BYTE5 TECH TALK / 17. NOVEMBER 2026

Wem gehört Rechnung 2001?

Eine SaaS-Fallstudie mit KI

Marcel Wege · byte5

byte5

Welche Rechte gelten in dieser Anwendung?

Ein Workspace ist der Datenbereich eines Kunden. Wir nennen ihn auch Mandant.

Person bei Nordlicht	Erlaubte Arbeit
Alice · Member	Nordlicht-Rechnungen lesen
Nora · Admin	Nordlicht-Rechnungen lesen und exportieren

Beide dürfen keine Rechnungen von Südstern erhalten.

Alice arbeitet bei Nordlicht

Die drei Nordlicht-Rechnungen gehören zu Alices erlaubtem Arbeitsbereich.

Alice · Member · Nordlicht

Rechnung	Kunde / Leistung	Status	Betrag
RE-1001 ↗	**Fjord & Form** Markenplattform · Phase 1	Offen	4.800,00 €
RE-1002 ↗	**Küstencode** SaaS-Portal · November	Bezahlt	7.250,00 €
RE-1003 ↗	**Morgenrot Labs** UX-Audit · Q4	Offen	2.400,00 €

Die Liste zeigt den erwarteten Normalfall. Diese Nutzung muss auch nach dem Fix funktionieren.

Welche Rechnung darf *Alice* öffnen?

Rechnung	Rechnungsempfänger	Workspace
1001	Fjord & Form	Nordlicht
2001	Alpenblick AG	Südstern

Die Workspace-Zuordnung entscheidet über das Recht. Auch Nora bleibt als Admin an Nordlicht gebunden.

Alice ist Member bei Nordlicht. Und was dürfte Nora als Nordlicht-Admin?

Wo die Anwendung über Rechte entscheidet

Der Browser stellt Anfragen. Der Server prüft Rechte und lädt Daten aus einer gemeinsamen Datenbank.

Datei	Aufgabe im Fall
`auth.ts`	Identität beim Login feststellen
`profile.ts`	Profiländerungen übernehmen
`billing.ts`	Rechnungen lesen und exportieren

Eingefrorener Prüfstand · 12 Dateien · Snapshot f09cc9a1354f… · server.ts verbindet die Routen

Unser tatsächlicher Analyseauftrag

Der Auftrag gibt die fachliche Grenze vor: Eine Admin-Rolle gilt nur im eigenen Workspace.

```
Prüfe die lokale Rechnungs-SaaS auf Sicherheitsfehler.
Benutzer dürfen nur Daten ihres Mandanten sehen.
Eine Admin-Rolle gilt innerhalb dieses Mandanten.
Verfolge verdächtige Pfade über die beteiligten Dateien.
Nenne Voraussetzungen, Codebelege und eine lokale Gegenprobe.
Ändere keinen Code. Behaupte keine ausgeführten Tests.
```

Wörtlicher Anfang des vorbereiteten Auftrags · vollständiger Auftrag im Paket

Ein Kandidat aus dem Analysebericht

> **„Alice kann etwa Rechnung 2001 einschließlich Kunde, vertraulichem Projekttitel, Betrag und Status lesen.“**

Der Bericht leitet das aus dem Code ab. Mit einem lokalen Request prüfen wir, ob Alice die fremden Daten tatsächlich erhält.

Vorbereitete KI-Analyse 01 · wörtlicher Auszug

Die Detailabfrage vergisst den Workspace

Liste: Nummern aus Nordlicht

Der Server wählt nur Rechnungen aus Alices Workspace.

```
SELECT * FROM invoices
WHERE tenant_id = ?
ORDER BY id
```

Detail: nur nach Nummer

Der Server prüft nur die ID. Die Zugehörigkeit zu Nordlicht fehlt.

```
SELECT * FROM invoices
WHERE id = ?
```

Ein direkter Aufruf erreicht damit Daten außerhalb der korrekt gefilterten Liste.

tenant_id bezeichnet den Workspace · SQL-Auszüge aus server.ts:70 und billing.ts:4–5

Welche Antwort schützt die Mandantengrenze?

Alice ist bei Nordlicht angemeldet. Rechnung 2001 gehört Südstern.

```
GET /api/invoices/2001
```

Welche Daten darf die Antwort keinesfalls enthalten?

Der gültige Login weist Alice aus. Das Recht auf diese Rechnung muss der Server zusätzlich prüfen.

Rechnung 2001 wird ausgeliefert

HTTP-ANTWORT

200

Alice / Nordlicht
liest Südstern



HTTP 200 bedeutet: ausgeliefert. Alice sieht Projekttitel und Betrag einer Südstern-Rechnung.

Der erste bestätigte Befund

Erlaubte Identität

Alice ist regulär angemeldet und gehört zu Nordlicht.

Verletzte Regel

Der Server liefert ihr trotzdem eine Südstern-Rechnung.

Fehlende Prüfung

Gehört die angefragte Rechnung zu Alices Workspace?

S2 aus der Analyse ist für diesen Datenpfad lokal reproduziert

Welche Profilfelder darf *Alice* selbst ändern?

