

EXPORT EXCEL IN GRAFANA

1. PURPOSE

Allowing Grafana to export to excel-readable format (.xlsx). The excel file will contain the data points of the current view graph. This will facilitate data manipulation in the future using excel features (bar charts, pie charts, pivot table, or VBA).

Furthermore, an additional feature is added so as to print all panels (within a dashboard) as PDF file for ease of monthly reporting.

2. INSTALLATION & MANUAL

Step 1: Download and install Grafana from Source Code (<https://github.com/grafana/grafana>). Follow the instruction to install all dependencies (Go 1.5, NodeJS V4+, Godep).

Grafana requires correct version of dependencies to be installed.

***Note:** PhantomJS installing from Grafana is currently faulty → solution install PhantomJS separately

Step 2: Replace the entire github.com/grafana/grafana/public folder with **public** folder (provided)

Step 3: Building from Grafana Source Code

Building the Backend

```
go run build.go build
```

Building the Frontend

```
npm install  
npm run build
```

Only need to do this once before running Grafana

Step 4: Start Grafana server and use as per normal.

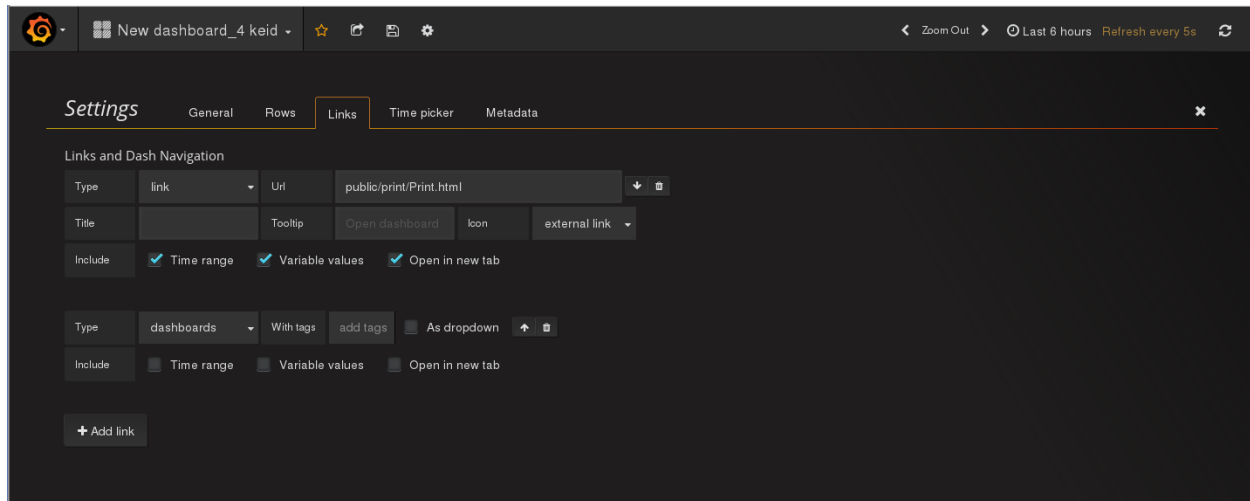
Running

```
./bin/grafana-server
```

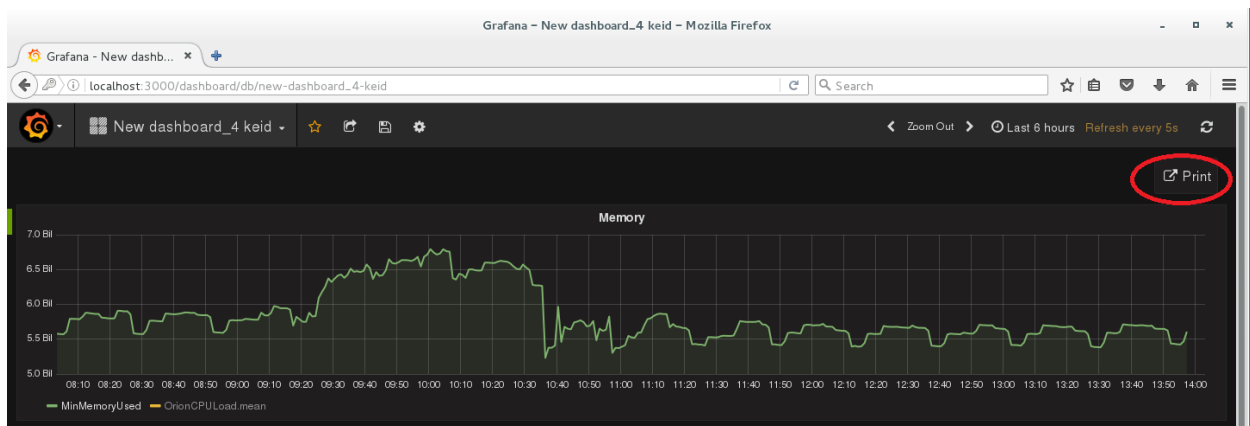
***Note:** Grafana server will have to be started again after reboot

