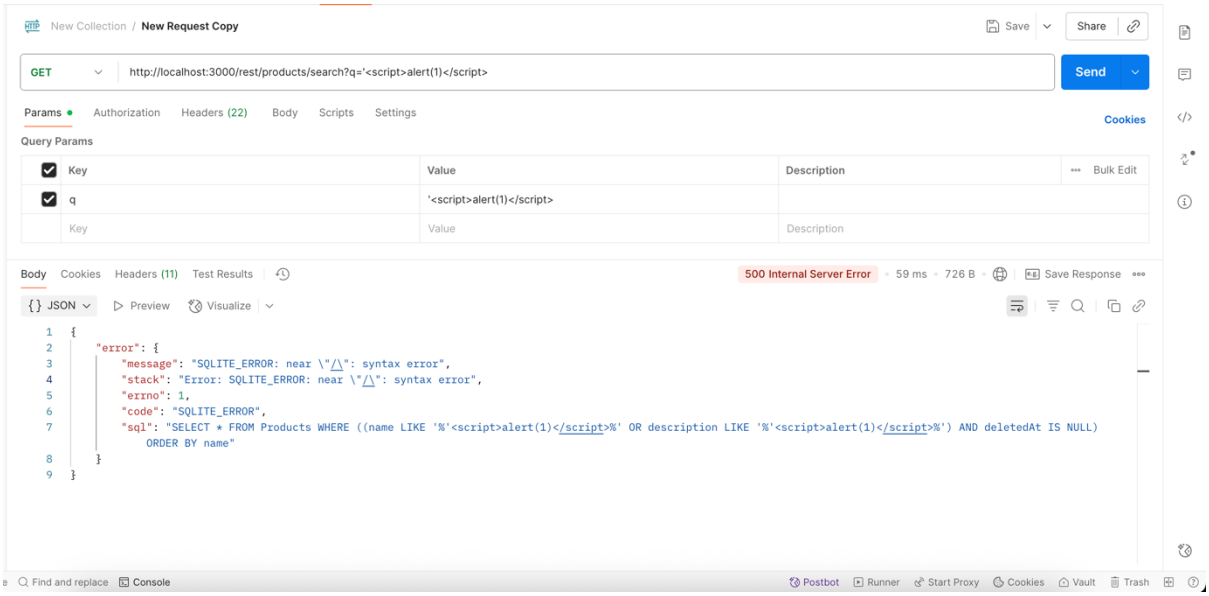
Task A - SQL Injection

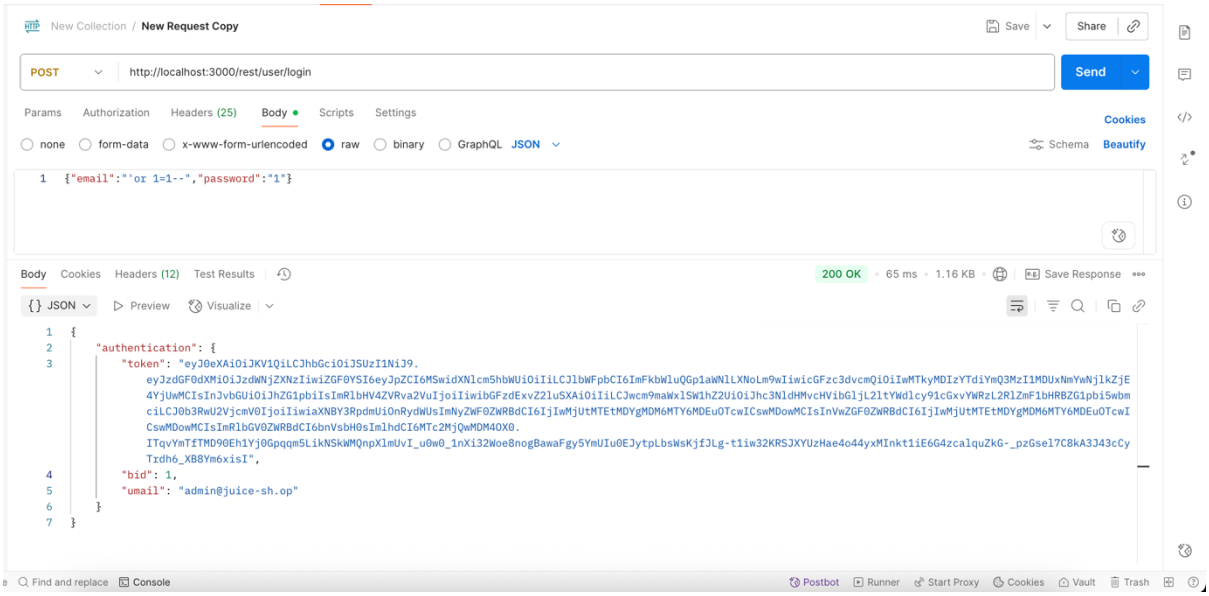


Kenapa ?

- query seharusnya terlihat seperti name LIKE '%somequery%'.
- Karena server menyusun query dengan menggabung (concatenate) string user langsung, input <script>... dimasukkan tanpa pembungkus yang benar.
- SQL parser melihat '%'<script> — yaitu: string literal '%'' diikuti token <script> yang bukan token SQL valid → SQLite mengeluh near "/": syntax error (karena simbol / dalam </script> dianggap bagian token yang tidak valid).
- Jadi payload memecah literal string SQL, dan menyebabkan syntax error.

Perbaikan

- Jangan menyusun SQL dengan string concatenation.
- Escape wildcard saat memakai LIKE dan pakai parameter binding untuk pola (%...%).
- Validasi input (panjang, karakter yang diizinkan)
- Jangan pamerkan stack trace / SQL ke client, log internal saja.



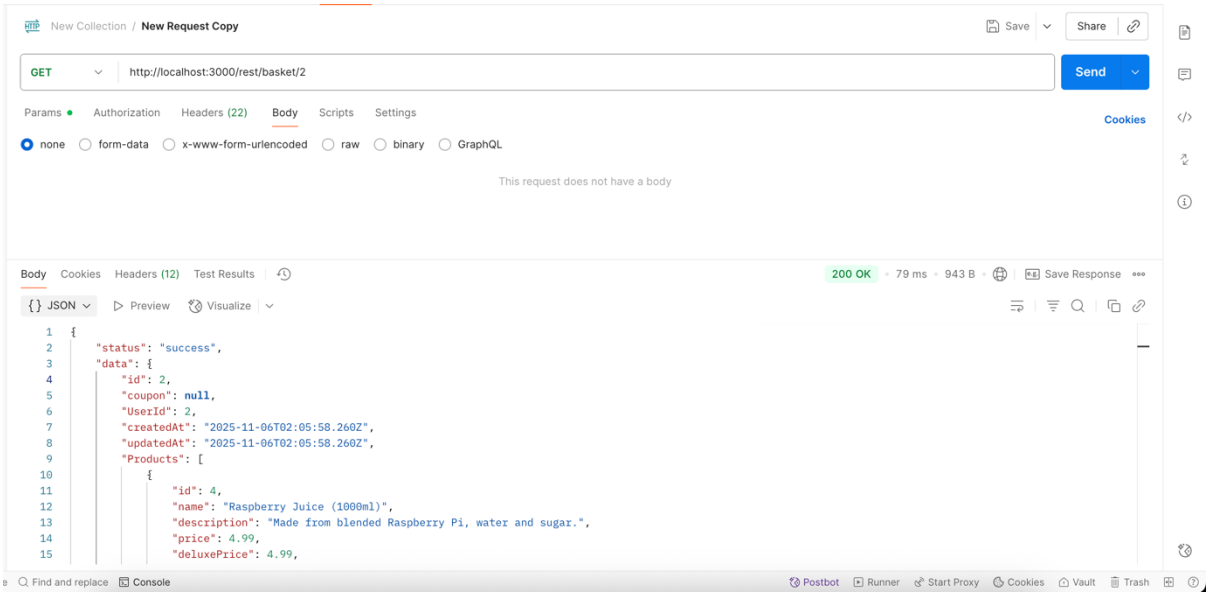
Kenapa ?

- Server menyisipkan input user langsung ke string query tanpa parameterization.
- Kutip tunggal ' dalam input memecah literal SQL sehingga bisa menulis kondisi sendiri.
- -- adalah komentar SQL yang mengabaikan sisa kondisi (misal pengecekan password).

Perbaikan

- Prepared statements / parameterized queries.
- Validasi input (contoh: valid email format, panjang maksimal).

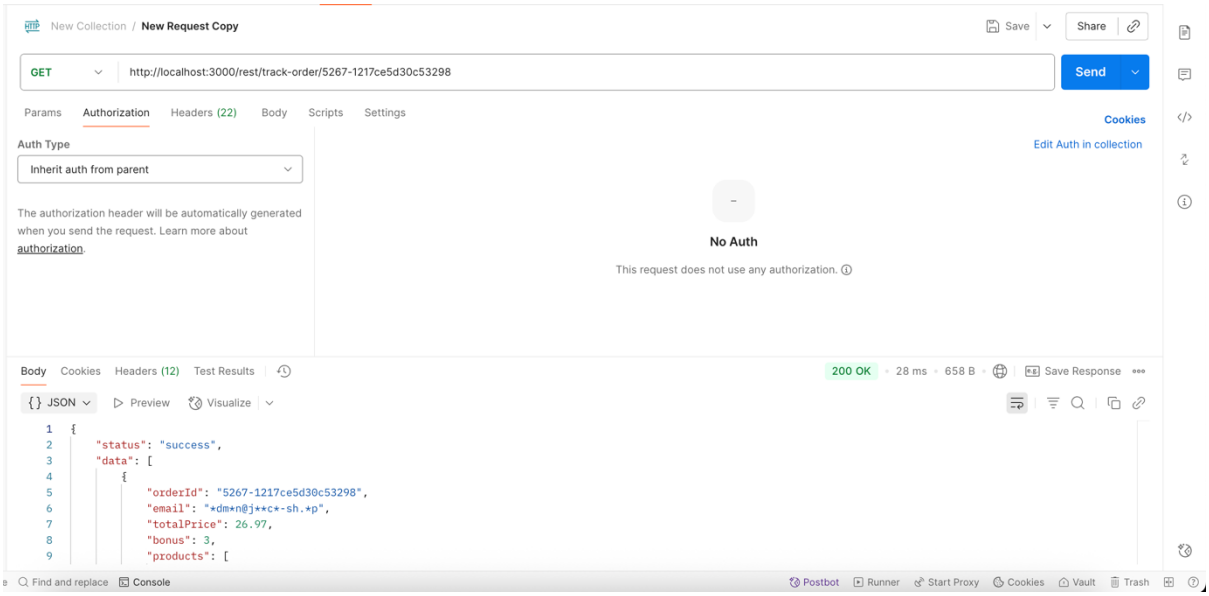
Task B – BAC (Broken Access Control)



Kenapa ? Server tidak memverifikasi kepemilikan objek sebelum mengembalikan data.
Token autentikasi hanya memastikan kamu adalah *user yang login*,
tetapi tidak memastikan bahwa kamu *berhak atas resource dengan ID tersebut*.

Perbaikan

- Tambahkan validasi kepemilikan (object ownership check)
- Idealnya, endpoint /rest/basket cukup mengembalikan basket milik user yang sedang login tanpa memerlukan parameter id



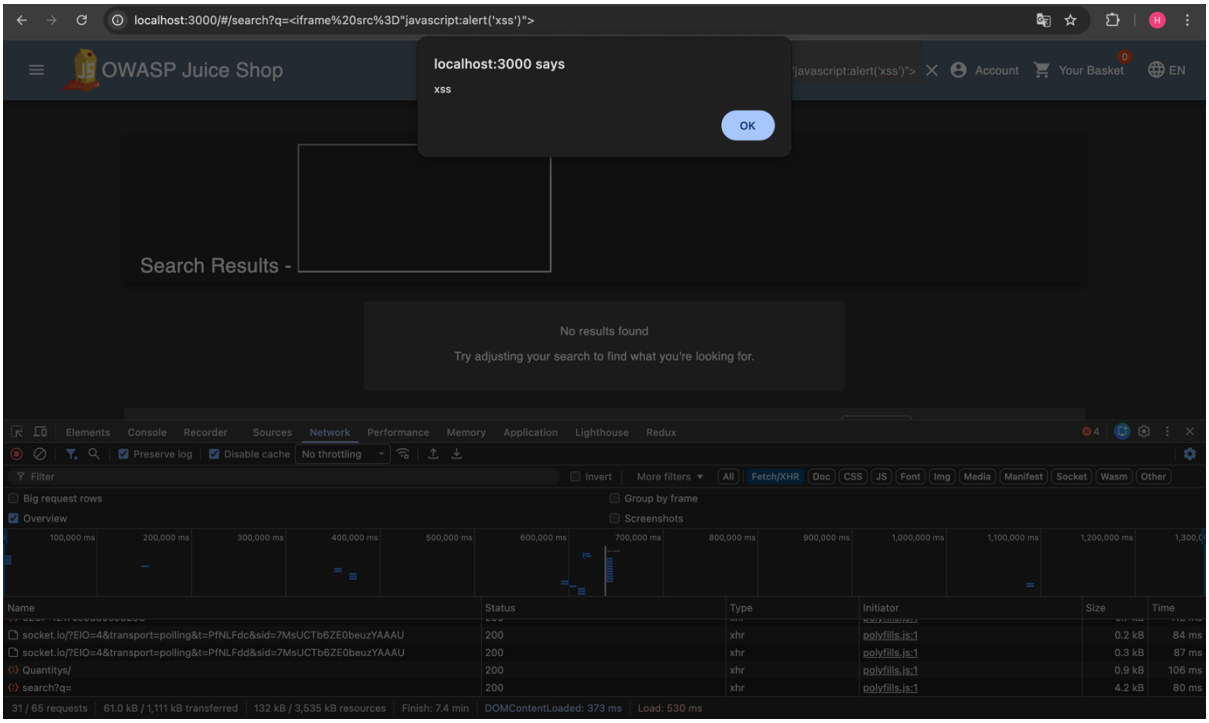
Kenapa ?

- Bisa mendapatkan data order berdasarkan order id milik orang lain berdasarkan order id

Perbaikan

- Tambahkan validasi kepemilikan (object ownership check)

Task C – XSS



Kenapa ?

- Tidak ada proses sanitasi input.
- Browser langsung menganggap teks <script>...</script> sebagai HTML sah yang bisa dieksekusi.

Perbaikan

- Validasi dan encode input
- Validasi nilai hash, query string, atau field input sebelum digunakan.
- Encode karakter berbahaya seperti <, >, &, ', dan " jika akan ditampilkan.