

1. Schema-Konformität

Dein JSON **muss** exakt dem Nodges-Schema entsprechen:

- **Keine Kommentare:** JSON erlaubt keine Zeilen, die mit // oder /* */ beginnen.
- **Kein Komma hinter dem letzten Listen-Eintrag** ([...] oder { ... }).
- **Alle Keys und Strings in Anführungszeichen** ("").
- **Pflichtfelder:** system, metadata, visualMappings, data mit entities und relationships.

Beispiel für einen Knoten:

```
{ "id": "n1", "type": "Gehirnregion", "label": "Großhirn", "position": { "x": -10, "y": 10, "z": 0 } }
```

2. Struktur deines JSON

Wichtige Abschnitte:

- **"system"**: Name des Graphs.
- **"metadata"**: Metadaten (Erstelldatum, Autor, Version, Beschreibung).
- **"visualMappings"**: Farb-, Größen- und Animationsdefinitionen für Knotentypen und Kantentypen.
- **"data"**:
 - **"entities"**: Knoten (Definition von über 100 Knoten für einen großen Graphen empfohlen).
 - **"relationships"**: Kanten (Verknüpfung der Knoten, Animationen für z.B. Nervenimpulse empfohlen).

3. Visuelle Mappings

Jeder Knotentyp und Beziehungstyp braucht einen eigenen Eintrag in **"visualMappings.defaultPresets"** mit eindeutiger Farbe usw.

Zum Beispiel:

```
"ZentralesNervensystem": {
  "color": { "source": "constant", "function": "linear", "params": { "color": "#2277BB" } },
  "size": { "source": "constant", "function": "linear", "range": [4.5, 4.5] }
}
```

Für Kanten mit Signal-Charakter:

```
"SignalUebertragung": {
  "color": { "source": "constant", "function": "linear", "params": { "color": "#FF2222" } },
```

```

    "thickness": { "source": "constant", "function": "linear", "range": [0.07, 0.07]
  },
  "animation": { "source": "constant", "function": "pulse", "params": {
    "frequency": 1.5
  }
}

```

4. Best Practices und häufige Fehler

- **Keine Kommentare einfügen.**
- **Klammern und Kommas:** Prüfe, dass Listen und Objekte korrekt mit `]` und `}` enden und dass nach dem letzten Listen-Element **kein Komma steht**.
- **Jeder Entität und Beziehung einen eindeutigen `id`-Wert zuweisen.**
- **Nur Strings, Zahlen und Arrays als Werte zulassen.**
- **Verwende für große Graphen unterschiedliche Positionen** im 3D-Raum, damit Knoten nicht überlappen.
- **Jeder Knotentyp und jede Beziehung** braucht einen eigenen Mapping-Eintrag (sonst werden sie grau/blau oder unbelebt angezeigt).
- **Verwende für animierte Kanten** eine `animation` mit `pulse`.
- **Jede Kante muss einen Typ und jeweils eine "source" und einen "target"-Knoten angeben.**

5. Minimales valides JSON-Beispiel

```
{
  "system": "Beispiel Nervensystem",
  "metadata": {
    "created": "2025-12-20T12:00:00Z",
    "version": "1.0",
    "author": "AI",
    "description": "Ein minimales valides Beispiel für Nodges."
  },
  "visualMappings": {
    "defaultPresets": {
      "Gehirnregion": {
        "color": { "source": "constant", "function": "linear", "params": {
          "color": "#ffcc00" } },
        "size": { "source": "constant", "function": "linear", "range": [2.0, 2.0]
      }
    },
    "SignalUebertragung": {
      "color": { "source": "constant", "function": "linear", "params": {
        "color": "#ff0000" } },
      "thickness": { "source": "constant", "function": "linear", "range": [0.2,
        0.2] },
      "animation": { "source": "constant", "function": "pulse", "params": {
        "frequency": 2.0
      }
    }
  }
}
```

```
"data": {  
    "entities": [  
        { "id": "n1", "type": "Gehirnregion", "label": "Großhirn", "position": {  
            "x": 0, "y": 0, "z": 0 } },  
        { "id": "n2", "type": "Gehirnregion", "label": "Kleinhirn", "position": {  
            "x": 10, "y": 0, "z": 0 } }  
    ],  
    "relationships": [  
        { "id": "e1", "type": "SignalUebertragung", "source": "n1", "target": "n2",  
            "label": "Signal-Transport" }  
    ]  
}
```