This is an Angular Directive implementation.

* **Variations:**

With JQuery plugin, we can implement functionality and DOM manipulations in the same plugin code.

But with Angular JS, being MVC framework, it encourages separation of concerns, hence interlinking functionality and DOM manipulations in same code is not recommended.

This introduces some variations:

1> Row Height: Rather than passing row height as a parameter to the directive, I have added a CSS class. We can use this class in different templates.

And because we can pass templates dynamically to my directive, hence, row height being dynamic is achieved.

Note: I can easily pass row height parameter to directive and achieve it with ng-style. But ng-style would render inline style for <tr>.

Having it in CSS and dynamically using it in template seemed a better approach in angular world.

2> "build row" event: Bubbling up this event to page controller would mean handling DOM manipulations in the controller, which would be an MVC anti-pattern. Directives and Templates serve that purpose.

* **Implementation:**

Directive accepts following parameters (scope values):

1> datasource: data for our scrollable table

2> templateid: dynamically passed template id. This will enable to render dynamic templates based on the page requirement. Currently I have a single column template which was required. If required, we can have another template which renders 2 columns and respective 2 column JSON structure/array can be passed in.

No code changes would be required in the directive.

3> preloaded: I have implemented 2 variations for the directive. If we have all the data preloaded in the table datasource. We send this "preloaded" parameter as true. Else, we can send it "false" and send below parameter (4)

4> loadfn: When preloaded parameter is false, we can send a function argument which would then be called when you scroll the table to the bottom. This will load more rows. As this would be requirement based, getting more rows has to be invoked on the calling controller as different pages would be fetching data in from different sources. Hence the abstraction of fetch function from directive.

Note: To test "Pre-Load" and "Asyn Load" just comment/uncomment following lines in Controller:

SetAsyncLoad();

//SetPreloaded();

* **Considerations:**

1> Embedded Template: I have used embedded templates using script tag :

<script type="text/ng-template" id="scrollTable">...</script>.

We are serving this page directly from file system and not using any web server. Templates files cannot be served directly from file system because all browsers will block these kind of XHR calls. The only way to get this to work is to have templates rendered in script tag OR use inline template HTML in the directive. I have used the first option which is cleaner and easier to maintain.

In reality, these templates would sit somewhere in Templates folder on the server and would be served by web server on call basis.

2> as this is not a production ready code, for simplicity, I have script.js file to write my App module, App controller and scrollableTable directive.

In reality, these would be separated in different files.