Step 5: To add the function Print To PDF. Login to Grafana → Select Dashboard → Setting → Links

- **Type:** Choose link
- **URL:** public/print/Print.html
- **Include:** Time range, Variable values (optional), Open in new tab
- **Title:** Print (Or other title that you want)

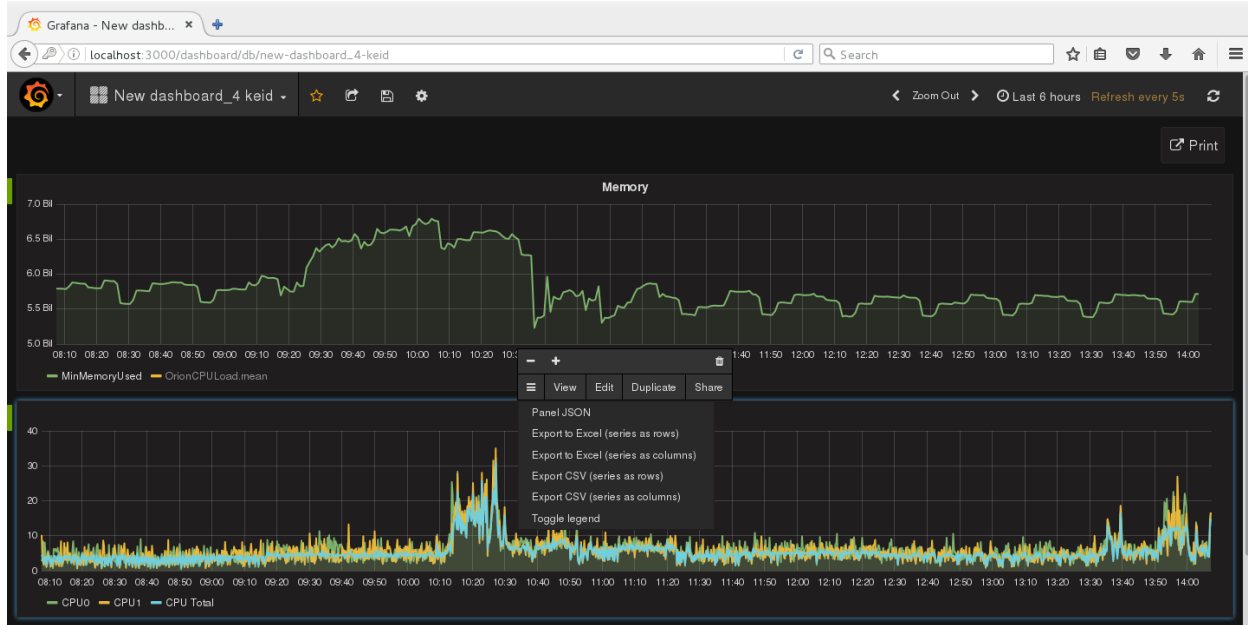


Step 6: Save Dashboard. Print button will appear on the right corner of the dashboard.



