

Thresholds Data Fetch Fix

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



```
useEffect(() =>{
  const fetchAndCheck = async () => {
    const currentAlerts = [];
    try {
      for(const containerId of containerIds){
        const sentences = [];

        // Fetch serial Number of Id
        const containerSerialNumberResponse = await
        axiosInstance.get(`/rest/container/${containerId.id}/serial-number`);
        const fetchedSerialNumber = containerSerialNumberResponse.data.serial_number;

        // Fetch thresholds of id
        const threshHoldsResponse = await
        axiosInstance.get(`/rest/container/${containerId.id}/thresholds`);
        const fetchedThresholds = threshHoldsResponse.data?.thresholds || [];

        let thresholds = fetchedThresholds;

        const validThresholds = Array.isArray(thresholds) ? thresholds : [];

        // Fetch additional data for thresholds
        if (validThresholds.length > 0) {
          for (let i = 0; i < validThresholds.length; i++) {
            // Parameter
            const parameterId = validThresholds[i].parameter;
            const parameterResponse = await
            axiosInstance.get(`/rest/threshold/parameter/${parameterId}`);
            if(parameterResponse.data.parameter[0].name === "Air Pressure"){
              const parameterValue = "Air-Pressure"
            }else{
              const parameterValue = parameterResponse.data.parameter[0].name;
            }
            // Rule
            const ruleId = validThresholds[i].rule;
            const ruleResponse = await axiosInstance.get(`/rest/threshold/rule/${ruleId}`);
            const ruleValue = ruleResponse.data.rule[0].name;

            // Level
            const levelId = validThresholds[i].level;
            const levelResponse = await axiosInstance.get(`/rest/threshold/level/${levelId}`);
            const levelValue = levelResponse.data.level[0].name;
```

```

// Sentence
const sentence = `${parameterValue} ${ruleValue} ${validThresholds[i].value} =
${levelValue}`;
sentences.push(sentence);
}

}
currentAlerts.push(checkForAlerts(sentences, fetchedSerialNumber));
sentences.length = 0;
}
setAlerts(currentAlerts);
} catch (error) {
console.error("Failed to fetch containers of ship:", error.message);
}

```

Fix the code

Your code has a couple of issues, including variable scope and proper handling of the `parameterValue`. I'll help fix it for you:

1. `parameterValue` is defined within an `if` block, making it inaccessible in the outer scope. To fix that, move it outside the block and ensure it retains its value.
2. Make sure `checkForAlerts` is properly used as a function.
3. Minor improvements like clearing `sentences` properly after each iteration.

Here's the corrected version:

```

javascript

useEffect(() => {
  const fetchAndCheck = async () => {
    const currentAlerts = [];
    try {
      for (const containerId of containerIds) {
        const sentences = [];

        // Fetch serial Number of Id
        const containerSerialNumberResponse = await axiosInstance.get(`/rest/container/${containerId}`);
        const fetchedSerialNumber = containerSerialNumberResponse.data.serial_number;

        // Fetch thresholds of id
        const threshHoldsResponse = await axiosInstance.get(`/rest/container/${containerId.id}/thresholds`);
        const fetchedThresholds = threshHoldsResponse.data?.thresholds || [];

        let thresholds = fetchedThresholds;
        const validThresholds = Array.isArray(thresholds) ? thresholds : [];

        // Fetch additional data for thresholds
        if (validThresholds.length > 0) {
          for (let i = 0; i < validThresholds.length; i++) {
            // Parameter
            const parameterId = validThresholds[i].parameter;
            const parameterResponse = await axiosInstance.get(`/rest/threshold/parameter/${parameterId}`);
            let parameterValue = parameterResponse.data.parameter[0].name;

            if (parameterValue === "Air Pressure") {
              parameterValue = "Air-Pressure";
            }

            // Rule
            const ruleId = validThresholds[i].rule;

```

```

const ruleResponse = await axiosInstance.get(`/rest/threshold/rule/${ruleId}`);
const ruleValue = ruleResponse.data.rule[0].name;

// Level
const levelId = validThresholds[i].level;
const levelResponse = await axiosInstance.get(`/rest/threshold/level/${levelId}`);
const levelValue = levelResponse.data.level[0].name;

// Sentence
const sentence = `${parameterValue} ${ruleValue} ${validThresholds[i].value} = ${levelValue}`;
sentences.push(sentence);
}
}

// Check for alerts after collecting sentences for the container
currentAlerts.push(checkForAlerts(sentences, fetchedSerialNumber));
sentences.length = 0; // Clear sentences array for the next iteration
}
setAlerts(currentAlerts);
} catch (error) {
console.error("Failed to fetch containers of ship:", error.message);
}
};

fetchAndCheck();
}, [containerIds]); // Ensure to add any necessary dependencies

```

Key fixes:

- Moved `parameterValue` assignment outside of the `if` block to ensure it's accessible throughout the loop.
- Fixed the `checkForAlerts` usage by calling it correctly.
- Clear `sentences` after each container iteration to ensure it doesn't accumulate past data.

Let me know if there's anything else you'd like to adjust!



Ab hier sind Nachrichten nur für dich sichtbar