

13. D3JS Paths API.

Paths are used to draw Rectangles, Circles, Ellipses, Polylines, Polygons, Straight Lines, and Curves. SVG Paths represent the outline of a shape that can be Stroked, Filled, Used as a Clipping Path, or any combination of all three. This chapter explains Paths API in detail.

13.1. Configuring API

You can configure the API directly using the script below.

Example:

```
<script src = "https://d3js.org/d3-path.v1.min.js"></script>
<body>
  <script>
    </script>
</body>
```

13.2. Paths API Methods

Some of the most commonly used Paths API methods are briefly described as follows.

- **d3.path()** – This method is used to create a new path.
- **path.moveTo(x, y)** – This method is used to move the specified x and y values.
- **path.closePath()** – This method is used to close the current path.
- **path.lineTo(x, y)** – This method is used to create a line from current point to defined x,y values.

- **path.quadraticCurveTo(cpx, cpy, x, y)** – This method is used to draw a quadratic curve from current point to the specified point.
- **path.bezierCurveTo(cpx1, cpy1, cpx2, cpy2, x, y)** – This method is used to draw a bezier curve from current point to the specified point.
- **path.arcTo(x1, y1, x2, y2, radius)** – This method is used to draw a circular arc from the current point to a specified point (x1, y1) and end the line between the specified points (x1, y1) and (x2, y2).
- **path.arc(x, y, radius, startAngle, endAngle[, anticlockwise])** – This method is used to draw a circular arc to the specified center (x, y), radius, startAngle and endAngle. If anticlockwise value is true then the arc is drawn in the anticlockwise direction, otherwise it is drawn in the clockwise direction.
- **path.rect(x, y, w, h)** – This method is used to create new sub path containing just the four points (x, y), (x + w, y), (x + w, y + h), (x, y + h). With these four points connected by straight lines marks the subpath as closed. Equivalent to context.rect and uses SVG's "lineto" commands.
- **path.toString()** – Returns the string representation of this path according to SVG's path data specification.

Let us go through each of these API in detail.

Example: Let us draw a simple line in D3 using path API. Create a webpage linepath.html and add the following changes in it.

```
<!DOCTYPE html>
<meta charset = "UTF-8">
  <head>
    <title>SVG path line Generator</title>
  </head>

  <style>
    path {
      fill: green;
      stroke: #aaa;
    }
  </style>

  <body>
    <svg width = "600" height = "100">
      <path transform = "translate(200, 0)" />
    </svg>

    <script src = "https://d3js.org/d3.v4.min.js"></script>
    <script>
      var data = [[0, 20],[50, 30],[100, 50],[200, 60],[300, 90]];
      var lineGenerator = d3.line();
      var pathString = lineGenerator(data);
      d3.select('path').attr('d', pathString);
    </script>

  </body>
</html>
```

Now, request the browser and we will see the following result.

Output:

