

Angular Google Charts - Quick Guide

Advertisements



[⊕ Previous Page](#)

[Next Page](#) 

Angular Google Charts - Overview

Google Charts is a pure JavaScript based charting library meant to enhance web applications by adding interactive charting capability. It supports a wide range of charts. Charts are drawn using SVG in standard browsers like Chrome, Firefox, Safari, Internet Explorer(IE). In legacy IE 6, VML is used to draw the graphics.

angular-google-charts is a open source angular based wrapper for Google Charts to provides an elegant and feature rich Google Charts visualizations within an Angular application and can be used along with Angular components seamlessly. There are chapters discussing all the basic components of Google Charts with suitable examples within a Angular application.

Features

Following are the salient features of Google Charts library.

Compatability – Works seamlessly on all major browsers and mobile platforms like android and iOS.

Multitouch Support – Supports multitouch on touch screen based platforms like android and iOS. Ideal for iPhone/iPad and android based smart phones/ tablets.

Free to Use – Open source and is free to use for non-commercial purpose.

Lightweight – loader.js core library, is extremely lightweight library.

Simple Configurations – Uses json to define various configuration of the charts and very easy to learn and use.

Dynamic – Allows to modify chart even after chart generation.

Multiple axes – Not restricted to x, y axis. Supports multiple axis on the charts.

Configurable tooltips – Tooltip comes when a user hover over any point on a charts. googlecharts provides tooltip inbuilt formatter or callback formatter to control the tooltip programmatically.

DateTime support – Handle date time specially. Provides numerous inbuilt controls over date wise categories.

Print – Print chart using web page.

External data – Supports loading data dynamically from server. Provides control over data using callback functions.

Text Rotation – Supports rotation of labels in any direction.

Supported Chart Types

Google Charts library provides following types of charts:

Sr.No.	Chart Type / Description
1	Line Charts Used to draw line/spline based charts.
2	Area Charts Used to draw area wise charts.
3	Pie Charts Used to draw pie charts.
4	Sankey Charts, Scatter Charts, Stepped area charts, Table, Timelines, TreeMap, Trendlines Used to draw scattered charts.
5	Bubble Charts Used to draw bubble based charts.
6	Dynamic Charts Used to draw dynamic charts where user can modify charts.
7	Combinations Used to draw combinations of variety of charts.

8	3D Charts Used to draw 3D charts.
9	Angular Gauges Used to draw speedometer type charts.
10	Heat Maps Used to draw heat maps.
11	Tree Maps Used to draw tree maps.

In next chapters, we're going to discuss each type of above mentioned charts in details with examples.

Licence

Google Charts is open source and is free to use. Follow the link – [Terms of Service](#) .

Angular Google Charts - Environment Setup

This tutorial will guide you on how to prepare a development environment to start your work with Google Charts and Angular Framework. In this chapter, we will discuss the Environment Setup required for Angular 6. To install Angular 6, we require the following –

Nodejs

Npm

Angular CLI

IDE for writing your code

Nodejs has to be greater than 8.11 and npm has to be greater than 5.6.

Nodejs

To check if nodejs is installed on your system, type **node -v** in the terminal. This will help you see the version of nodejs currently installed on your system.