Step 7: Press Print button will open new tab with all panels view. Use Print function (Ctrl+P) in Chrome and save as PDF

Step 8: To export data to Excel file. Navigate to a specific panel that you want to extract, dropdown button → choose Export to Excel (series as rows) or Export to Excel (series as column) to export



3. SOURCE CODE

1. Adding buttons to Grafana console

Depend on what type of panel, you need to update the button accordingly

For example:

Update `./public/app/plugins/panel/graph/module.ts` for graph

Or Update `./public/app/plugins/panel/table/module.ts` for table

Add `actions.push({text: 'Your button name', click: 'buttonFunc()});`

```
~/go/src/github.com/grafana/grafana/public/app/plugins/panel/table/module.ts (table) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
  table
    img
    specs
    renderer_spec
    transformers
    README.md
    editor.html
    editor.ts
    module.html
    module.ts
    options.html
    plugin.json
    renderer.ts
    transformers.ts

52 if (this.panel.styles === void 0) {
53   this.panel.styles = this.panel.columns;
54   this.panel.columns = this.panel.fields;
55   delete this.panel.columns;
56   delete this.panel.fields;
57 }
58
59 _defaults(this.panel, this.panelDefaults);
60
61 this.events.on('data-received', this.onDataReceived.bind(this));
62 this.events.on('data-error', this.onDataError.bind(this));
63 this.events.on('data-snapshot-load', this.onDataReceived.bind(this));
64 this.events.on('init-edit-mode', this.onInitEditMode.bind(this));
65 this.events.on('init-panel-actions', this.onInitPanelActions.bind(this));
66
67
68 onInitEditMode() {
69   this.addEditorTab('Options', tablePanelEditor, 2);
70 }
71
72 onInitPanelActions(actions) {
73   actions.push({text: 'Export to Excel', click: 'ctrl.exportXML()'});
74   actions.push({text: 'Export CSV', click: 'ctrl.exportCsv()'});
75 }
76
77 issueQueries(datasource) {
78   this.pageIndex = 0;
79
80   if (this.panel.transform === 'annotations') {
81     this.setTimeQueryStart();
82     return this.annotationsSrv.getAnnotations(this.dashboard).then(annotations => {
83       return {data: annotations};
84     });
85   }
86
87   return super.issueQueries(datasource);
88 }
89
90
```

Add `buttonFunc(){ FileExport.buttonFunc(...);}`

```
~/go/src/github.com/grafana/grafana/public/app/plugins/panel/table/module.ts (table) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
  table
    img
    specs
    renderer_spec
    transformers
    README.md
    editor.html
    editor.ts
    module.html
    module.ts
    options.html
    plugin.json
    renderer.ts
    transformers.ts

124 // remove sort flag from current column
125 if (this.table.columns[this.panel.sort.col]) {
126   this.table.columns[this.panel.sort.col].sort = false;
127 }
128
129 if (this.panel.sort.col === colIndex) {
130   if (this.panel.sort.desc) {
131     this.panel.sort.desc = false;
132   } else {
133     this.panel.sort.col = null;
134   }
135 } else {
136   this.panel.sort.col = colIndex;
137   this.panel.sort.desc = true;
138 }
139 this.render();
140
141
142 exportCsv() {
143   FileExport.exportTableDataToCsv(this.table);
144 }
145
146 exportXML() {
147   FileExport.exportTableDataToXML(this.table);
148 }
149
150 link(scope, elem, attrs, ctrl) {
151   var data;
152   var panel = ctrl.panel;
153   var pageCount = 0;
154   var formatters = [];
155
156   function getTableHeight() {
157     var panelHeight = ctrl.height;
158   }
159 }
160
161
```

2. Implement function of button

Navigate to `./public/app/core/utils/file_export.ts`

Import necessary files and then write your own function

~/go/src/github.com/grafana/grafana/public/app/core/utills/file_export.ts (table) - Sublime Text (UNREGISTERED)

```

1  //<reference path="../../headers/common.d.ts" />
2
3  import _ from 'lodash';
4  //import * as XLSX from 'XLSX';
5  import * as Excel from 'excel';
6  declare var window: any;
7  declare function require (path: string): any;
8
9
10 export function exportTableDataToXML(table){
11     var title = [];
12     var titleCount = 0;
13     var ep = new Excel.ExcelPlus();
14     ep.createFile("Grafana");
15     _each(table.columns, function(column) {
16         title[titleCount++] = column.text;
17     });
18     //Write titles
19     ep.write({ "content": [title] });
20     //Write data
21     _each(table.rows, function(row) {
22         var data = [];
23         var dataCount = 0;
24         _each(row, function(value) {
25             if (dataCount === 0){
26                 data[dataCount++] = convertTime(value);
27             } else {
28                 data[dataCount++] = value;
29             }
30         });
31         ep.writeNextRow(data);
32     });
33     try {
34         ep.saveAs("Table_Data.xlsx");
35     } catch (e){
36         alert("Table_Data_Error");
37     }
38 };
```