Feld	Bedeutung	Selbst ändern?
`name`	Anzeigename	Ja
`role`	Member oder Admin	Nein
`tenant_id`	Workspace-Zuordnung	Nein

Rolle und Workspace bestimmen Zugriffsrechte. Sie werden administrativ vergeben.

Fachliche Regel für das eigene Profil · ein fehlendes UI-Eingabefeld schützt die API nicht

Der Bericht verbindet Profil und Rechte

> **„Die Änderungen gelten deshalb sofort für das bestehende Sitzungstoken. […] Ein Member kann sich selbst zum Admin machen.“**

Der Server lädt die Rolle bei jeder Anfrage aus dem Konto. Weil Alice ihre Rolle selbst ändern kann, wirkt die unzulässige Änderung sofort beim nächsten Export.

Vorbereitete KI-Analyse 01 / S3 · gekürzter wörtlicher Auszug

Alice ändert ihre Rolle

Alice darf ihren Namen ändern. Die API übernimmt jedoch auch eine neue Berechtigungsrolle.

Alice sendet als Member

```
PATCH /api/profile
{ "role": "admin" }
```

Der Server liest danach

```
{
  "role": "admin",
  "tenant_id": 1
}
```

Der nächste Request verwendet die unzulässig gespeicherte Admin-Rolle.

Echte Requests 08 und 09 · GET /api/me bestätigt den gespeicherten Benutzerzustand

Die Adminprüfung begrenzt den Datenumfang nicht

Darf diese Person exportieren?

Die Rollenprüfung sperrt gewöhnliche Mitglieder.

```
if (user.role !== 'admin')
  throw new HttpError(403, ...);
```

Welche Daten werden exportiert?

Die Abfrage lädt alle Rechnungen. Eine Bedingung für Nordlicht fehlt.

```
SELECT * FROM invoices
ORDER BY id
```

Eine Nordlicht-Admin darf exportieren. Ihr Export muss trotzdem auf Nordlicht begrenzt sein.

billing.ts:8–10 · Rollenprüfung als gekürzter Codeauszug, Datenabfrage als SQL-Auszug

Im Export stehen fünf Rechnungen

Auch als Admin dürfte Alice nur die drei Nordlicht-Rechnungen exportieren.

5

Rechnungen
aus 2 Mandanten

Rechnung	Workspace
1001	Nordlicht
1002	Nordlicht
1003	Nordlicht
2001	Südstern
2002	Südstern

Die zusätzlichen IDs 2001 und 2002 gehören Südstern. Genau diese Kundengrenze wird verletzt.

Drei Requests ergeben einen Angriffspfad

01 `POST /api/login`

Alice meldet sich mit ihrem gültigen Passwort an.

02 `PATCH /api/profile  {"role":"admin"}`

Das Profil erlaubt ihr, selbst eine Admin-Rolle zu setzen.

03 `GET /api/export`

Der Export akzeptiert die Rolle und liefert auch Südstern-Rechnungen.

Zwei Ursachen wirken zusammen: selbst vergebene Rechte und fehlende Workspace-Grenze beim Export.

Nora braucht den Profiltrick nicht

Nora darf als Nordlicht-Admin exportieren. Ihre Rechte bleiben auf Nordlicht begrenzt.

ECHTE NORDLICHT-ADMIN

5

Rechnungen
aus beiden Mandanten

```
GET /api/export
HTTP 200

1001  Nordlicht
1002  Nordlicht
1003  Nordlicht
2001  Südstern
2002  Südstern
```

Nora erhält trotzdem alle fünf Rechnungen. Der Exportfehler besteht auch ohne Rollenmanipulation.

Was muss der erste Patch schützen?

Profil: Wer darf eine Rolle vergeben?

Detail: Gehört die Rechnung zum eigenen Workspace?

Export: Welche Rechnungen darf diese Admin sehen?

Welche Änderung verhindert welchen unerlaubten Zugriff?

Der Teilfix begrenzt das Profil auf den Namen

Ein Self-Service-Profil darf den Anzeigenamen ändern. Es darf keine neuen Zugriffsrechte vergeben.

Prüfung vor dem Update

```
Object.keys(changes)
  .some(key => key !== 'name')
```

Wirkung der Prüfung

Ein anderes Feld als name?
Der Server weist den Request mit 400 zurück.

Rolle und Workspace bleiben unverändert.

Der Teilfix schützt das Profil. Die Abfragen für Rechnungsdetail und Export bleiben unverändert.

case-partial/src/profile.ts:5–6 · Ausschnitt der Feldprüfung

Der Teilfix lässt Noras Export offen

Alice / Rollenänderung

400

Rolle bleibt member
Export bleibt gesperrt

Nora / Export

200

5 Rechnungen
inklusive Südstern

Der Profilfix funktioniert. Im Export fehlt weiterhin die Begrenzung auf Noras Workspace.

Die zweite Codeprüfung findet den offenen Export

> **„Nora als Administratorin von Mandant 1 erhält auch die Rechnungen 2001 und 2002 von Mandant 2.“**

Das Review bestätigt die Profilbeschränkung. In der Exportabfrage fehlt weiterhin die Bedingung für Noras Workspace.

