faults in the client’s processing environments.
* **Multi-threading.**
* Converted several long running Autosys jobs to multi-threaded java applications achieving close to 1000% increase in thru-put.
* **Password Encryption.**
* Enhanced WebLogic GUI, the main portal to the back end Risk Analysis Engine, to conform to company compliance initiative to encrypt all user passwords. This was done using one-way encryption; sha family of algorithms utilizing randomized salt and variable hash iterations. The underlying API utilized was Java’s message digest.
* **Program Analysis:**
* Analyzed client applications for non-thread-safe constructs and was able to fix long standing bugs that were causing various important race conditions.
* **Mentor:**
* Taught other developers how to create multi-threaded applications using Java’s new concurrent classes. Acted as a java resource for junior developers.

**IBM (AT & T), Middletown New Jersey,**

**Sep 2010 – Feb 2011**

**Consultant/Java Developer**

* Responsible for developing back end services for a Spring Framework based Front End web application.
* Create a service layer interface for the GUI developers. The service layer was implemented based on the DAO idiom using Spring Framework’s IOC/AOP to help both in maintaining a loose-coupling of components and to wrap the service layer with a Transaction manager. Data access was implemented mostly using Hibernate. Pure JDBC was used to implement mapping utilities. Additional responsibilities included moving low level java-based Factories into a Spring Framework managed environment.

**Johnson & Johnson Pharmaceutical Research & Development, Raritan New Jersey**

**May 2010 – Sept 2010**

**Consultant/Java Developer**

* Responsible for developing back end services within a web services stack running inside a managed Weblogic server. The services utilized Oracle/Hibernate for persistence. My main responsibility was to implement various transactions that the business layer dictated. These transactions can be described as vanilla processing of requests sent in from a web GUI. The basic design was that of a state diagram. Transactions are implemented by the methods within the web services stack. Each time a transaction is processed, it moves to another state. The GUI itself, not my responsibility, was generated using an in-house content management system that converted XLS spread sheets to Struts enabled JSP pages.

**Free Lance Core Java and Web Development**

**Jul 2009 – Present**

* Developing and maintaining a small e-commerce web site to sell various art related items.
* Developing and maintaining a stock analysis application used to predict possible buy notifications of stocks. This application uses financial data including p/e ratio, percentage increase, volume, 52wk data, etc. to determine when to buy a particular stock.
* Developing a small programmer help site to illustrate the usage of various design patterns and to compile a list of descriptions and solutions for various Core Java, Spring, Hibernate3, Generic and threading related questions.

**New York Client, Stone Castle Partners LLC**

**Mar 2009 – Jul 2009**

**Consultant**

* Responsible for creating and maintaining a Java Swing application that is used to stream data from a MySQL database and populate a Swing Jtable. The client requires easy access to any table in the database. There is also a requirement to automatically jump to any point in a table without loading the full table into memory. This is a generic interface to Database tables. Heavy OO Design Pattern Usage, Eclipse, JDBC, SQL, XML and Ant.
* UI allows the user to input multiple restrictions on all columns including the use of Regular Expressions. The resultant SQL query is generated using a Decorator Design pattern. Subject/Observer pattern utilized heavily within the Swing API.
* JSlider, JScrollBar and keyboard pagedown/up commands all control data acquisition, executing Database retrieval only when the current buffer of data is processed

**Citigroup, Warren, New Jersey**

**Mar 2003 – Dec 2008**

**Senior Programmer Analyst**

This job includes various aspects web programming. The technology includes Javascript, Java Server Pages, JSTL, Spring Framework, Hibernate, XML processing, Sybase, Sybase-Perl, Unix Scripting and Java Bean programming. The core application is a web site, e-Sales, that is used to maintain account information of Citigroup's institutional and global clients. In addition to web and core java development, the system is composed hundreds of batch processes written in Unix Shell, Perl and Java.

