

## Frontend Challenge Problem

As a frontend engineer you should have working knowledge of Javascript, HTML, CSS fundamentals and frameworks. The following specification reflects a common problem, and resourcefulness/following best practices will be favored over completely from-scratch solutions.

For example, use of AngularJS and data-binding is preferred over JQuery for instances where it is appropriate. (Hint there are some great samples out there that could get you 50% of the points with AngularJS).

The challenge is defined in two main parts, the front-end data model, and the user interface controls. The goal is to create an interactive table of a users active deals. These deals exist in two main categories, Single Vendor Deals, and Request for Quotes. As a user may have a large number of these items, we would like to stream the data to the frontend to speed page loading. A RESTful service is typically available for this purpose, but for this challenge this will be simulated by an AJAX get of two json response samples which are included below. These samples provide all of the necessary data for the table, from which you can build the frontend models. The requirements for the user interface are detailed below, and a sample of the current table is provided for reference. A table is not all that exciting, and should not take too long to hook up, so come up with a good idea and implement it or tell us about what you would have done next; you'll get bonus points for not simply following the script! Have fun!

Your Deals				
		START DEAL	START REQUEST	
Name	Type	Description	Status	Created
<a href="#">Awesome Deal A</a>	Request	Awesome Deal A is awesome!	Active	Mar 20, 2014
				REMOVE

# Frontend Data Model

## Necessary Features

- Store a list of Deal/Quote objects in a resource controller instance
- Deal/Quote objects should be created from the data returned by the two json files
  - Data should be loaded from data.json on page load
  - data-1.json should be streamed into the list in the background.
- Deal/Quote objects
  - contain data members matching the sample data
  - additional javascript functions such as print to console, and method signatures (stubs) showing how these models could be expanded.

## +1 Features

- Demonstrate use of AngularJS routes in an interesting way

# User Interface

## Necessary Features

- Display the data from the frontend model in a table
- Results should be paginated on the frontend based on the number of entries (5 per page)
- Allow text search on the title and description fields
- Sorting by date should be possible
- A user should be able to toggle between showing only deals or quotes
  - A combined view should be possible which groups deals and quotes
- Column with buttons for additional actions which when clicked act on the deal/quote in that row

## +1 Features

- Create a bootstrap modal with form
  - +5 if this form is linked to the table row and is ready to be attached to a REST service
- Impress us! A table is not all that exciting, and should not take too long to hook up, come up with a good idea, and you'll get bonus points for not simply following the script!

## Sample Data

These responses should be retrieved from the file names specified.

### Data Sample (initial data) data.json

```
[{"id": 110987, "description": "Description Deal A", "title": "Deal A", "last_updated": 1397847965149, "type": 0}, {"id": 110983, "description": "Awesome Description Deal B", "title": "Awesome Deal B", "last_updated": 1398020765149, "type": 0}, {"id": 110982, "description": "Cool Description C", "title": "Cool Deal C", "last_updated": 1398193565149, "type": 0}, {"id": 110980, "description": "Sweet Deal D", "title": "Deal D", "last_updated": 1398107165149, "type": 0}, {"id": 110999, "description": "Description Request A", "title": "Request A", "last_updated": 1398711965149, "type": 1}, {"id": 110905, "description": "Description Request B", "title": "Request B", "last_updated": 1398625565149, "type": 1}, {"id": 111000, "description": "Description Request C", "title": "Request C", "last_updated": 1398539165149, "type": 1}, {"id": 110973, "description": "Expensive Description Deal E", "title": "Expensive Deal E", "last_updated": 1398711965149, "type": 0}]
```

### Data Sample (additional rows) data-1.json

```
[{"id": 120000, "description": "Description Deal F", "title": "Deal F", "last_updated": 1397847965149, "type": 0}, {"id": 120560, "description": "Description Request X", "title": "Request X", "last_updated": 1398625565149, "type": 1}]
```