Introduction to Ember.js and Why Ember.js or React.js

Jiwei Ye

NetID:jy54

Abstract

This paper is going to introduce the basic concepts and usage of Ember.js, an open source MVC framework. Besides, by comparing to React.js, the advantages and disadvantages of Ember.js will be listed and discussed. Also, some personal suggestions on which framework to choose in different situations will be provided.

About Ember.js

To meet with the growing popularity of mobile web application, Single Page Application(SPA) consisting of a special structure separates the data from the presentation of data by having a model layer that handles data and a view layer that reads from model. This feature makes it possible that when redrawing any part of the user interface, the application doesn't need to require a server round-trip to retrieve HTML so as to offer a more-native-app-like experience to users[1]. And as one of those famous SPA frameworks, Ember.js derives from SproutCore MVC framework, and is, as stated by its creator, a framework for creating ambitious web applications[2].

By ambitious, they mean the applications built with Ember can make use of features which are generally available on native apps, e.g those apps can have 'states' and the application logic can be managed by the 'state' that the application is in.

Besides, Ember follows the principle of 'Convention over Configuration' which means that the framework is very heavily opinionated and there are a lot of conventions regarding the best practices of developing apps in Ember[4].

Handlebars is used by Ember to craft the markup of the application, which enables the developer to write less code in his templates and make use of curly brace helper to provide more functionality to the HTML. Utilizing Handlebars helpers, developer can pass real-time data to those templates.

Ember.js, as mentioned by Rey Bango[3] who once interviewed with the Ember.js team, enjoys many superb features like building desktop-like experiences, having data-binding event routing and state management when compared to some traditional libraries like jQuery, which is the only dependency of Ember.js. Before focusing on

the advantages&disadvantages and comparing it to react,js, some key concepts of Ember.js will be introduced firstly.
Routes: In Ember, an application's state is presented by a URL which will have a route object to determine what should be visible by the user
Models: Model contains the data associated with the state that the application is in and every route has an associated model.
Templates: Written with Handlebars, template are used to construct the HTML of the application, i.e template is what instructs Ember what HTML to render to your end user for each page
Components: A custom HTML tag. Also written with Handlebars.

Comparison to React.js

Comparing to Ember.js, a heavyweight framework, React.js is an open source front-end library run by Facebook and it is now getting more popular than Ember.js thanks to its great features. Now, let focus on the advantages and disadvantages of Ember,js when compared to React.js.

Undoubtedly, Ember.js does enjoy some great features. Unlike React.js which is more about View in MVC model, Ember is a framework truly based on MVC pattern, and developer can use the best coding practices and method to create single-page applications that are scalable. While React.js supports only one-way binding of data, Ember uses two-way bindings for registering HTML updates on a web page. Ember favors Convention over Configuration, which means that Ember will manage all the boilerplate code and can automatically set up most of the configuration, e.g to set up a new route, all you need to do is just typing in an one-line command and Ember will automatically generate files needed and update the router, which can save a lot of efforts and more importantly, you don't need to worry about the perfect think to do, since the choices made by the framework are right in about 90%[5]. Besides, Ember data, an optional data layer in Ember, integrates really nicely with a Ruby or Rails backend or any other RESTful JSON API that follows a set of conventions. Another feature that sets Ember apart is that a lot of extensions, e.g ember addons, can be easily added to your application with one terminal command. What's more, Ember has powerful routing at the expense of added complexity while React can not deal with routing.

On the other hand, Ember does have some drawbacks comparing to React.js. While React is arguably the most proformant and flexible rendering library around, even when rendering large list, Ember's patterns for structuring applications are baked into the framework in a convention-over-configuration manner, i.e Ember's view layer is less performant when rendering large lists[6]. Besides, Ember has no server-side rendering, which is so hot in the trending some time ago and now mostly required feature in any SPA application. And react does have server-rendering both with Alt - Flux - Managing your state or Read Me | Redux and reactjs/react-router, while Ember Fastboot can not work stably when compared to React. What's more, although

Ember Data performs well when dealing with standard REST API querying, you will end writing your own AJAX requests using jQuery middleware in Ember if you require to fetch data from API that is not bound by REST standards, or if you have API endpoint that returns some data in different way[7]. While React is considerably less opinionated, which makes development simpler, Ember is opinionated since it has stronger opinion about how an application should be built and assume that you application will conform to those expectation.

Which to Use and Conclusion

Personally, I think Ember.js can be a good choice if you want to be very productive and don't want to worry much about your developing environment in that Ember.js is a comprehensive front-end framework based on MVC. Also, if you have API in Rails or you have worked with Rails, it is easy to learn and work with Ember.js. Besides, if you don't mind having less space for customization and control of everything, Ember is the way to go.

On the contrary, if you don't mind constructing the developing environment as you want it to be, e.g webpack_react+redux+axios, React can be a good choice. In addition, if you are working on more than a SPA CRUD application and need much more control and customization power, then React should be a better choice.

Reference:

[1] *Single Page Apps in Depth*

http://singlepageappbook.com/goal.html

[2] *A Framework For Creating Ambitious Web Applications*

http://emberjs.com/

[3] *Getting into Ember.js*

https://code.tutsplus.com/tutorials/getting-into-emberjs--net-30709

[4] *Flame on! A beginner's guide to Ember.js*

http://www.adobe.com/devnet/archive/html5/articles/flame-on-a-beginners-guide-to-emberjs.html

[5] *7 Reasons to Use Ember.js*

https://www.codeschool.com/blog/2015/10/26/7-reasons-to-use-ember-js/

[6] *React Goes Head-to-Head with Ember.js*

http://www.creativebloq.com/web-design/react-goes-head-head-emberjs-31514361

[7] *What are the benefits and drawbacks of Ember.js comparing to React and Angular?*

https://www.quora.com/What-are-the-benefits-and-drawbacks-of-Ember-js-comparing-to-React-and-Angular