1. На странице подключить следующие скрипты и стили (все стили и скрипты находятся в архиве ресурсов):

<script src="bower\_components/jquery/dist/jquery.min.js"></script>

<script src="bower\_components/angular/angular.min.js"></script>

<script src="bower\_components/ng-grid/ng-grid-2.0.14.debug.js"></script>

<script src="app/custom-grid.js"></script>

<script src="app/testApp.js"></script>

<script src="app/grid-upload-controller.js"></script>

<link href="bower\_components/ng-grid/ng-grid.min.css" rel="stylesheet" />

<link href="css/styles.css" rel="stylesheet" />

1. Создать новый модуль ангулара и добавить в нём зависимость на модуль ‘gridTaskApp’, либо использовать модуль ‘gridTaskApp’ на аттрибуте ng-app. Например:

angular.module('test', ['gridTaskApp']) или <body ng-app="gridTaskApp">

1. Создать контроллер для инициализации данных и необходимых опций.
2. Добавить директиву на странице, проинициализировав нужные ей элементы в аттрибуты элемента:

<page-content grid-data="data" content-options="contentOptions" grid="grid" grid-options="gridOptions"></page-content>

Где data – коллекция данных, необходимых для гриды, contentOptions – опции для контента, grid – значения описания гриды (имя, кол-во записей), grid-options – опции для гриды (на основе опции от ng-grid).

Content options:

|  |  |  |
| --- | --- | --- |
| Имя свойства | По умолчанию | Определение |
| loading | false | Отображение loading на контенте гриды. |
| checks | {  options: {  actions: {  all: { label: 'All', isAll: true },  noOne: { label: 'No one', isNoOne: true },  marked: { label: 'Marked', isMarked: true },  notMarked: { label: 'Not marked', isNotMarked: true }  },  callback: function (check) {  if (check) {  if (check.isAll) {  scope.data.forEach(function (value) {  value.isCheck = true;  });  }  else if (check.isNoOne) {  scope.data.forEach(function (value) {  value.isCheck = false;  });  }  else if (check.isMarked) {  scope.data.forEach(function (value) {  });  }  else if (check.isNotMarked) {  scope.data.forEach(function (value) {  value.isCheck = !value.isCheck;  });  }  }  }  } | Опции для дропдауна с селектом(значения и функция обратного вызова) |
| mores | {  options:  {  label: 'More',  values: [{ label: 'View reports' }],  isMenu: true  }  }; | Дропдаун меню ‘Mores’ |
| filtrate | function (value) {  scope.gridOptions.filterOptions.filterText = convertFilterOptions(value).filterText;  }; | Функция по нажатия на кнопку ‘Show records’ в окне филтра ‘Filter’. |
| search | function (value) {  scope.gridOptions.filterOptions.filterText = value;  } | Функция по изменению значения в поле search быстрого фильтра |
| refresh | function () {  scope.contentOptions.refreshCallback();  scope.grid.count = scope.data.length;  }; | Действие по нажатию на кнопку refresh. |
| refreshCallback | undefined | Сallback по нажатию на кнопку refresh. |
| withUpload | false | Отображение кнопки upload с действие по умолчанию. |
| upload | function (data) {  scope.data = data;  scope.grid.count = scope.data.length;  scope.$apply();  } | Функция по нажатию на кнопку upload. |

Grid:

Default :

grid.name = 'Default grid';

if (Array.isArray(data)) {

grid.count = scope.data.length;

}

Grid options (by http://angular-ui.github.io/ng-grid/):

Default: