

JS Condition Checker



Hi I need a JS Script that does the following:
 you get this:
 A String Sentence that looks like one of these:
 Air Pressure > 101.32500 = Medium
 Humidity < 50.00000 = Low
 Temperature == 25.00000 = Ok
 Vibration != 9.80000 = Low
 Altitude >= 100.00000 = Ok
 Latitude > 45.00000 = Critical
 Longitude < 90.00000 = High

And one line of an array like this:

```
const [tableData, setTableData] = useState([
  { environment: "Temperature", value: "-", unit: "°C", alert: "" },
  { environment: "Pressure", value: "-", unit: "Pa", alert: "" },
  { environment: "Humidity", value: "-", unit: "%", alert: "" },
  { environment: "Vibration", value: "-", unit: "m/s2", alert: "" },
  { environment: "Altitude", value: "-", unit: "m", alert: "" },
  { environment: "Latitude", value: "-", unit: "DD", alert: "" },
  { environment: "Longitude", value: "-", unit: "DD", alert: "" },
]);
```

For each line i want you to take the value of the Array then check with the sentence that has the correct parameter (like for environment Pressure, take the sentence that starts with Air Pressure.

For example you choose this line:
 Air Pressure > 101.32500 = Medium

because this says Pressure: { environment: "Pressure", value: "200", unit: "Pa", alert: "" },
 Now you do something like this:
 you replace "Air Pressure" with value (200):
 200 > 101.32500 = Medium
 If the statement is true you return "Medium" or whatever there stands if it is not true, you return
 Nothing



You can create a function that iterates over your `tableData`` and matches the ``environment`` from the data to the corresponding sentence. It will then evaluate the condition and return the appropriate alert value. Here's how you can implement this in JavaScript:

javascript

```
// Sample sentence array
const sentences = [
  "Air Pressure > 101.32500 = Medium",
  "Humidity < 50.00000 = Low",
  "Temperature == 25.00000 = Ok",
  "Vibration != 9.80000 = Low",
  "Altitude >= 100.00000 = Ok",
  "Latitude > 45.00000 = Critical",
  "Longitude < 90.00000 = High"
];

// Array of data
const tableData = [
  { environment: "Temperature", value: "200", unit: "°C", alert: "" },
  { environment: "Pressure", value: "200", unit: "Pa", alert: "" },
  { environment: "Humidity", value: "45", unit: "%", alert: "" },
  { environment: "Vibration", value: "9.8", unit: "m/s2", alert: "" },
  { environment: "Altitude", value: "120", unit: "m", alert: "" },
  { environment: "Latitude", value: "50", unit: "DD", alert: "" },
  { environment: "Longitude", value: "85", unit: "DD", alert: "" },
];

// Function to check the sentence and match conditions
const checkConditions = (tableData, sentences) => {
  return tableData.map((data) => {
    // Find the sentence that matches the environment
    const sentence = sentences.find((sentence) => sentence.toLowerCase().includes(data.environment.toLowerCase()));

    if (!sentence) {
      return { ...data, alert: "" }; // No matching sentence found
    }

    // Extract condition and alert from sentence
    const [condition, result] = sentence.split(" = ");
    const conditionParts = condition.split(" ");
    const parameterValue = parseFloat(data.value);
    const comparator = conditionParts[1];
    const conditionValue = parseFloat(conditionParts[2]);

    // Evaluate condition
    let isValid = false;

    switch (comparator) {
      case ">":
        isValid = parameterValue > conditionValue;
        break;
      case "<":
        isValid = parameterValue < conditionValue;
        break;
      case "==":
        isValid = parameterValue === conditionValue;
        break;
      case "!=":
        isValid = parameterValue !== conditionValue;
        break;
      case ">=":
        isValid = parameterValue >= conditionValue;
        break;
      case "<=":
        isValid = parameterValue <= conditionValue;
        break;
      default:
        isValid = false;
        break;
    }

    return { ...data, alert: isValid ? result : "" };
  });
};
```

```
// Call the function and log the results
const updatedTableData = checkConditions(tableData, sentences);
console.log(updatedTableData);
```

Explanation:

1. ``sentences` Array`: Holds the conditions as strings.
2. ``tableData` Array`: Represents your data, with environmental parameters (like Temperature, Pressure) and their values.
3. ``checkConditions` Function`:
 - For each item in ``tableData``, it looks for a matching sentence based on the ``environment`` property.
 - It then extracts the comparison operator (``>``, ``<``, ``==``, etc.) and evaluates whether the condition is true.
 - If the condition is true, it sets the ``alert`` property to the result from the sentence (e.g., "Medium", "Low").
 - If the condition is false, it sets the ``alert`` property to an empty string.
4. **Result**: The function returns the ``tableData`` array with updated ``alert`` properties based on the conditions.

Example Output:

json

```
[
  { "environment": "Temperature", "value": "200", "unit": "°C", "alert": "" },
  { "environment": "Pressure", "value": "200", "unit": "Pa", "alert": "Medium" },
  { "environment": "Humidity", "value": "45", "unit": "%", "alert": "Low" },
  { "environment": "Vibration", "value": "9.8", "unit": "m/s2", "alert": "" },
  { "environment": "Altitude", "value": "120", "unit": "m", "alert": "Ok" },
  { "environment": "Latitude", "value": "50", "unit": "DD", "alert": "Critical" },
  { "environment": "Longitude", "value": "85", "unit": "DD", "alert": "High" }
]
```

This script will dynamically check each environment value against the sentence and return the appropriate alert or nothing if the condition doesn't hold.



Kannst du sehr kurz in FLIEßTEXT erklären, wie das Script funktioniert?



Das Script nimmt die Umweltdaten aus dem ``tableData``-Array und prüft für jede Zeile, ob es einen passenden Satz aus der ``sentences``-Liste gibt, basierend auf dem ``environment``-Wert. Wenn ein passender Satz gefunden wird, wird die Bedingung aus dem Satz extrahiert (z. B. Vergleichsoperator und Grenzwert) und mit dem Wert aus ``tableData`` verglichen. Ist die Bedingung erfüllt, wird das Ergebnis (z. B. "Medium" oder "Low") in das ``alert``-Feld der Zeile geschrieben. Wenn die Bedingung nicht erfüllt ist oder kein passender Satz gefunden wird, bleibt das ``alert``-Feld leer.