

20. D3JS Zooming API.

Zooming helps to scale your content. You can focus on a particular region using the click-and-drag approach. In this chapter, we will discuss Zooming API in detail.

20.1. Configuring API

You can load the Zooming API directly from the “d3js.org” using the following script.

Example:

```
<script src = "https://d3js.org/d3-color.v1.min.js"></script>
<script src = "https://d3js.org/d3-dispatch.v1.min.js"></script>
<script src = "https://d3js.org/d3-ease.v1.min.js"></script>
<script src = "https://d3js.org/d3-interpolate.v1.min.js"></script>
<script src = "https://d3js.org/d3-selection.v1.min.js"></script>
<script src = "https://d3js.org/d3-timer.v1.min.js"></script>
<script src = "https://d3js.org/d3-transition.v1.min.js"></script>
<script src = "https://d3js.org/d3-drag.v1.min.js"></script>
<script src = "https://d3js.org/d3-zoom.v1.min.js"></script>
```

20.2. Zooming API Methods

Following are some of the most important zooming API methods in D3.js.

- **d3.zoom()**
- **zoom(selection)**
- **zoom.transform(selection, transform)**
- **zoom.translateBy(selection, x, y)**
- **zoom.translateTo(selection, x, y)**
- **zoom.scaleTo(selection, k)**
- **zoom.scaleBy(selection, k)**

- **zoom.filter([filter])**
- **zoom.wheelDelta([delta])**
- **zoom.extent([extent])**
- **zoom.scaleExtent([extent])**
- **zoom.translateExtent([extent])**
- **zoom.clickDistance([distance])**
- **zoom.duration([duration])**
- **zoom.interpolate([interpolate])**
- **zoom.on(typenames[, listener])**

Let us now understand each of these in detail.

d3.zoom()

It creates a new zoom behavior. We can access it using the script below.

Example:

```
<script>  
  var zoom = d3.zoom();  
</script>
```

zoom(selection)

It is used to apply the zoom transformation on a selected element. For example, you can instantiate a `mousedown.zoom` behavior using the following syntax.

Example:

```
selection.call(d3.zoom().on("mousedown.zoom", mousedowned));
```

zoom.transform(selection, transform)

It is used to set the current zoom transform of the selected elements to the specified transform. For example, we can reset the zoom transform to the identity transform using the syntax below.

Example:

```
selection.call(zoom.transform, d3.zoomIdentity);
```

We can also reset the zoom transform to the identity transform for 1000 milliseconds using the following syntax.

Example:

```
selection.transition().duration(1000).call(zoom.transform,  
d3.zoomIdentity);
```

zoom.translateBy(selection, x, y)

It is used to translate the current zoom transform of the selected elements by x and y values. You can specify x and y translation values either as numbers or as functions that returns numbers. If a function is invoked for the selected element, then it is passed through the current datum 'd' and index 'i' for DOM. A sample code is defined below.

Example: Let us consider the following example.

```
zoom.translateBy(selection, x, y) {  
  zoom.transform(selection, function() {  
    return constrain(this.__zoom.translate(  
      x === "function" ? x.apply(this, arguments) : x,  
      y === "function" ? y.apply(this, arguments) : y  
    ));  
  });  
}
```

```
};
```

zoom.translateTo(selection, x, y)

It is used to translate the current zoom transform of the selected elements to the specified position of x and y.

zoom.scaleTo(selection, k)

It is used to scale the current zoom transform of the selected elements to k. Here, k is a scale factor, specified as numbers or functions.

Example:

```
zoom.scaleTo = function(selection, k) {  
    zoom.transform(selection, function() {  
        k = == "function" ? k.apply(this, arguments) : k;  
    });  
};
```

zoom.scaleBy(selection, k)

It is used to scale the current zoom transform of the selected elements by k. Here, k is a scale factor, specified either as numbers or as functions that returns numbers.

Example:

```
zoom.scaleBy = function(selection, k) {  
    zoom.scaleTo(selection, function() {  
        var k0 = this.__zoom.k,  
            k1 = k == "function" ? k.apply(this, arguments) : k;  
        return k0 * k1;  
    });  
};
```

zoom.filter([filter])

It is used to set the filter to the specified function for zoom behavior. If the filter is not specified, it returns the current filter as shown below.

Example:

```
function filter() {  
    return !d3.event.button;  
}
```

zoom.wheelDelta([delta])

The value of Δ is returned by the wheel delta function. If delta is not specified, it returns the current wheel delta function.

zoom.extent([extent])

It is used to set the extent to the specified array points. If the extent is not specified, it returns the current extent accessor, which defaults to $[[0, 0], [\text{width}, \text{height}]]$, where width is the client width of the element and height is its client height.

zoom.scaleExtent([extent])

It is used to set the scale extent to the specified array of numbers $[k_0, k_1]$. Here, k_0 is the minimum allowed scale factor. While, k_1 is the maximum allowed scale factor. If extent is not specified, it returns the current scale extent, which defaults to $[0, \infty]$. Consider the sample code that is defined below.

Example:

```
selection
  .call(zoom)
  .on("wheel", function() { d3.event.preventDefault(); });
```

The user can try to zoom by wheeling, when already at the corresponding limit of the scale extent. If we want to prevent scrolling on wheel input regardless of the scale extent, register a wheel event listener to prevent the browser default behavior.

zoom.translateExtent([extent])

If the extent is specified, it sets the translate extent to the specified array of points. If extent is not specified, it returns the current translate extent, which defaults to $[[-\infty, -\infty], [+\infty, +\infty]]$.

zoom.clickDistance([distance])

This method is used to set the maximum distance that the zoomable area can move between up and down, which will trigger a subsequent click event.

zoom.duration([duration])

This method is used to set the duration for zoom transitions on double-click and double-tap to the specified number of milliseconds and returns the zoom behavior. If the duration is not specified, it returns the current duration, which defaults to 250 milliseconds, which is defined below.

Example:

```
selection
  .call(zoom)
  .on("dblclick.zoom", null);
```

zoom.interpolate([interpolate])

This method is used to interpolate for zoom transitions to the specified function. If interpolate is not specified, it returns the current interpolation factory, which defaults to d3.interpolateZoom.

zoom.on(typenames[, listener])

If the listener is specified, it sets the event listener for the specified typenames and returns the zoom behavior. The typenames is a string containing one or more typename separated by whitespace. Each typename is a type, optionally followed by a period (.) and a name, such as zoom.one and zoom.second. The name allows multiple listeners to be registered for the same type. This type must be from one of the following –

- **Start** – after zooming begins (such as on mousedown).
- **Zoom** – after a change to the zoom transform (such as on mousemove).
- **End** – after zooming ends (such as on mouseup).

In the next chapter, we will discuss the different requests API in D3.js.