**Main Accomplishments:**

* (Core Java, Java, Collection classes, XML) Configuration Object to enable internal objects within the system to easily acquire configuration information needed to run in a variety of environments, i.e. DR (disaster recovery), development, staging, production. Configuration information includes location of servers, encrypted passwords and a large amount of other information which enables all systems to be dynamically changed at runtime. Before this system was implemented, it took a full day to deploy a large system from Staging to Production. Now this is done with the click of a switch.
* (Core Java, JSTL, Java Beans, Hibernate, Spring) - Allowing Russian language text to be entered into eSales Account Data.
* Maintain, modify legacy C++ code.
* (JSTL, Java Beans, JDBC, Hibernate, Spring, Java XLS API POI) Bulk upload process utilizing Spring under Jrun along with JDBC. This web utility allowed the user to bulk upload data in XLS format to a Sybase data server. Created the web-based front end using Spring Framework. In addition used a standalone java server to upload the data.
* (Core Java, Apache XML Sax, Collection classes, Threading) Java new account form processing. This process converts a new account to XML format and automatically transfers the XML to an image processing down-stream system. Again, utilizing a java server that was created to process the request.
* (Core Java, Collection classes) Java approval processing system. Certain accounts are flagged as unapproved based upon various attributes of the account. For example, a user changes an account description or bank id. This system is driven by a truth table and causes an account to go unapproved if certain conditions are matched. This was implemented in such a way that the approval engine can reside anywhere in Citigroup’s internal network.
* Various data access utilities using Hibernate.
* Various web utilities/pages using spring.
* (Java threads, Xerxes XML, java, JSP, JavaScript) Scrubbing implementation used to detect illegal participants in the Accounting System. For example, certain countries and/or certain people are on the US terrorist watch list. This system is real time and utilizes outside systems to scrub data in a federal-government regulated manner.
* (Core Java, JSP, JSTL, java, javascript, Java Threading)Controller implementation. This is a MVC implementation of a servlet controller. It is an integral part of the Sales/Accounts infrastructure. It performs a growing number of functions for the Sales/Accounts team.
* Various simple to moderately complex Perl scripts used to load various types of data into Sybase.
* (Core Java, Reflection API, Java threads, Collection classes, Apache Xerces Sax and DOM parsers)
* Developed a server (AcceptHandler) that allows our web application to easily interact with outside processes and/or data. The AcceptHandler is xml-driven and allows additional services to be added without the user having any knowledge of the inner workings of the server itself. One need only drop a class that extends a base in a particular directory. You might call this server a thin implementation of an Axis Soap Engine without the requirement of running inside a servlet container. Xerces Sax parser is the underlying xml parser being used. The accept handler is capable of running in SSL mode via a simple configurable parameter (see configuration object above)
* (Core Java)Created various java-class utilities:
  + Class to transform Sybase results into a java program with getters and setters set up to easily be used in Spring and /or Hibernate. Uses DOM xml parser.
  + Class to easily use Xerces SAX xml parser.
  + Simple logging facility.
  + Utility to automatically generate XLS reports based on dynamic SQL query.
  + Generic query processor utilizing java-beans, javascript, JSP and Sybase.
  + Thin generic wrapper around JDBC allowing programmers to easily utilize JDBC without having any knowledge of JDBC. For example, with one simple call, data is encapsulated into an ArrayList of HashMaps.

**AT&T in Florham Park, NJ**

**Jul 2000 - Feb 2003**

**Consultant/Programmer/Tester**

Developed software to test AT&T’s B2B server with their many clients around the world/ created software to simulate clients.

**Main accomplishments:**

* (Java servlet, xml, JDBC, Oracle, sockets, UNIX, JSP, Perl)Developed a JAVA-SWING functional simulator that other testers used to functionally test AT&Ts webmethods B2B server.
* (Java, servlet, xml)Created a generic set of call backs that convert XML files into a THIN internal structure for later processing using Apaches Standard SAX XML parser.
* (Core Java, xml, UNIX, Perl)Created a simulation system for AT&T to test their B2B server. The system could simulate a configurable number of clients performing random actions at a configurable rate.
* Written many Perl and UNIX scripts to process various trace, log and statistics information.
* Many of my Perl and Java Simulators will continue to increase Tester and Developer productivity after my contract ends with AT&T.

**Hobbies:photography, painting, walking and studying algorithms.**