Displaying Nested Event Data from JSON

In this tutorial, you will work with nested JSON data by displaying the security events associated with each device.

The JSON file contains devices, and each device contains its own list of events.

Understanding the JSON Structure

The JSON file contains:

```
Home
 └ Devices
     └ Events
```

Example:

```
{
    "deviceId": "CAM-01",
    "events": [
        {
            "eventType": "motion_detected"
        }
    ]
}
```

To display the events, we need to loop through:

1. The devices
2. The events within each device

Create an Events Table

Open:

```
script.js
```

Replace the previous code with:

```
fetch("smart_security_data.json")
    .then(response => response.json())
    .then(data => {

        let html = `
            <h2>Security Events</h2>

            <table border="1">

                <tr>
                    <th>Device ID</th>
                    <th>Event Type</th>
                    <th>Severity</th>
                </tr>
        `;

        data.devices.forEach(device => {

            device.events.forEach(event => {

                html += `
                    <tr>
                        <td>${device.deviceId}</td>
                        <td>${event.eventType}</td>
                        <td>${event.severity}</td>
                    </tr>
                `;

            });

        });
```

```
        html += "</table>";

        document.getElementById("output").innerHTML = html;

    });
```

View the Results

Open:

```
http://localhost/json-demo/
```

You should see a table similar to:

Device ID	Event Type	Severity
CAM-01	motion_detected	high
CAM-01	remote_access_attempt	critical
LOCK-02	unlock_attempt	medium

Add Additional Event Information

The JSON file contains more information about each event.

Each event includes:

```
timestamp
eventType
dataTransmitted
severity
accessResult
```

Update the table headings:

```
<tr>
    <th>Device ID</th>
```

```
    <th>Timestamp</th>
    <th>Event Type</th>
    <th>Severity</th>
    <th>Access Result</th>
</tr>
```

Update the Table Rows

Replace the existing row code with:

```
html += `
    <tr>
        <td>${device.deviceId}</td>
        <td>${event.timestamp}</td>
        <td>${event.eventType}</td>
        <td>${event.severity}</td>
        <td>${event.accessResult}</td>
    </tr>
`;
```

Refresh the page.

The table should now display more detailed event information.

Smart Home Dashboard

Security Events

Device ID	Timestamp	Event Type	Severity	Access Result
CAM-01	2026-03-14T02:18:45	motion_detected	high	authorised
CAM-01	2026-03-14T02:19:10	remote_access_attempt	critical	blocked
LOCK-02	2026-03-13T23:47:02	unlock_attempt	medium	denied

Display the Device Type

Sometimes multiple devices may generate events.

Add another column heading:

```
<th>Device Type</th>
```

Update the row:

```
<td>${device.deviceType}</td>
```

Your table will now identify which type of device generated each event.

Complete JavaScript File

```
fetch("smart_security_data.json")
    .then(response => response.json())
    .then(data => {

        let html = `
            <h2>Security Events</h2>

            <table border="1">

                <tr>
                    <th>Device ID</th>
                    <th>Device Type</th>
                    <th>Timestamp</th>
                    <th>Event Type</th>
                    <th>Severity</th>
                    <th>Access Result</th>
                </tr>
        `;

        data.devices.forEach(device => {
```

```
            device.events.forEach(event => {

                html += `
                    <tr>
                        <td>${device.deviceId}</td>
                        <td>${device.deviceType}</td>
                        <td>${event.timestamp}</td>
                        <td>${event.eventType}</td>
                        <td>${event.severity}</td>
                        <td>${event.accessResult}</td>
                    </tr>
                `;

            });

        });

        html += "</table>";

        document.getElementById("output").innerHTML = html;

    });
```

What Happened?

devices

```
In this tutorial, you looped through:

```javascript id="rjlwmz"
data.devices
```

and then:

```
device.events
```

This is known as a nested loop and is commonly used when working with JSON files and APIs.
```

You have successfully displayed nested JSON data in a table.

Revision #2
Created 2026-06-10 02:00:19 UTC by Mr Napper
Updated 2026-08-19 00:00:00 UTC by Mr Napper