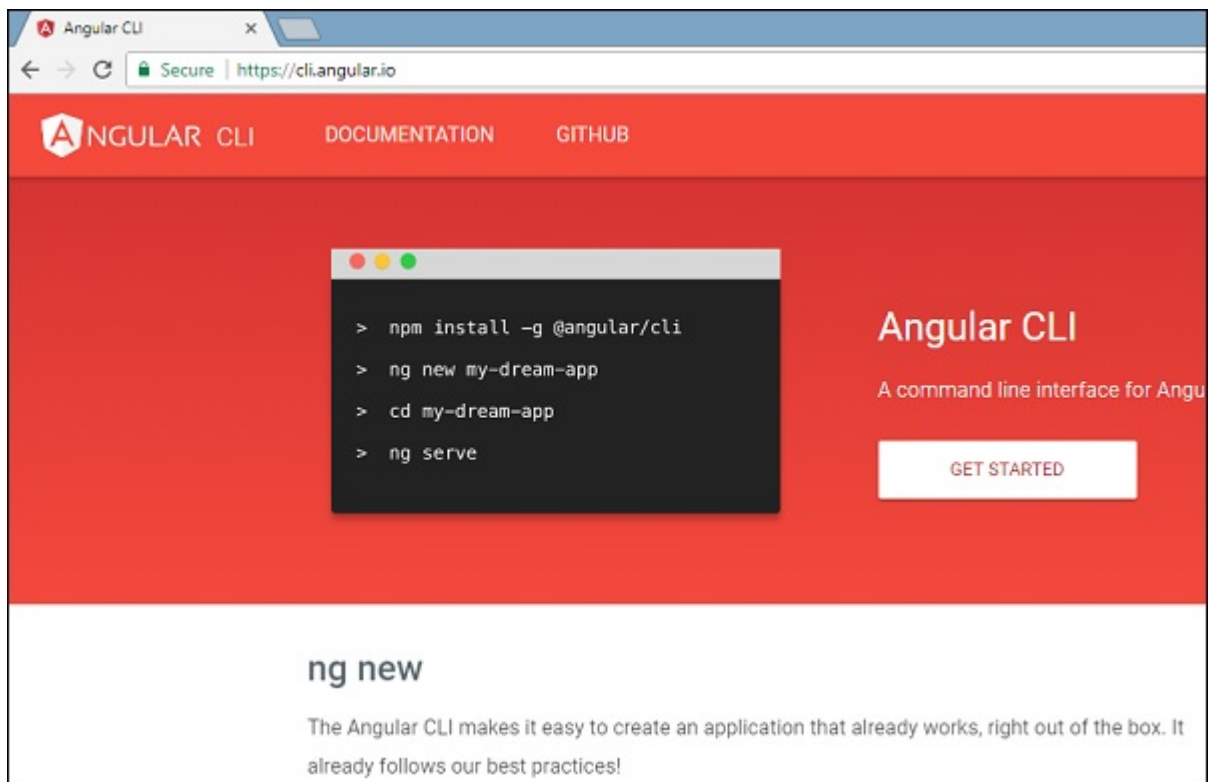
```
C:\>node -v  
v8.11.3
```

If it does not print anything, install nodejs on your system. To install nodejs, go the homepage <https://nodejs.org/en/download/> of nodejs and install the package based on your OS.

Based on your OS, install the required package. Once nodejs is installed, npm will also get installed along with it. To check if npm is installed or not, type `npm -v` in the terminal. It should display the version of the npm.

```
C:\>npm -v  
5.6.0
```

Angular 6 installations are very simple with the help of angular CLI. Visit the homepage <https://cli.angular.io/> of angular to get the reference of the command.



Type **`npm install -g @angular/cli`**, to install angular cli on your system.

```
Command Prompt

+-- extglob@0.3.2
+-- filename-resolve@2.0.1
+-- normalize-path@2.1.1
+-- remove-trailing-separator@1.0.2
+-- object.omit@2.0.1
+-- for-own@0.1.5
+-- parse-glob@3.0.4
+-- glob-base@0.3.0
+-- is-dotfile@1.0.3
+-- regex-cache@0.4.3
+-- is-equal-shallow@0.1.3
+-- is-primitive@2.0.0
+-- serve-index@1.9.0
+-- batch@0.6.1
+-- http-errors@1.6.1
+-- sockjs@0.3.18
+-- faye-websocket@0.10.0
+-- websocket-driver@0.6.5
+-- websocket-extensions@0.1.1
+-- uuid@2.0.3
+-- sockjs-client@1.1.2
+-- eventsource@0.1.6
+-- original@1.0.0
+-- url-parse@1.0.5
+-- querystringify@0.0.4
+-- faye-websocket@0.11.1
+-- json3@3.3.2
+-- url-parse@1.1.9
+-- querystringify@1.0.0
+-- spdy@3.4.7
+-- handle-thing@1.2.5
+-- http-deceiver@1.2.7
+-- select-hose@2.0.0
+-- spdy-transport@2.0.20
+-- detect-node@2.0.3
+-- hpack.js@2.1.6
+-- obuf@1.1.1
+-- wbuf@1.7.2
+-- minimalistic-assert@1.0.0
+-- yargs@6.6.0
+-- camelcase@3.0.0
+-- cliui@3.2.0
+-- string-width@1.0.2
+-- is-fullwidth-code-point@1.0.0
+-- string-width@1.0.2
+-- is-fullwidth-code-point@1.0.0
+-- yargs-parser@4.2.1
+-- webpack-merge@2.6.1
+-- zone.js@0.8.12

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.0.0 (node_modules\@a
ngular\cli\node_modules\chokidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@
1.1.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
>

C:\project\4>
```

You will get the above installation in your terminal, once Angular CLI is installed. You can use any IDE of your choice, i.e., WebStorm, Atom, Visual Studio Code, etc.

Install Google Charts Wrapper

Run the following command to install Google Charts Wrapper module in the project created.

```
googleChartsApp> npm install angular-google-charts
```

```
+ angular-google-charts@0.1.0
added 2 packages in 20.526s
```

Add the following entry in app.module.ts file

```
import { GoogleChartsModule } from 'angular-google-charts';

imports: [
  ...
  GoogleChartsModule
],
```

Angular Google Charts - Configuration Syntax

In this chapter, we will showcase the configuration required to draw a chart using the Google Chart API in Angular.