Import External JS Library

Write your own function for the button

3. Set directories for external library (if you want to use external Javascript library)

Navigate to `./public/app/system.conf.js` and then update the correct path to your JS library. In this case: path to `"excel"` module is `vendor/excel/excelplus-2.4.min`

~/go/src/github.com/grafana/grafana/public/app/system.conf.js (table) - Sublime Text (UNREGISTERED)

```

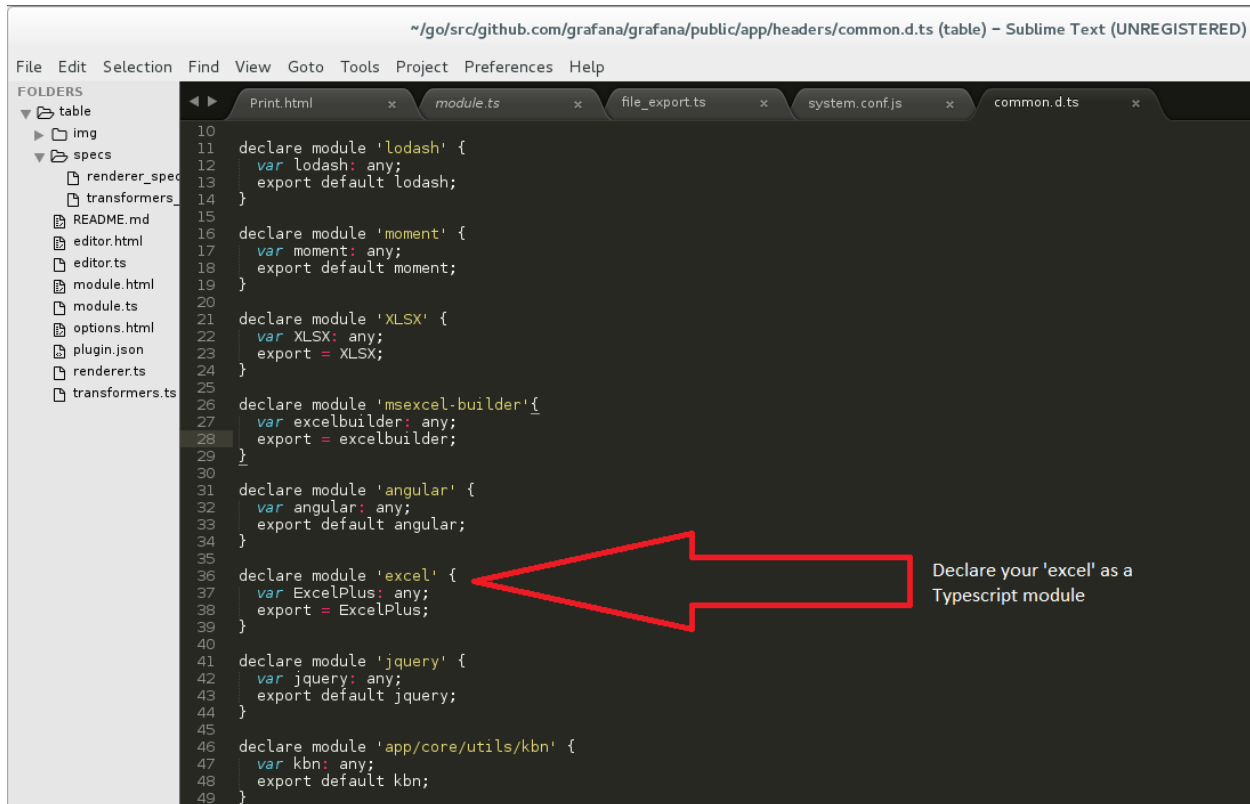
1  System.config({
2      defaultJSExtensions: true,
3      baseUrl: 'public',
4      paths: {
5          'remarkable': 'vendor/npm/remarkable/dist/remarkable.js',
6          'tether': 'vendor/npm/tether/dist/js/tether.js',
7          'eventemitter3': 'vendor/npm/eventemitter3/index.js',
8          'tether-drop': 'vendor/npm/tether-drop/dist/js/drop.js',
9          'moment': 'vendor/moment.js',
10         'jquery': 'vendor/jquery/dist/jquery.js',
11         'lodash-src': 'vendor/lodash.js',
12         'lodash': 'app/core/lodash_extended.js',
13         'angular': 'vendor/angular/angular.js',
14         'bootstrap': 'vendor/bootstrap/bootstrap.js',
15         'angular-route': 'vendor/angular-route/angular-route.js',
16         'angular-sanitize': 'vendor/angular-sanitize/angular-sanitize.js',
17         'angular-ui': 'vendor/angular-ui/ui-bootstrap-tpls.js',
18         'angular-strap': 'vendor/angular-other/angular-strap.js',
19         'angular-dragdrop': 'vendor/angular-native-dragdrop/draganddrop.js',
20         'angular-bindonce': 'vendor/angular-bindonce/bindonce.js',
21         'spectrum': 'vendor/spectrum.js',
22         'bootstrap-tagsinput': 'vendor/tagsinput/bootstrap-tagsinput.js',
23         'jquery.flot': 'vendor/flot/jquery.flot',
24         'jquery.flot.pie': 'vendor/flot/jquery.flot.pie',
25         'jquery.flot.selection': 'vendor/flot/jquery.flot.selection',
26         'jquery.flot.stack': 'vendor/flot/jquery.flot.stack',
27         'jquery.flot.stackpercent': 'vendor/flot/jquery.flot.stackpercent',
28         'jquery.flot.time': 'vendor/flot/jquery.flot.time',
29         'jquery.flot.crosshair': 'vendor/flot/jquery.flot.crosshair',
30         'jquery.flot.fillbelow': 'vendor/flot/jquery.flot.fillbelow',
31         'jquery.flot.gauge': 'vendor/flot/jquery.flot.gauge',
32         'XLSX': 'vendor/js-xlsx-master/dist/xlsx.core.min.js',
33         'msexcel-builder': 'vendor/msexcel-builder/lib/msexcel-builder',
