Argument for Solution Responses to BorderGuru

Matheus Pavin Pedroso
pavinmatheus@gmail.com
+55 48 99966-8528

a) Why did you pick your first particular your design? What assumptions did you make, and what tradeoffs did you consider?

I started by defining that the input and output data would be in JSON. I have not found information on its use, but I am quite sure that it is the standard in the IT industry.

So, I then proceeded, defining that I would use node.js and express. The routing would be done by the express itself (router), because it is less complicated than a UIRouter, for example, and for a small application, to be perfect.

Finally, I defined that I would use the SQLite database, since it does not need a server to run, and this would make the application easier. Here is the first tradeoff, as the SQLite database is somewhat limited compared to PostGreSQL (my first alternative), preventing larger query's and causing some things I have brought into memory and handled in the backend.

Before start coding, I set up a test environment and started testing. This makes me quieter, because in the end I can refactor the code without fear.

Do not worry about handling exceptions / edge cases. In this way, some things are functional in a controlled environment, but they may escape control in a larger environment.

One of the assumptions I made was to believe that all the company names would have no space and then capture them by the URL parameters.

And that was the end of the first half.

b) What did you consider when you had to design the second solution? And which assumptions and tradeoffs you made?

For the second solution, i was rigid with the requested "you can not edit the order structure you created in previous exercise". Thus, the biggest tradeoff was that the orders table has no relationship with the companies table, thus making it difficult to pair and include the companies.

In this way, every time some company information is requested, the system goes to the order table and verifies if there is any new company and if you see, add in the companies table.

Another tradeoff, was that companies can be inserted, differently, by minor typos. Thus, the company SuperTrader, can be inserted a second time, as SuperTarder. This in a production system would be unacceptable, however, i did not deal with extreme cases again.

At no point in the two steps did I worry about security, so i did not use any kind of access token. However, in both parts, all the services were tested in unitary form, but no integration test (routes) was done.

c) What do you like (and dislike) about Node/Javascript compared to other programming languages?

I like NodeJs, for a few reasons. I can elucidate some important ones here. A crucial one, is the agility in handling requests, the Node event loop, the non block paradigm, the way that client and server can both start the communication, is phenomenal for this purpose and if well used, withstands huge applications and is likely to be very optimized.

An important detail is that by using a Javascript framework for the front and backend, developer will feel more comfortable working with one language, which can lead to increased productivity.

Another point to consider is the use of NPM (Node package manager), because this tool is phenomenal in development, freeing the programmer from a large load. In addition, the whole deployment issue is facilitated by NodeJS, helping in day-to-day development.

Another major problem is to come from a strongly typed and object-oriented language. Although the classes have appeared in ECMAScript 2015, they still do not follow the standardization that a java would have. In this way, my first months learning and developing in Javascript, were of much pain and suffering.

Finally, after falling in love with Javascript, you end up embracing the cause and studying more and more, because there is always something very different to learn and this is very captivating. Just look at the passion with which the speakers speak, in JSConf, that you can get an idea of how passionate is the language.