

REPOSITORY-MISSIONSDOSSIER · ILLUSTRATIVE BEISPIELDATEN

BRANCHWARDEN

Härtung eines Kunden-Login-Flows

ILLUSTRATIVE BEISPIELDATEN · KEIN KUNDENFALL

MUSTERDOSSIER · VERSION 1.0

Härtung eines Kunden-Login-Flows

Repository-Mission für beispiel-org/customer-portal: Passwort-Reset gegen Kontoenumeration härten, Rate-Limit-Fehler verständlich anzeigen und bestehende Loginpfade regressionstesten.. Lieferung endet bei einem überprüfbaren Pull Request.

STATUS	SCORE	FREIGABEN
READY_FOR_REVIEW	98 / 100	4 offen

Strukturierte Eingaben

REPOSITORY	beispiel-org/customer-portal
OBJECTIVE	Passwort-Reset gegen Kontoenumeration härten, Rate-Limit-Fehler verständlich anzeigen und bestehende Loginpfade regressionstesten.
STACK	React 19; TypeScript; Supabase Auth; TanStack Router; Vitest.
ACCEPTANCE	Gleiche öffentliche Antwort für existente und unbekannte E-Mail; 12-Zeichen-Regel; Rate-Limit-Fall getestet; kein Secret im Diff; Build und Tests bestehen.
COMMANDS	npm test; npm run lint; npm run build; npm audit --omit=dev
SCHUTZBEREICHE	Produktionsdaten; Billing; main branch; Supabase Service Role; DNS und SMTP.

Kennzahlen

MISSIONSREIFE	ABNAHMEKRITERIEN	SCHUTZBEREICHE
98 / 100	5	5

Ausführungsplan

- Repository-Berechtigung, Zielbranch und aktuellen Hauptstand verifizieren.
 - Isolierten Arbeitsbranch ohne direkte Änderungen am Hauptbranch erstellen.
 - Stack und Abhängigkeiten prüfen: React 19; TypeScript; Supabase Auth; TanStack Router; Vitest..
 - Auftrag in kleinste überprüfbare Änderung zerlegen: Passwort-Reset gegen Kontoenumeration härten, Rate-Limit-Fehler verständlich anzeigen und bestehende Loginpfade regressionstesten..
 - Vorhandene Tests zuerst ausführen und Ausgangszustand protokollieren.
 - Änderung implementieren, Geheimnisse und personenbezogene Daten aus Logs fernhalten.
 - Tests, Lint, Build und Sicherheitsprüfung ausführen.
 - Diff gegen Auftrag, Schutzbereiche und unbeabsichtigte Nebenwirkungen prüfen.
 - Commit mit nachvollziehbarer Beschreibung erzeugen und Branch pushen.
 - Pull Request mit Nachweisen, Risiken und Rollback-Hinweis zur menschlichen Freigabe öffnen.
-

Abnahmematrix

- A1: Gleiche öffentliche Antwort für existente und unbekannte E-Mail
 - A2: 12-Zeichen-Regel
 - A3: Rate-Limit-Fall getestet
 - A4: kein Secret im Diff
 - A5: Build und Tests bestehen.
-

Prüfkommandos

- Nur nach Autorisierung ausführen: npm test
 - Nur nach Autorisierung ausführen: npm run lint
 - Nur nach Autorisierung ausführen: npm run build
-

→ Nur nach Autorisierung ausführen: `npm audit --omit=dev`

Schutz- und Abbruchregeln

→ Nicht ohne separate Freigabe ändern: Produktionsdaten

→ Nicht ohne separate Freigabe ändern: Billing

→ Nicht ohne separate Freigabe ändern: main branch

→ Nicht ohne separate Freigabe ändern: Supabase Service Role

→ Nicht ohne separate Freigabe ändern: DNS und SMTP.

→ Keine direkten Pushes auf geschützte Hauptbranches.

→ Bei Secret-Fund, Datenverlustgefahr oder unerwartetem Produktionszugriff sofort stoppen.

Freigabegates

OFFEN

Repository und Auftrag sind ausdrücklich autorisiert

OFFEN

Abnahmekriterien und Prüfkommandos sind definiert

OFFEN

Keine Geheimnisse oder nicht freigegebenen Produktionsänderungen im Diff

OFFEN

Menschliche Prüfung und Merge-Freigabe erforderlich

WICHTIGER HINWEIS

BRANCHWARDEN plant kontrollierte Repository-Arbeit bis zum Pull Request. Es führt keinen ungeprüften Merge oder Produktionseinsatz aus; API-, GitHub- und Drittdienstkosten sind nicht enthalten.

Dieses Musterdossier wurde aus frei erfundenen Referenzdaten mit der jeweiligen AXIOMDOCK-Engine erzeugt. Es dokumentiert Struktur und Ausgeart, nicht die Eignung für einen realen Einzelfall.