Configurely App

# Git - <https://github.com/MasteRaven/Configurely>

First I would like to thank Boxplosive for this opportunity.

This test was clearly a challenge. In MVC I have never felt the need to build a dynamic form.

My first approach was to build the database and build all the tables and relations to be able to have each department with each own set of fields (from a table of respective fields).

Next I focused on building the WCF Service that served the data to the MVC. I used Rest because in my opinion the way it uses the Http protocols is way easier and faster to use.

Between the construction of the database and the WCF I created a simple project just to store the solution entities. I find it to be a good practice not to use the Entity Framework model entities. So all the entities coming from the WCF Service the EF entities converted.

After that I started to work on the MVC app. I wanted to use React to build the dynamic form. So I had a view just to have my parameters and my “div root” as my point of entry into to the React app. Every time the react need to communicate with the backend it uses fetch and send the requests over Http. At first I thought about building the totality of the App with React making it a Single Page Application, but the more I read the test the more I felt like I should make MVC my focus and have just a little demonstration of React. Since I have been working with React for almost 2 years I find my opinion biased, but I feel that the V in MVC using React is way easier, faster and lighter.

Some of the improvements that I would make would be:

* Some refactoring of the code especially in the controllers (for example: build a Core to make the WebRequests in a single place and reusable);
* The React is a little “raw”. With more time I would have used Redux to manage all the state of the app in one single place. It takes some time to implement Redux (single store, middlewares, selectors and the reducers) but in my opinion, after all is set and done, it’s so much easier and faster to develop;
* The ability to create new departments and fields;
* The frontend and user experience could have been greatly improves with some css and bootstrap;

With all its flaws and defects, I enjoyed developing it and experience areas that I have never tried before. As a developer I’m always seeking new knowledge and experience so I would say objective done.

Thank you very much.