

Graph Daten hervorheben

Dies ist die Kopie eines Gesprächs zwischen ChatGPT und einem anonymen Benutzer.



```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Draggable Graph</title>
  <script src="https://d3js.org/d3.v7.min.js"></script>
  <style>
    body, html {
      margin: 0;
      padding: 0;
      overflow: hidden;
    }
    svg {
      width: 100vw;
      height: 100vh;
      background-color: #f9f9f9;
    }
    .node {
      fill: black;
      cursor: pointer;
    }
    .link {
      fill: none;
      stroke: #ccc;
      stroke-width: 1.5px;
    }
    text {
      font: 12px sans-serif;
      pointer-events: none;
      fill: black;
    }
    .highlight {
      fill: blue;
    }
    .connected {
      fill: green;
    }
    .highlight-link {
      stroke: orange;
      stroke-width: 2.5px;
    }
  </style>
```

```

</head>
<body>
<svg></svg>

<script>
  d3.json('graphSpecific.json').then(function(graphData) {
    const nodes = [];
    const links = [];
    const nodeMap = {};

    function traverse(subs, parentId) {
      subs.forEach(sub => {
        if (!nodeMap[sub.contId]) {
          nodes.push({ id: sub.contId });
          nodeMap[sub.contId] = true;
        }
        links.push({ source: parentId, target: sub.contId });
        if (sub.subs && sub.subs.length > 0) {
          traverse(sub.subs, sub.contId);
        }
      });
    }

    nodes.push({ id: graphData.contId });
    nodeMap[graphData.contId] = true;

    if (graphData.subs) {
      traverse(graphData.subs, graphData.contId);
    }

    Object.keys(nodeMap).forEach(id => {
      if (!nodes.find(node => node.id === id)) {
        nodes.push({ id });
      }
    });

    const svg = d3.select('svg');
    const width = window.innerWidth;
    const height = window.innerHeight;

    const simulation = d3.forceSimulation(nodes)
      .force('link', d3.forceLink(links).id(d => d.id).distance(100))
      .force('charge', d3.forceManyBody().strength(-300))
      .force('center', d3.forceCenter(width / 2, height / 2));

    const link = svg.append('g')
      .selectAll('.link')
      .data(links)
      .enter().append('line')
      .attr('class', 'link');
  });

```

```

const node = svg.append('g')
  .selectAll('.node')
  .data(nodes)
  .enter().append('g')
  .attr('class', 'node')
  .call(d3.drag()
    .on('start', dragStarted)
    .on('drag', dragged)
    .on('end', dragEnded))
  .on('click', highlightNode);

node.append('circle')
  .attr('r', 20)
  .attr('class', 'node');

node.append('text')
  .attr('dy', -25)
  .attr('text-anchor', 'middle')
  .text(d => d.id);

simulation.on('tick', function() {
  link
    .attr('x1', d => d.source.x)
    .attr('y1', d => d.source.y)
    .attr('x2', d => d.target.x)
    .attr('y2', d => d.target.y);

  node
    .attr('transform', d => `translate(${d.x},${d.y})`);
});

function dragStarted(event, d) {
  if (!event.active) simulation.alphaTarget(0.3).restart();
  d.fx = d.x;
  d.fy = d.y;
}

function dragged(event, d) {
  d.fx = event.x;
  d.fy = event.y;
}

function dragEnded(event, d) {
  if (!event.active) simulation.alphaTarget(0);
  d.fx = null;
  d.fy = null;
}

function highlightNode(event, selectedNode) {
  node.selectAll('circle').classed('highlight', false).classed('connected', false);
  link.classed('highlight-link', false);
}

```

```
// Knoten hervorheben
d3.select(event.currentTarget).select('circle').classed('highlight', true);

// Verbindungen hervorheben
link.filter(d => d.source.id === selectedNode.id || d.target.id === selectedNode.id)
  .classed('highlight-link', true);

// Verbundene Knoten hervorheben
node.filter(d =>
  links.some(link =>
    (link.source.id === selectedNode.id && link.target.id === d.id) ||
    (link.target.id === selectedNode.id && link.source.id === d.id)
  ))
  .select('circle')
  .classed('connected', true);
}
}).catch(function(error) {
  console.log(error);
});
</script>
</body>
</html>
```

Kannst du zu diesem Skript die Möglichkeit hinzufügen, dass über die URL ein gewisser Name mit übergeben werden kann (z.B. /cont0) und dieser dann automatisch farblich markiert wird?

Hier ist dein überarbeitetes Skript. Es liest einen Parameter aus der URL (z. B. `?highlight=cont0`) und hebt den entsprechenden Knoten automatisch hervor.

Jetzt kannst du in der URL einen Parameter wie `?highlight=cont0` angeben, und der entsprechende Knoten wird automatisch hervorgehoben. 😊



Ab hier sind Nachrichten nur für dich sichtbar