**Availity’s Backend Homework Assignment**

We highly recommend that you use one of the free source code management platforms (GitHub, GitLab or BitBucket) when storing your code. Once you are ready for us to look at your answers, send us the link to your code. If you have any questions about the homework, please do not hesitate to ask.

* Tell me about your proudest professional achievement. It can be a personal or school project.

When I was at BellSouth we had a legacy system for maintianing files called engineering wiring lists. Our license for this application was about to end and the replacement Oracle based system was not going to be implemented for over a year. The account executive came to me and asked if we could come up with an interim solution that not only allowed engineers access to the data but also provided a means to update the wiring lists to keep them current. This was mission critical data that the engineers depended on consisting of over 50000 iles in a prorrietary |(pipe) dilimited format.

My solution was to convert the data into spreadsheets with labeled pages and columns that could be loaded into a central office records management system and maintained by our central office data entry people until budgtet permitting the new software was available . There were over 28,000 drawings in 45 distince formats containing up to 30 drawing-code specific wiring diagrams. I worked closely with the account executive to create a unique xml template for each of the 45 drawing codes. I then created a java application using a freeware spreadsheet java API to parse the wiring lits and generate excel workbooks. After some trial and error my processing application generated the 28K spreadsheets with up to 30 pages in each workbook. We then loaded the spreadsheets into yhe central office records system that allowed the clerks to select and update the spreadsheets as required by BellSouth work orders. It worked fine, lived for 16 months and BellSouth enginners were satisfied to have up-to-date elictrical diagrams.

The real payoff came when the new Oracle-based system was brought online and their engineers were able to use my xml templates to drive their conversion of the spreadsheets into their Oracle databse. My team, who were all BellSouth employees won monetary awards and plaques congratulating them on the time and money saving design. Ironically, I had done all the work including design, but because I was a contractor I received no recognition. I was still happy becaus eto be honest I never thought about the xml templates being useable in the conversion to the new system. The final irony occurred when the new vendor charged $750,000 and took 3 months to convert the data. I could have done iy by myself in less than a month.

* Tell me a about a book, blog, article or GitHub repo you read or liked recently, and why you like it and why you should recommend I do the same.

I have 2 great books for implementing java microservices on SpringBoot. The better of the two is "Mastering Microservices in Java" from Packt. It provides a great blueprint for creation and administration of microservices on Spring Boot, including security, messaging, and deployment in any form including hosted and cloud-based on both VM's and containers with guidence for maintaining scalability. The book is a strong propenent of domain(model) driven design which I also belive in whole-heartedly.

* If you were to describe to a 7-year old what Availity does, what would you say?

Too a 7 year old I would describe Availity as a company that helps patients, doctors and hospitals, and insurance companies work together.

* Coding exercise: You are tasked to write a checker that validates the parentheses of a LISP code. Write a program (in Java or JavaScript) which takes in a string as an input and returns true if all the parentheses in the string are properly closed and nested.
* Coding exercise: Availity receives enrollment files from various benefits management and enrollment solutions (I.e. HR platforms, payroll platforms).  Most of these files are typically in EDI format.  However, there are some files in CSV format.  For the files in CSV format, write a program that will read the content of the file and separate enrollees by insurance company in its own file. Additionally, sort the contents of each file by last and first name (ascending).  Lastly, if there are duplicate User Ids for the same Insurance Company, then only the record with the highest version should be included. The following data points are included in the file:
* User Id (string)
* First and Last Name (string)
* Version (integer)
* Insurance Company (string)

Again, please let us know if you have any questions. Thanks!

-Availity Team