**Infinite scrolling in AngularJS**

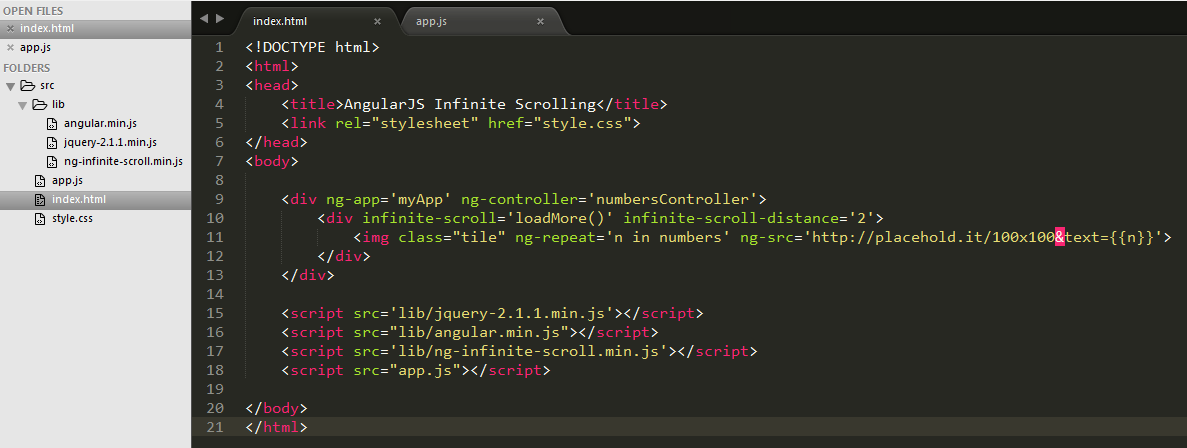
The other day I was discussing different JavaScript frameworks available to build Single Page Application [SPA] with my team. We discussed multiple options including ASP.Net MVC SPA template, Backbone, EmberJS, Durandal + knockout and AngularJS. By comparing different features available in these JavaScript frameworks, team member skillsets, application complexity and project timelines, team decided to go ahead with AngularJS. One of the important points discussed during these brainstorming sessions was how much data load is necessary during application start-up and how pagination can be implemented asynchronously. Business users also wanted team to come up with some *cool* ideas like fluid interface, dynamic loading of data etc. Apparently they all are active on Twitter and Facebook and they did not want to click on traditional ‘Next’ or ‘Previous’ buttons to navigate to next / previous set of records, and that’s when I started looking into infinite scrolling feature using AngularJS.

Now, for those who really don’t know what infinite scrolling is, it is also known as *endless scrolling*, is a technique where additional content for a web page is appended dynamically to the bottom of the page as the user approaches the end of the content. You may have seen this technique in Facebook or Twitter. If you want to quick demo, visit <http://gulpjs.com/plugins/> and notice as you scroll, additional content [plug-ins list in this case] gets loaded into the web page.

So now, the question was how to implement this feature using AngularJS. Off course you can implement this feature just by using vanilla JavaScript, but there are good people like [Samuel ROZE](https://github.com/sroze/)and open source community members who have created useful Angular directive [ngInfiniteScroll](https://github.com/sroze/ngInfiniteScroll) to achieve the same.

In this article, I will provide a quick walkthrough of how to use ngInfiniteScroll directive in your app to implement infinite scrolling. I assume that you have basic understanding of AngularJS as we are not going in depth to discuss what is Angular module, controller, view etc. Now we have the context set, it’s time to dive into the code.

I have downloaded ng-infinite-scroll.min.js from [ngInfiniteScroll](https://github.com/sroze/ngInfiniteScroll) github repository. Since it has dependency on jQuery and AngularJS go ahead and download [jQuery](http://jquery.com/), [AngularJS](https://angularjs.org/). Below screenshot shows how I setup the application. Even though it’s a fairly simple app, I prefer to keep vendor JavaScript files under *lib* directory and application code in app.js file.



Let’s understand Angular part of the HTML.

<div ng-app='myApp' ng-controller='numbersController'>

<div infinite-scroll='loadMore()' infinite-scroll-distance='2'>

<img class="tile" ng-repeat='n in numbers' ng-src='http://placehold.it/100x100&text={{n}}'>

</div>

</div>

The ng-app directive bootstraps the Angular application and you can have only one ng-app directive in an app. This directive can be initialized to an empty string [ng-app=''] or just without any value [ng-app] or using an angular module [ng-app='myApp']. In this case, we declared a myApp module in app.js file.

The ng-controller directive binds the DIV element to numbersController defined in JavaScript file.

var myApp = angular.module('myApp', ['infinite-scroll']);

var numbersController = function($scope){

$scope.numbers = [];

$scope.counter = 0;

$scope.loadMore = function () {

for (var i = 0; i < 25; i++) {

$scope.numbers.push(++$scope.counter);

};

}

$scope.loadMore();

}

myApp.controller('numbersController', numbersController);

The infinite-scroll directive contains an expression which gets executed when the bottom of the element approaches the bottom of browser window. In this case, since we have bind infinite-scroll directive to div element, as user scrolls the div [which contains list of images], it executes the loadMore function defined in numbersController; which then loads next 25 images and appends it to the list. The infinite-scroll-distance is an optional parameter which represents how close the bottom of the element must be to the bottom of browser window before the expression specified by infinite-scroll directive is triggered.

Note, here we are using a free image generator service <http://placehold.it> and providing the image dimension [100 x 100 px]. The {{n}} expression assigns the appropriate image number to the image source.

That’s all you need to do to implement infinite scrolling using AngularJS. Once you execute the code browser, you should get output as shown below.

<<Infinite scrolling output>>