Jede Datenauswahl braucht den eigenen Workspace

Einzelne Rechnung

Nummer und Workspace müssen gemeinsam passen.

```
SELECT * FROM invoices
WHERE id = ? AND tenant_id = ?
```

Export nach der Adminprüfung

Der Export lädt nur Rechnungen des eigenen Workspaces.

```
SELECT * FROM invoices
WHERE tenant_id = ?
ORDER BY id
```

Der Workspace stammt aus dem serverseitig geladenen Konto. Das eigene Profil darf ihn nicht ändern.

SQL-Auszüge aus case-fixed/src/billing.ts · Platzhalter binden Werte, die Workspace-Bedingung begrenzt Daten

Der Fix erhält die erlaubte Arbeit

Member lesen eigene Rechnungen. Admins dürfen sie zusätzlich exportieren. Beide Rollen bleiben im eigenen Workspace.

Person / Rolle	Eigene Rechnung	Fremde Rechnung	Export
Alice · Member	200 · erlaubt	404 · gesperrt	403 · gesperrt
Nora · Admin	200 · erlaubt	404 · gesperrt	3 · Nordlicht
Bob · Member	200 · erlaubt	404 · gesperrt	403 · gesperrt
Beate · Admin	200 · erlaubt	404 · gesperrt	2 · Südstern

Ausgeführte Gegenproben: 200 liefert Daten · 404 sperrt das fremde Objekt · 403 sperrt die Funktion

Der Arbeitsauftrag benennt Fehler und Gegenproben

Auswirkung

Nordlicht erhält vertrauliche Südstern-Rechnungen.

Korrektur

Profil auf den Namen begrenzen. Detail und Export an den Workspace binden.

Akzeptanz

Alice liest eigene Rechnungen. Nora exportiert Nordlicht. Beide erhalten keine Südstern-Rechnungen.

Vollständiger Befund, Codeänderungen und Retestreferenzen im Paket

Was die KI-Analysen konkret beigetragen haben

01 **Die Liste filtert nach Workspace, das Detail nicht.**

Analyse 01 / S2 · der lokale Alice-Request bestätigt den Fremdzugriff.

02 **Das Profil verändert die Rolle, die der Export prüft.**

Analyse 01 / S3 · die getrennte Requestfolge bestätigt diese Verbindung.

03 **Das Profil ist geschützt, der Export bleibt offen.**

Review 02 / S3 · die Nora-Gegenprobe bestätigt die verbleibende Ursache.

Die Berichte erklären Codepfade. Getrennte HTTP-Tests zeigen deren tatsächliches Verhalten.

Welcher Datenpfad ist bei euch vergleichbar?

Anna arbeitet in Workspace A. Dokument 7301 gehört Workspace B.

```
GET /api/files/7301
```

Reicht Annas gültiger Login für diesen Download?
Wie prüft ihr fremden und erlaubten eigenen Zugriff?

Transferfall für die Diskussion · Eigentümerschaft und Workspace-Zuordnung vor dem Test festlegen

SQL-Injection umgeht die Passwortprüfung

Beate soll nur mit ihrem richtigen Passwort eine Sitzung erhalten.

Warum das Passwort entfällt

Das ' beendet den E-Mail-Text im SQL-Befehl. -- kommentiert die folgende Passwortprüfung aus.

```
beate@suedstern.test' --
Passwort: x
```

Tatsächlich mit Beate geprüft

Login-Versuch	Original	Fix
Passwort falsch	401	401
SQLi + falsch	200	401
Passwort richtig	200	200

Fix: E-Mail als Parameter binden und danach das Passwort prüfen.

Separater HTTP-Nachweis · der Teilfix verhält sich wie das Original · 200 = angemeldet, 401 = abgewiesen

Eine Notiz führt Code in Noras Browser aus

Alice speichert eine Notiz. Beim Lesen darf ihr Inhalt keine Befehle in Noras Browser ausführen.

Original: Inhalt wird HTML

```
body.innerHTML = note.body;
```

Der Browser verarbeitet die Notiz als HTML. Der gespeicherte Ereigniscode läuft beim Öffnen.

Fix: Inhalt bleibt Text

```
body.textContent = note.body;
```

Der Browser zeigt die Notiz als Text. HTML-Zeichen bleiben sichtbarer Inhalt.

Beobachtet: Original und Teilfix führen den lokalen Marker aus. Im Fix bleibt der Marker aus und die Notiz lesbar.

Echter Browsernachweis: Alice speichert, Nora öffnet · gemessen wurde eine lokale DOM-Markierung

FALLSTUDIE / FRAGEN UND AUSTAUSCH

Vielen Dank
Welchen Datenpfad prüfen wir als Nächstes?

Wer darf auf welche Daten zugreifen?

Welche tatsächliche Antwort verletzt diese Regel?

Schützt der Fix fremde Daten und erhält die eigene Arbeit?

Zeigt mir auf LinkedIn,
was ihr herausgefunden habt.

VERNETZEN WIR UNS

Marcel Wege

byte5 GmbH

mwege@byte5.de

LinkedIn-Profil öffnen

QR scannen und vernetzen