# Creating a new component TABLE

**Implementing a component**

1. Add a new folder in packages\components\src\table
2. Add new typescript file in class packages\components\src\table.ts
3. Import core classed and interfaces

import { BaseComponent, ProjectorOptions, VNode, h, hParser } from './../libs';

1. Define interface for ITableComponent

export interface IListComponent {

refresh(): void;

}

1. Define interface for ITableOptions

export interface ITableOptions {

height?: number;

data?: Array<any>;

pageSize?: number;

autoPage?: boolean;

template?: string;

}

1. Implement ITableComponent and define default properties

export class Table extends BaseComponent<ITableOptions> implements ITableComponent {

public static defaultOptions: ITableOptions = {

height: 40,

pageSize: 0,

data: [],

autoPage: true,

template: ''

}

constructor(element: HTMLElement, options?: ITableOptions) {

super('Table', element, { ...Table.defaultOptions, ...options });

this.projector.append(this.element, this.render.bind(this));

}

public render(): VNode {

return h('div',[ 'Hello Table!']);

}

public refresh() {

this.projector.scheduleRender();

}

}

super() merges the default options with user provided options and sets to the component

this.projector.append() replaces the rendered html inside the html element

render() generated the html

refresh(): ’this.projector.scheduleRender()’ re-renders the html

1. Export table component to public; **packages\components\src\components.ts**

import { Table, ITableComponent, ITableOptions } from './table/table';

export {

// Add table classes and interfaces

Table, ITableComponent, ITableOptions

};

Add component to the components build

1. To add table to the **@batman\components-es5** build; Update **entry** object in **webpack\webpack-components.es5.config.js** , **webpack\ webpack-components.es5.prod.config.js**

table: './packages/components/src/table/table.ts'

1. To add table to the **@batman\components** build; Update **entry** object in **webpack\webpack-components.umd.config.js** , **webpack\ webpack-components.umd.prod.config.js**
2. Build packages; open command prompt and build

npm run build-components

npm run build-components-es5

This will add the table component to the packages which you can install in your project.

# Create a demo page and run the component

1. Build process creates **dist\output\components-es5** folder in the project root; refer the **tree.js** file from the **dist** folder in **public\js\src\index.html** (referring from dist folder is only for debugging the component, for production don’t refer from this folder)
2. In **public\js\src\index.html** add

<script src="../../../dist/output/components-es5/table.js"></script>

1. Create a folder **table** in **public\js\src\**
2. Add a controller and views in this folder (please refer list folder; **insider** is a light weight MVC framework to run components in SPA model; which is part of example application)
3. Typically **copy basic.htm, home.html and list.controller.js** from list folder and rename, update code accordingly.
4. Add table.controller.js reference in **public\js\src\index.html file**

<script src="./table/table.controller.js"></script>

1. table.controller.js

(function (module) {

var Table = Batman.table.Table;

module.controller('tableHomeController', function () {

this.header = 'Table Home';

})

module.controller('tableBasicController', function () {

this.header = 'Table Basic Example';

this.data = [{ "text": "0" }, { "text": "1" }];

this.onload = function (view) {

new Table(view.querySelector('#t1'), {

data: this.data,

height: 30,

});

}

})

})(AppModule || (AppModule = {}));

1. basic.hml

<div class="panel flex item-full column no-mar-collapse">

<h2> {{header}} </h2>

<div>

options { height: 30}

</div>

<div id="t1" class="flex item-full"></div>

</div>

1. Add route information in public\js\src\index.js

{ text: 'Table', url: '/table-home', template: 'table/home.htm', controller: 'tableHomeController' },

{ text: 'Basic', url: '/table-basic', template: 'table/basic.htm', controller: 'tableBasicController' }

1. Add styles; create a folder **table** in file **packages\sass\**
2. Create \_table.scss file and refer it in packages\sass\main.scss

@import "table/\_table";

1. Create .table class and write all CSS styles in \_table.scss

.table {

overflow: hidden;

}

1. Run the application in debug mode

npm run start-debug

1. Navigate to the web demo page

http://localhost:8000/public/js/src/index.html#/table-basic

1. Basic structure is ready now implement Table functionalities; ☺

Happy Coding!