34         'excel': 'vendor/excel/excelplus-2.4.min',
35         'sheet': 'vendor/excel/sheetjs.all.min'
36     },
37     packages: {
38         '': {
39             'main': 'index.js',
40             'test': 'test.js',
41             'dev': 'dev.js',
42             'dist': 'dist.js',
43             'lib': 'lib.js',
44             'src': 'src.js',
45             'testbed': 'testbed.js',
46             'testbed2': 'testbed2.js',
47             'testbed3': 'testbed3.js',
48             'testbed4': 'testbed4.js',
49             'testbed5': 'testbed5.js',
50             'testbed6': 'testbed6.js',
51             'testbed7': 'testbed7.js',
52             'testbed8': 'testbed8.js',
53             'testbed9': 'testbed9.js',
54             'testbed10': 'testbed10.js',
55             'testbed11': 'testbed11.js',
56             'testbed12': 'testbed12.js',
57             'testbed13': 'testbed13.js',
58             'testbed14': 'testbed14.js',
59             'testbed15': 'testbed15.js',
60             'testbed16': 'testbed16.js',
61             'testbed17': 'testbed17.js',
62             'testbed18': 'testbed18.js',
63             'testbed19': 'testbed19.js',
64             'testbed20': 'testbed20.js',
65             'testbed21': 'testbed21.js',
66             'testbed22': 'testbed22.js',
67             'testbed23': 'testbed23.js',
68             'testbed24': 'testbed24.js',
69             'testbed25': 'testbed25.js',
70             'testbed26': 'testbed26.js',
71             'testbed27': 'testbed27.js',
72             'testbed28': 'testbed28.js',
73             'testbed29': 'testbed29.js',
74             'testbed30': 'testbed30.js',
75             'testbed31': 'testbed31.js',
76             'testbed32': 'testbed32.js',
77             'testbed33': 'testbed33.js',
78             'testbed34': 'testbed34.js',
79             'testbed35': 'testbed35.js',
80             'testbed36': 'testbed36.js',
81             'testbed37': 'testbed37.js',
82             'testbed38': 'testbed38.js',
83             'testbed39': 'testbed39.js',
84             'testbed40': 'testbed40.js',
85             'testbed41': 'testbed41.js',
86             'testbed42': 'testbed42.js',
87             'testbed43': 'testbed43.js',
88             'testbed44': 'testbed44.js',
89             'testbed45': 'testbed45.js',
90             'testbed46': 'testbed46.js',
91             'testbed47': 'testbed47.js',
92             'testbed48': 'testbed48.js',
93             'testbed49': 'testbed49.js',
94             'testbed50': 'testbed50.js',
95             'testbed51': 'testbed51.js',
96             'testbed52': 'testbed52.js',
97             'testbed53': 'testbed53.js',
98             'testbed54': 'testbed54.js',
99             'testbed55': 'testbed55.js',
100            'testbed56': 'testbed56.js',
101            'testbed57': 'testbed57.js',
102            'testbed58': 'testbed58.js',
103            'testbed59': 'testbed59.js',
104            'testbed60': 'testbed60.js',
105            'testbed61': 'testbed61.js',
106            'testbed62': 'testbed62.js',
107            'testbed63': 'testbed63.js',
108            'testbed64': 'testbed64.js',
109            'testbed65': 'testbed65.js',
110            'testbed66': 'testbed66.js',
111            'testbed67': 'testbed67.js',
112            'testbed68': 'testbed68.js',
113            'testbed69': 'testbed69.js',
114            'testbed70': 'testbed70.js',
115            'testbed71': 'testbed71.js',
116            'testbed72': 'testbed72.js',
117            'testbed73': 'testbed73.js',
118            'testbed74': 'testbed74.js',
119            'testbed75': 'testbed75.js',
120            'testbed76': 'testbed76.js',
121            'testbed77': 'testbed77.js',
122            'testbed78': 'testbed78.js',
123            'testbed79': 'testbed79.js',
124            'testbed80': 'testbed80.js',
125            'testbed81': 'testbed81.js',
126            'testbed82': 'testbed82.js',
127            'testbed83': 'testbed83.js',
128            'testbed84': 'testbed84.js',
129            'testbed85': 'testbed85.js',
130            'testbed86': 'testbed86.js',
131            'testbed87': 'testbed87.js',
132            'testbed88': 'testbed88.js',
133            'testbed89': 'testbed89.js',
134            'testbed90': 'testbed90.js',
135            'testbed91': 'testbed91.js',
136            'testbed92': 'testbed92.js',
137            'testbed93': 'testbed93.js',
138            'testbed94': 'testbed94.js',
139            'testbed95': 'testbed95.js',
140            'testbed96': 'testbed96.js',
141            'testbed97': 'testbed97.js',
142            'testbed98': 'testbed98.js',
143            'testbed99': 'testbed99.js',
144            'testbed100': 'testbed100.js'
145        }
146    });
```