Step 1 - Create Angular Application

Follow the following steps to update the Angular application we created in *Angular 6 - Project Setup* chapter –

Step	Description
1	Create a project with a name <i>googleChartsApp</i> as explained in the <i>Angular 6 - Project Setup</i> chapter.
2	Modify <i>app.module.ts</i> , <i>app.component.ts</i> and <i>app.component.html</i> as explained below. Keep rest of the files unchanged.
3	Compile and run the application to verify the result of the implemented logic.

Following is the content of the modified module descriptor **app.module.ts**.

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppComponent } from './app.component';
import { GoogleChartsModule } from 'angular-google-charts';
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule, GoogleChartsModule
  ],
  providers: [], bootstrap: [AppComponent]
})
export class AppModule { }
```

Following is the content of the modified HTML host file **app.component.html**.

```
<google-chart #chart
  [title]="title"
  [type]="type"
  [data]="data"
  [columnNames]="columnNames"
  [options]="options"
  [width]="width"
```

```
[height]="height">
</google-chart>
```

We'll see the updated app.component.ts in the end after understanding configurations.

Step 2 - Use Configurations

Set Title

```
title = 'Browser market shares at a specific website, 2014';
```

Set Chart Type

```
type='PieChart';
```

data

Configure the data to be displayed on the chart.

```
data = [
  ['Firefox', 45.0],
  ['IE', 26.8],
  ['Chrome', 12.8],
  ['Safari', 8.5],
  ['Opera', 6.2],
  ['Others', 0.7]
];
```

column names

Configure the column names to be displayed.

```
columnNames = ['Browser', 'Percentage'];
```

options

Configure the other options.

```
options = {
  colors: ['#e0440e', '#e6693e', '#ec8f6e', '#f3b49f', '#f6c7b6'], is3D: true
};
```

Example

Consider the following example to further understand the Configuration Syntax –

app.component.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
```

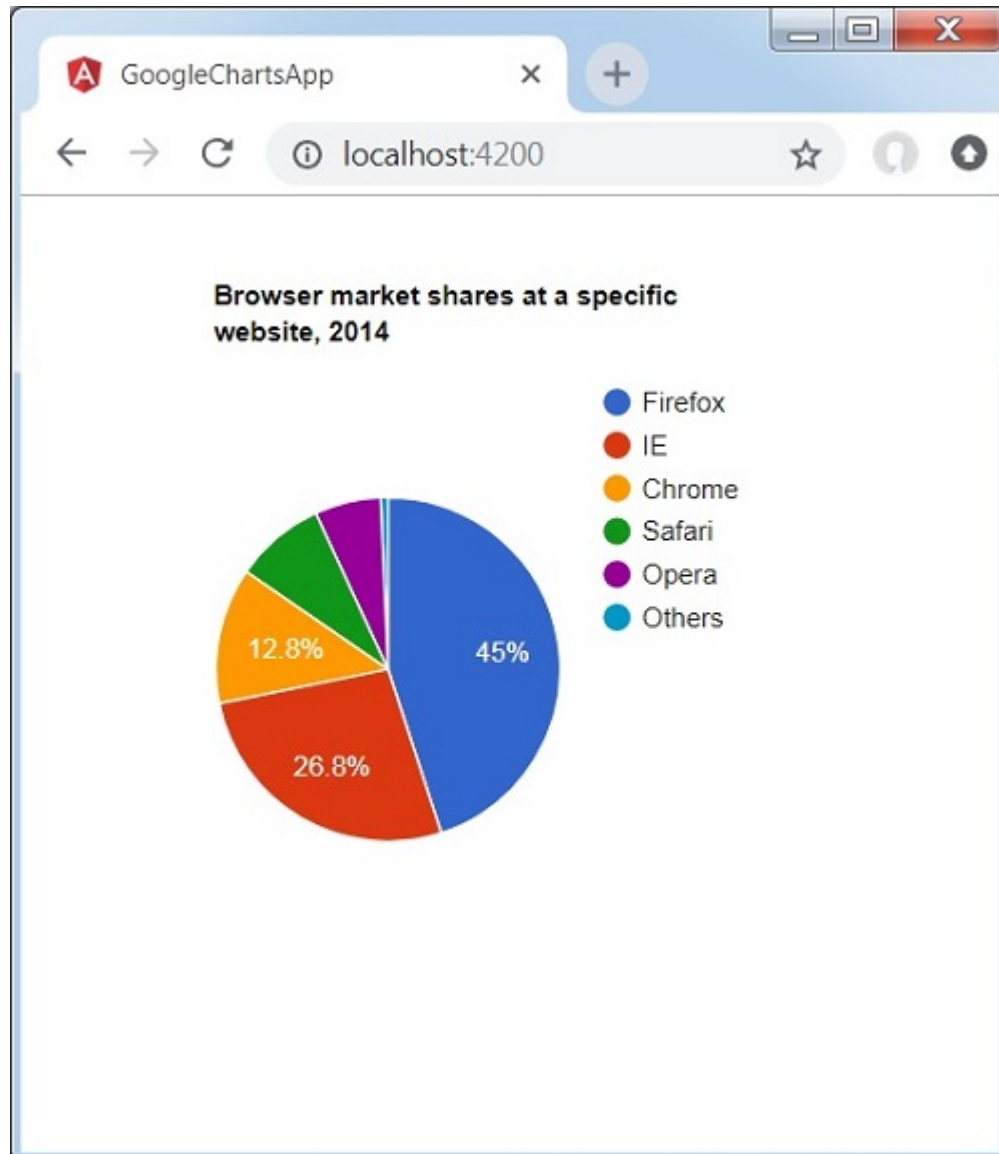
```

}))
export class AppComponent {
  title = 'Browser market shares at a specific website, 2014';
  type = 'PieChart';
  data = [
    ['Firefox', 45.0],
    ['IE', 26.8],
    ['Chrome', 12.8],
    ['Safari', 8.5],
    ['Opera', 6.2],
    ['Others', 0.7]
  ];
  columnNames = ['Browser', 'Percentage'];
  options = {
  };
  width = 550;
  height = 400;
}

```

Result

Verify the result.



Angular Google Charts - Area Charts

Area charts are used to draw area based charts. In this section we're going to discuss following types of area based charts.

Sr.No	Chart Type / Description
1	Basic Area Basic area chart
2	Area with negative values Area chart having negative values.
3	Stacked area Chart having areas stacked over one another.
4	Area with missing points Chart with missing points in the data.

Angular Google Charts - Bar Charts

Bar charts are used to draw bar based charts. In this section we're going to discuss following types of bar based charts.

Sr.No	Chart Type / Description
1	Basic Bar Basic bar chart
2	Grouped Bar Chart Grouped Bar chart.
3	Stacked Bar Bar chart having bar stacked over one another.
4	Negative Bar Bar chart with negative stack.

Angular Google Charts - Bubble Charts

Bubble charts are used to draw bubble based charts. In this section we're going to discuss following types of bubble based charts.

Sr.No	Chart Type / Description
1	Basic Bubble Basic bubble chart.
2	Bubble chart with data labels Bubble chart with data labels.

Angular Google Charts - Candlestick Charts

Candlestick charts are used to show opening and closing value over a value variance and are normally used to represent stocks. In this section we're going to discuss following types of candlestick based charts.

Sr.No	Chart Type / Description
1	Basic Candlestick Basic Candlestick chart.
2	Candlestick with customized colors Customized Candlestick Chart.

Angular Google Charts - Column Charts

Column charts are used to draw column based charts. In this section we're going to discuss following types of column based charts.

Sr.No	Chart Type / Description
1	Basic Column Basic columnn chart
2	Grouped Column Chart Grouped Columnn chart.
3	Stacked Column Columnn chart having columnn stacked over one another.
4	Negative Stacked Column Columnn chart with negative stack.

Angular Google Charts - Combination Chart

Combination chart helps in rendering each series as a different marker type from the following list: line, area, bars, candlesticks, and stepped area. To assign a default marker type for series, use the `seriesType` property. Series property is to be used to specify properties of each series individually. Following is an example of a Column Chart showing differences.

We have already seen the configurations used to draw a chart in Google Charts Configuration Syntax chapter. Now, let us see an example of a Column Chart showing differences.

Configurations

We've used **ComboChart** class to show a Combination Chart.

```
type='ComboChart';
```

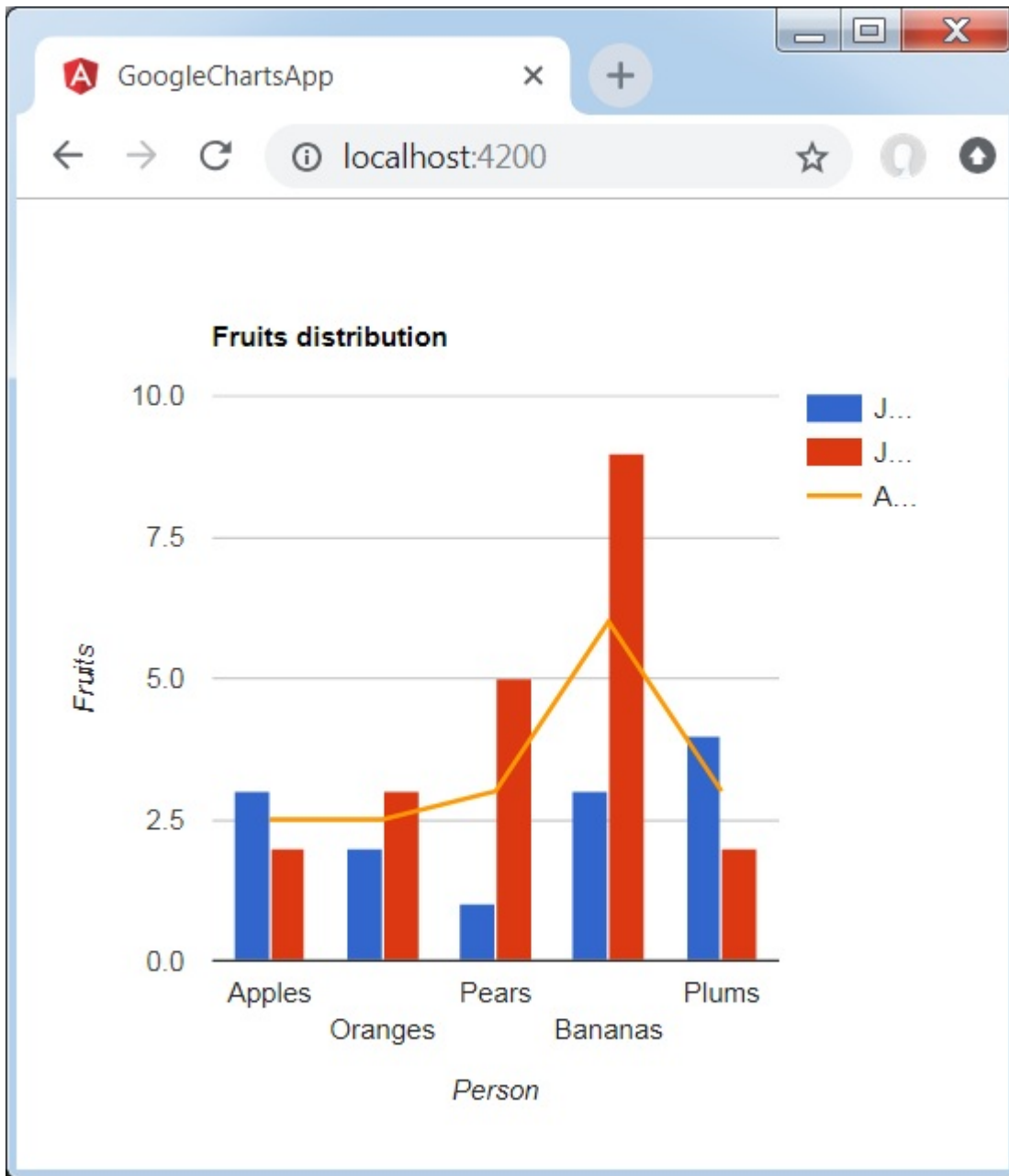
Example

app.component.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'Fruits distribution';
  type = 'ComboChart';
  data = [
    ["Apples", 3, 2, 2.5],
    ["Oranges", 2, 3, 2.5],
    ["Pears", 1, 5, 3],
    ["Bananas", 3, 9, 6],
    ["Plums", 4, 2, 3]
  ];
  columnNames = ['Fruits', 'Jane', 'Jone', 'Average'];
  options = {
    hAxis: {
      title: 'Person'
    },
    vAxis: {
      title: 'Fruits'
    },
    seriesType: 'bars',
    series: {2: {type: 'line'}}
  };
  width = 550;
  height = 400;
}
```

Result

Verify the result.



Angular Google Charts - Histogram Charts

A histogram is a chart that groups numeric data into buckets, displaying the buckets as segmented columns. They're used to depict the distribution of a dataset as how often values fall into ranges. Google Charts automatically chooses the number of buckets for you. All buckets are equal width and have a height proportional to the number of data points in the bucket. Histograms are similar to column charts in other aspects. In this section we're going to discuss following types of histogram based charts.

Sr.No	Chart Type / Description
-------	--------------------------

1	Basic Histogram Basic Histogram chart.
2	Controlling Color Customized Color of Histogram Chart.
3	Controlling Buckets Customized Buckets of Histogram Chart.
4	Multiple Series Histogram Chart having multiple series.

Angular Google Charts - Line Charts

Line charts are used to draw line based charts. In this section we're going to discuss following types of line based charts.

Sr.No	Chart Type / Description
1	Basic line Basic line chart.
2	With visible points Chart with visible data points.
3	Customizable background color Chart with customized background color.
4	Customizable line color Chart with customized line color.
5	Customizable axis and tick labels Chart with customized axis and tick labels.
6	Crosshairs Line charts showing crosshairs at data point on selection.
7	Customizable line style Chart with customized line color.
8	Line Charts with curved lines

Chart with smooth curve lines.

Angular Google Charts - Map Charts

A Google Map Chart uses Google Maps API to display Map. Data values are displayed as markers on the map. Data values may be coordinates (lat-long pairs) or actual addresses. The map will be scaled accordingly so that it includes all the identified points.

Sr.No	Chart Type / Description
1	Basic Map Basic Google Map.
2	Map using Latitude/Longitude Map having locations specified using Latitude and Longitude.

Angular Google Charts - Organization Charts

Following is an example of a Organization Chart.

Organization chart helps in rendering a hierarchy of nodes, used to portray superior/subordinate relationships in an organization. For example, A family tree is a type of org chart. We have already seen the configurations used to draw a chart in Google Charts Configuration Syntax chapter. Now, let us see an example of a Organization Chart.

Configurations

We've used **OrgChart** class to show a Organization Chart.

```
type='OrgChart';
```

Example

app.component.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = '';
  type = 'OrgChart';
  data = [
```

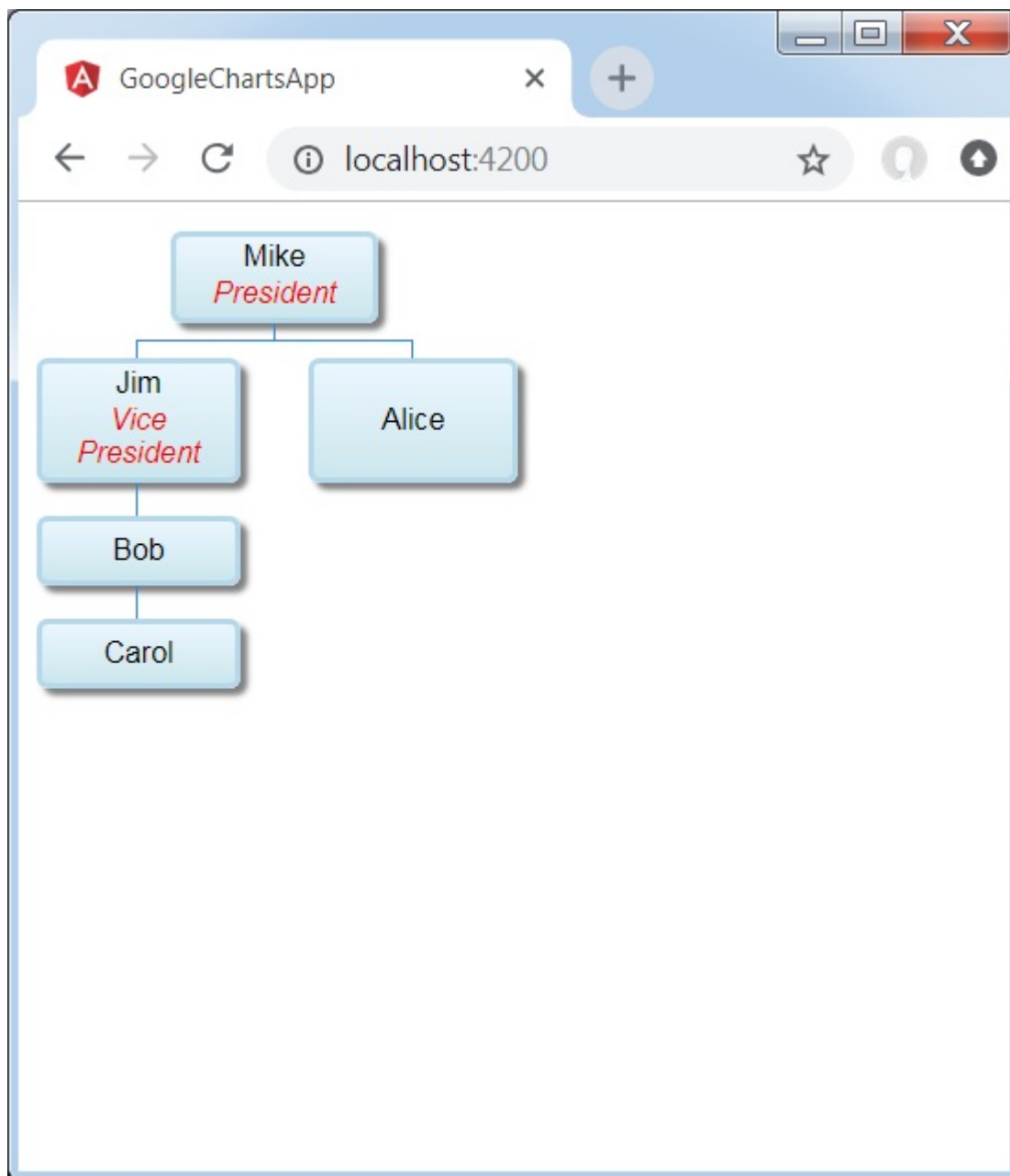
```

    [{v:'Mike', f:'Mike<div style="color:red; font-style:italic">President</div>'},
      '', 'The President'],
    [{v:'Jim', f:'Jim<div style="color:red; font-style:italic">Vice President</div>'},
      'Mike', 'VP'],
    ['Alice', 'Mike', ''],
    ['Bob', 'Jim', 'Bob Sponge'],
    ['Carol', 'Bob', '']
  ];
  columnNames = ["Name", "Manager", "Tooltip"];
  options = {
    allowHtml: true
  };
  width = 550;
  height = 400;
}

```

Result

Verify the result.



Angular Google Charts - Pie Charts

Pie charts are used to draw pie based charts. In this section we're going to discuss following types of pie based charts.

Sr.No	Chart Type / Description
1	Basic Pie Basic pie chart.
2	Donut Chart Donut Chart.
3	3D Pie chart

	3D Pie chart.
4	Pie chart with exploded slices Pie chart with exploded slices.

Angular Google Charts - Sankey Charts

A sankey chart is a visualization tool and is used to depict a flow from one set of values to another. Connected objects are called nodes and the connections are called links. Sankeys are used to show a many-to-many mapping between two domains or multiple paths through a set of stages.

Sr.No	Chart Type / Description
1	Basic Sankey Chart Basic Sankey Chart.
2	Multilevel Sankey Chart Multilevel Sankey Chart.
3	Customizing Sankey Chart Customized Sankey Chart.

Angular Google Charts - Scatter Chart

Following is an example of a Scatter Chart.

We have already seen the configurations used to draw a chart in Google Charts Configuration Syntax chapter. Now, let us see an example of a Scatter Chart.

Configurations

We've used **ScatterChart** class to show a Scatter chart.

```
type = 'ScatterChart';
```

Example

app.component.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
```

```
    styleUrls: ['./app.component.css']
  })
  export class AppComponent {
    title = 'Age vs Weight';
    type='ScatterChart';
    data = [
      [8,12],
      [4, 5.5],
      [11,14],
      [4,5],
      [3,3.5],
      [6.5,7]
    ];
    columnNames = ['Age', 'Weight'];
    options = {
    };
    width = 550;
    height = 400;
  }
```

Result

Verify the result.



Angular Google Charts - Stepped Charts

A stepped area chart is a step based area chart. We're going to discuss following types of stepped area charts.

Sr.No	Chart Type / Description
1	Basic Stepped Chart Basic Stepped Area Chart.
2	Stacked Stepped Chart Stacked Stepped Area Chart.

Angular Google Charts - Table Charts

Table chart helps in rendering a table which can be sorted and paged. Table cells can be formatted using format strings, or by directly inserting HTML as cell values. Numeric values are right-aligned by default; boolean values are displayed as check marks or cross marks. Users can select single rows either with the keyboard or the mouse. Column headers can be used for sorting. The header row remains fixed during scrolling. The table fires events corresponding to user interaction.

We have already seen the configurations used to draw a chart in Google Charts Configuration Syntax chapter. Now, let us see an example of a Table Chart.

Configurations

We've used **Table** class to show a Table chart.

```
type = 'Table';
```

Example

app.component.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = '';
  type = 'Table';
  data = [
    ['Mike', {v: 10000, f: '$10,000'}, true],
    ['Jim', {v: 8000, f: '$8,000'}, false],
    ['Alice', {v: 12500, f: '$12,500'}, true],
    ['Bob', {v: 7000, f: '$7,000'}, true]
  ];
  columnNames = ["Name", "Salary", "Full Time Employee"];
  options = {
    alternatingRowStyle: true,
    showRowNumber: true
  };
  width = 550;
  height = 400;
}
```

Result

Verify the result.

	Name	Salary	Full Time Employee
1	Mike	\$10,000	✓
2	Jim	\$8,000	x
3	Alice	\$12,500	✓
4	Bob	\$7,000	✓

Angular Google Charts - Tree Map

TreeMap is a visual representation of a data tree, where each node may have zero or more children, and one parent (except for the root). Each node is displayed as a rectangle, can be sized and colored according to values that we assign. Sizes and colors are valued relative to all other nodes in the graph. Following is an example of a treemap chart.

We have already seen the configurations used to draw a chart in Google Charts Configuration Syntax chapter. Now, let us see an example of a TreeMap Chart.

Configurations

We've used **TreeMap** class to show a TreeMap chart.

```
type = 'TreeMap';
```

Example

app.component.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = '';
  type='TreeMap';
  data = [
    ["Global",null,0,0],
    ["America","Global",0,0],
    ["Europe","Global",0,0],
    ["Asia","Global",0,0],
    ["Australia","Global",0,0],
    ["Africa","Global",0,0],

    ["USA","America",52,31],
    ["Mexico","America",24,12],
    ["Canada","America",16,-23],

    ["France","Europe",42,-11],
    ["Germany","Europe",31,-2],
    ["Sweden","Europe",22,-13],

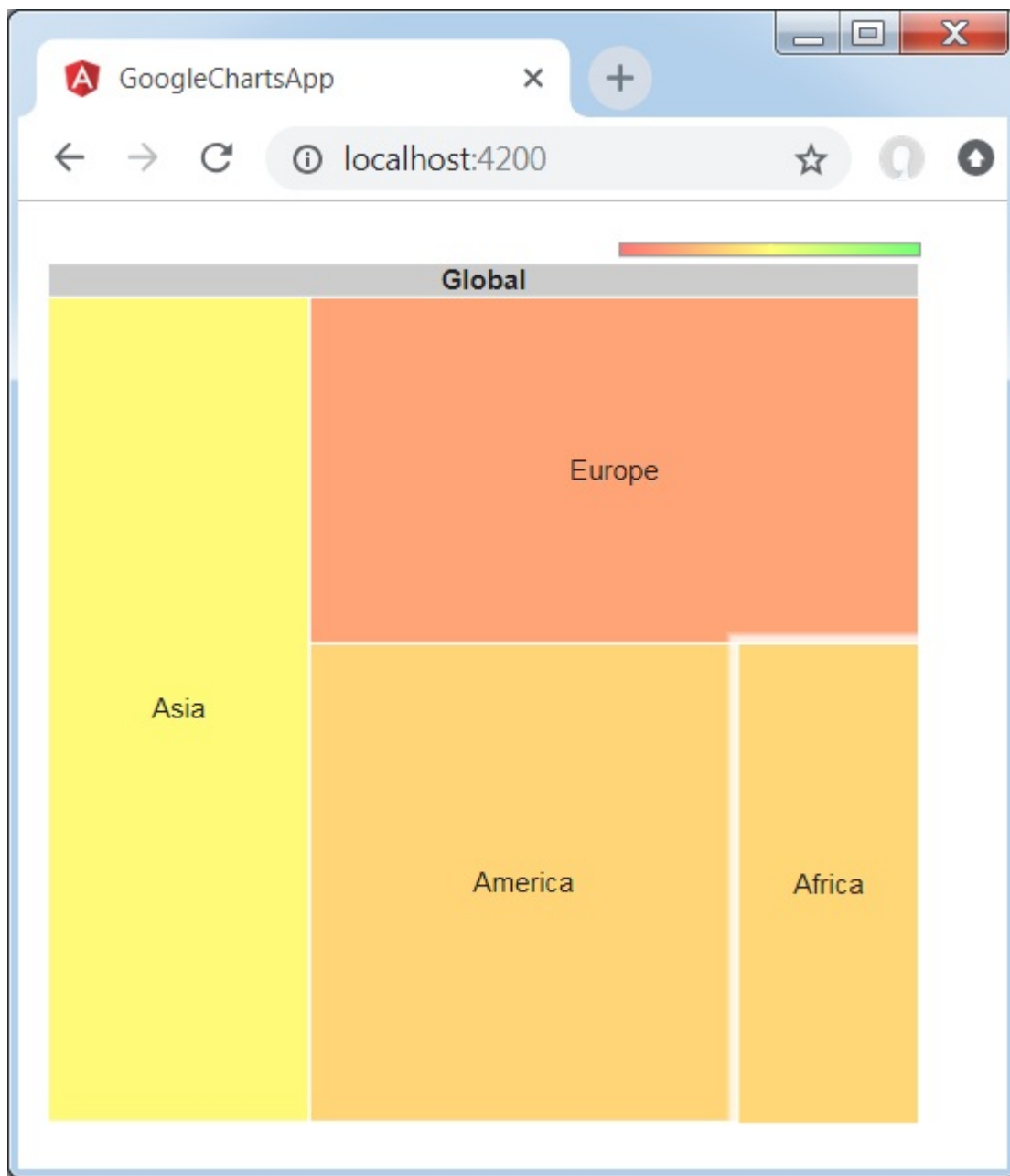
    ["China","Asia",36,4],
    ["Japan","Asia",20,-12],
    ["India","Asia",40,63],

    ["Egypt","Africa",21,0],
    ["Congo","Africa",10,12],
    ["Zaire","Africa",8,10],

  ];
  columnNames = ["Location", "Parent","Market trade volume (size)","Market increase/decrease (co
  options = {
    minColor:"#ff7777",
    midColor:'#ffff77',
    maxColor:'#77ff77',
    headerHeight:15,
    showScale:true
  };
  width = 550;
  height = 400;
}
```

Result

Verify the result.



[⬅ Previous Page](#)

[Next Page ➡](#)

Advertisements



[FAQ's](#) [Cookies Policy](#) [Contact](#)

© Copyright 2018. All Rights Reserved.