JS Text Exercise Documentation

Overview

The purpose of the test is to demonstrate good architectural design. The main 2 aspects of such design will be:

- Modularisation: Ability to reuse the module(s).
- Scalability: extend the functionality of the module(s).

Constraints

- Browser support is latest Chrome or Firefox.
- No full frameworks such as angular or backbone are permitted although libraries such as jQuery, templating tools, AMD etc. are all acceptable.
- Exactly replicating the design pictured above isn't necessary.
- A written description and/or comments in your code explaining the rationalisation behind your architectural decisions, and what you would have done with more time, is acceptable.

Extra points

- Use of TDD/BDD.
- Storing values in Local storage so that they persist on reload.
- Demonstrating your knowledge of currently trending tools, best practices and design patterns.

Background

I've been developing web based solutions for over 15 yrs. I've created many JS based frameworks and worked at Sencha as an engineer and senior technical trainer on their key product which is ExtJS.

Considering the amount of time I had for this particular challenge, I tackled it based on what I felt was most important:

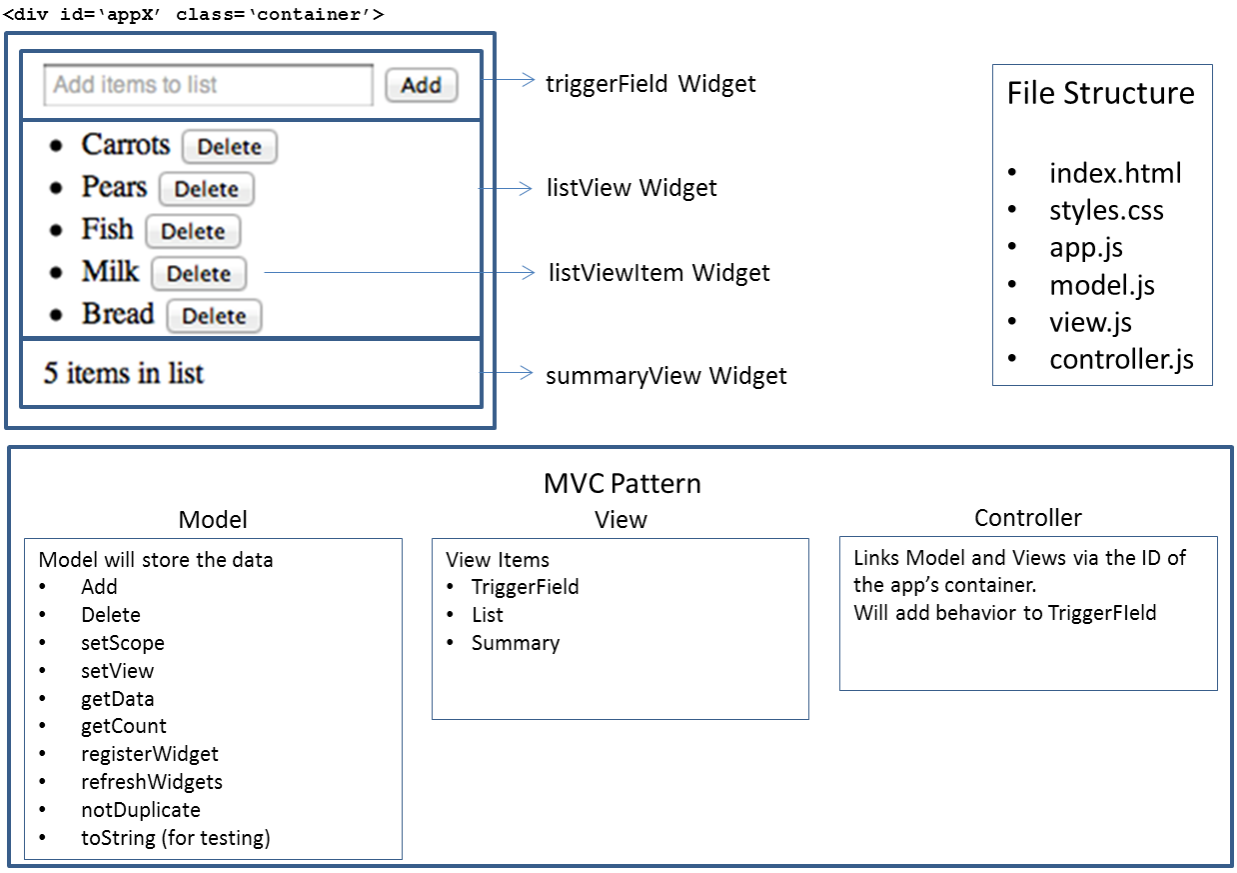
1. Modular Reusability from the ground up
2. Demonstrating how easy it is to extend the work

As such, I used an MVC like approach in the design and development of this project.

I tackled the project as follows:

1. design diagram (schematics) (30 minutes, this is the app architecture seen in this document)
2. index.html (5 minutes)
3. app.js (throughout the development of the app, using it to quickly test in-browser)
4. model.js (10 minutes)
5. view.js (30 minutes)
6. styles.css (30 minutes)
7. controller.js (30 minutes)
8. This document was written in a ongoing manner during development

Diagram



Quick Overview

- View will provide the HTML for all widgets as well as the refresh method for views which will be bound to the model.
- Model will be responsible for calling each view's refresh method when data is added or deleted.
- The controller will provide the link between view and model and will provide behavior to Add and Delete buttons.

Potential Improvement

There are many things that can be done from this work in order to improve upon it. The list is far from exhaustive by any means.

1. namespacing the entire project as well as the various APIs into some form of a JS framework name
2. loading data from a data storage
3. adding unit test and functional test
4. improving on the API provided, the techniques for adding html and refreshing views are very rudimentary
5. separate the View file into widget files and have each widget file associated with a CSS file.
6. using build tool like Gulp or Grunt, etc.. to build the application from the ground up to ensure a single JS file and a single CSS file which is context sensitive to the application needs.
7. Look and feel of the app..
8. Usability, setting focus, adding Enter key for keyboard event on textfield, etc..
9. Code optimization

Conclusion

Considering the many JS based framework out there, we rarely see projects where an entire web application is based on a DIY (Do It Yourself) JS framework. Typically, Angular JS, jQuery, React JS, as well as other templating and UI libraries will be used, or the use of framework like ExtJS which is a one-stop framework with all the necessary architectural design patterns and UI widgets will be used.

So, this exercise, and the time constraint I had to deal with, was to demonstrate one of the key critical aspect of being a Front-End Web Architect, which is how to architect reusable and scalable solutions with the least amount of code and effort in a clear and efficient manner.

Documentation and up-front preparation are key to a successful implementation, which is why I've put much energy towards the building of the document, because in this line of work, not only must we be clever in how we perform our task, but we must also be able to clearly document and inform those around us.

Claude Gauthier
(510) 393 0175