Set a correct directories to the JS library that you want to use

4. Declare the newly added Javascript library as a module for Typescript

Since you are using JS library for Typescript (Grafana), you will need to declare your JS library as a MODULE

Navigate to [./public/app/headers/common.d.ts](#) and declare 'excel' module



```

10
11 declare module 'lodash' {
12   var lodash: any;
13   export default lodash;
14 }
15
16 declare module 'moment' {
17   var moment: any;
18   export default moment;
19 }
20
21 declare module 'xlsx' {
22   var XLSX: any;
23   export = XLSX;
24 }
25
26 declare module 'msexcel-builder' {
27   var excelbuilder: any;
28   export = excelbuilder;
29 }
30
31 declare module 'angular' {
32   var angular: any;
33   export default angular;
34 }
35
36 declare module 'excel' {
37   var ExcelPlus: any;
38   export = ExcelPlus;
39 }
40
41 declare module 'jquery' {
42   var jquery: any;
43   export default jquery;
44 }
45
46 declare module 'app/core/utils/kbn' {
47   var kbn: any;
48   export default kbn;
49 }

```

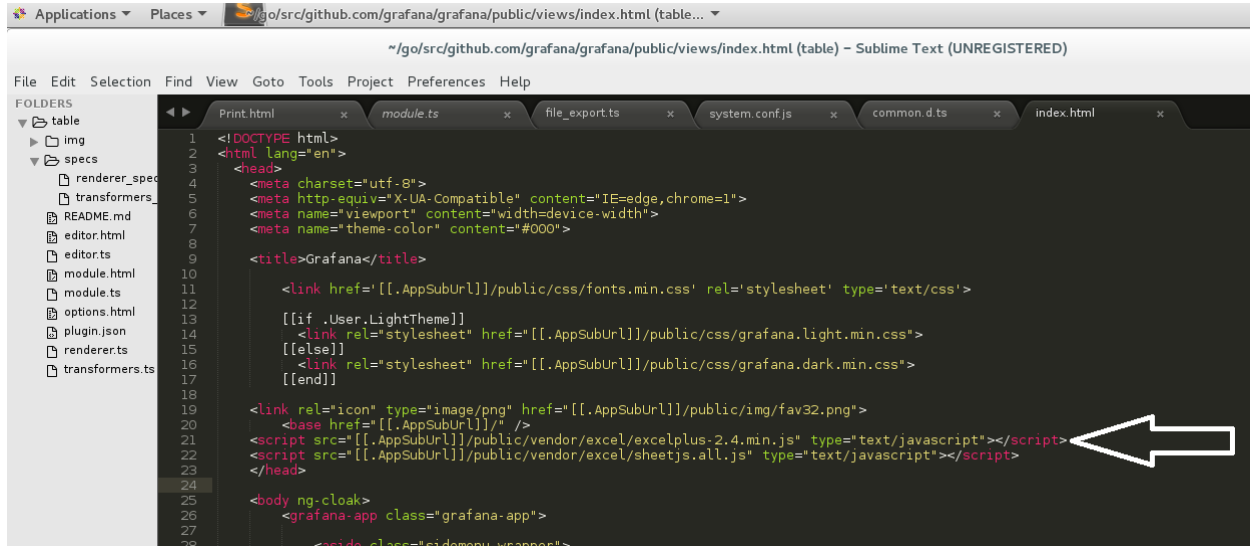
Declare your 'excel' as a Typescript module

5. Place the external library in correct directories

Put your external JS library to correct directory that you have set in step 3. For example [./public/vendor/excel/](#)

6. Import Javascript (*.js) file from external library to the require page that uses the function

Since I am using my external JS function in Grafana Homepage. I will edit the `./public/views/index.html` to include the library `<script src = ".../excelplus.js">`



```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
6 <meta name="viewport" content="width=device-width">
7 <meta name="theme-color" content="#000">
8
9 <title>Grafana</title>
10
11 <link href="[[.AppSubUrl]]/public/css/fonts.min.css" rel="stylesheet" type="text/css">
12
13 [[if .User.LightTheme]]
14 <link rel="stylesheet" href="[[.AppSubUrl]]/public/css/grafana.light.min.css">
15 [[else]]
16 <link rel="stylesheet" href="[[.AppSubUrl]]/public/css/grafana.dark.min.css">
17 [[end]]
18
19 <link rel="icon" type="image/png" href="[[.AppSubUrl]]/public/img/fav32.png">
20 <base href="[[.AppSubUrl]]/" />
21 <script src="[[.AppSubUrl]]/public/vendor/excel/excelplus-2.4.min.js" type="text/javascript"></script>
22 <script src="[[.AppSubUrl]]/public/vendor/excel/sheetjs.all.js" type="text/javascript"></script>
23 </head>
24
25 <body ng-cloak>
26 <grafana-app class="grafana-app">
27
28 <aside class="sidemenu-wrapper">

```

4. AUTHOR

Name: Nguyen Luong Chuong Thien

Last updated: July 5, 2016

Email: terrryn@ncs.com.sg or nl.chuongthien@u.nus.edu