19. D3JS Dragging API.

Drag and drop is one of the most familiar concept in d3.js. This chapter explains dragging and its methods in detail.

19.1. Configuring API

You can configure the API directly using the script below.

Example:

```
<script src = "https://d3js.org/d3-dispatch.v1.min.js"></script>
<script src = "https://d3js.org/d3-selection.v1.min.js"></script>
<script src = "https://d3js.org/d3-drag.v1.min.js"></script>
```

19.2. Dragging API Methods

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

- **d3.drag()**
- **drag(selection)**
- **drag.container([container])**
- **drag.filter([filter])**
- **drag.subject([subject])**
- **drag.clickDistance([distance])**
- **drag.on(typenames, [listener])**
- **d3.dragDisable(window)**
- **d3.dragEnable(window[, noclick])**

Let us now understand each of these in detail.

d3.drag()

This method is used to create a new dragging. You can call this method using the following script.

Example:

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

drag(selection)

This method is used to apply the dragging to the specified selection. You can invoke this function using selection.call. A simple example is defined below.

Example:

```
d3.select(".node").call(d3.drag().on("drag", mousemove));
```

Here, the drag behavior applied to the selected elements is via selection.call.

Example:

```
drag.container([container])
```

It is used to set the container to the specified function for dragging. If a container is not specified, it returns the current accessor. To drag any graphical elements with a Canvas, you can redefine the container as itself. It is defined below.

Example:

```
function container() {
   return this;
}
```

drag.filter([filter])

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

Example:

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

drag.subject([subject])

It is used to set the subject to the specified function for dragging and is defined below.

Example: Let us consider the following example.

```
function subject(d) {
   return d = =  null ? {x: d3.event.x, y: d3.event.y} : d;
}
```

Here, the subject represents the thing being dragged. For example, if you want to drag rectangle elements in SVG, the default subject is datum of the rectangle being dragged.

drag.clickDistance([distance])

This method is used to set the maximum distance for clicking a mousedown and mouseup event. If distance is not specified, it points to zero.

drag.on(typenames, [listener])

This method is used to set the event listener for the specified typenames for dragging. 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 drag.one and drag.two. This type should be from one of the following :

- **start** − starts a new pointer.
- **drag** − drags an active pointer.
- **end** − Inactive an active pointer.

d3.dragDisable(window)

This method is used to disable the drag and drop selection. It prevents mousedown event action. Most of the selected browsers supports this action by default. If not supported, you can set the CSS property to none.

d3.dragEnable(window[, noclick])

This method is used to enable the drag and drop selection on the specified window location. It is used to call mouseup event action. If you assign the noclick value is true then click event expires a zero millisecond timeout.

19.4. Dragging API - Drag Events

The D3.event method is used to set the drag event. It consists of the following fields −

- **Target** − It represents the drag behavior.
- **Type** − It is a string and can be any one of the following– "start", "drag" or "end".
- **Subject** − The drag subject, defined by drag.subject.

event.on(typenames, [listener])

The event object exposes the event.on method to perform dragging. It is defined as follows.

Example:

```
d3.event.on("drag", dragged).on("end", ended);
```