

Integrare externă rezistentă la duplicate

Fișă publică de proiect · exerciții și criterii de acceptare

Proiectezi o integrare demonstrativă de plăți sau notificări. Lucrezi cu contracte, sandbox și mock-uri, fără tranzacții financiare reale.

Livrabile

- Contract
- Autentificare
- Retry
- Idempotență
- Webhook
- Teste

Criterii de acceptare

- Semnătura este verificată
- Duplicatul are un singur efect
- Timeout-ul nu este tratat ca plată eșuată definitiv

Contract de integrare înaintea primului apel

O integrare are nevoie de autentificare, scheme de cerere și răspuns, limite, erori și reguli de reluare. Citește documentația endpointului concret și folosește sandboxul când există. Nu conecta o operație reală cu efect financiar doar pentru a verifica formatul. Păstrează identificatorul intern și cel al furnizorului pentru reconciliere. O integrare este completă când poate trata și eșecurile, nu doar un răspuns de succes ales manual.

Exemplu de lucru

Fișă: scop; mediu sandbox; metodă și URL; autentificare; câmpuri obligatorii; statusuri; timeout; idempotență; webhook; log minim; responsabil.

Test minim: succes, validare, acces invalid, limită, timeout, răspuns neașteptat.

Date: numai fictive în laborator.

Interpretarea rezultatului

Un 200 poate conține un status de business care cere altă acțiune, potrivit contractului furnizorului. Nu deduce succesul dintr-un singur câmp fără să verifici documentația. Păstrează adapterul furnizorului separat de regula principală a aplicației.

Exercițiu de consolidare

Completează fișa pentru o integrare de generare AWB fără a executa o expediție reală.

1. Anatomia unei integrări

Completează fișa pentru o integrare de generare AWB fără a executa o expediție reală.

Verificare orientativă

Un 200 poate conține un status de business care cere altă acțiune, potrivit contractului furnizorului. Nu deduce succesul dintr-un singur câmp fără să verifici documentația. Păstrează adapterul furnizorului separat de regula principală a aplicației.

De evitat

Redirectul utilizatorului nu este dovadă suficientă a plății.

API key, OAuth și JWT nu sunt sinonime

O cheie API identifică sau autorizează un client conform furnizorului; OAuth este un cadru de delegare, iar JWT este un format de token care poate fi folosit în diferite protocoale. Nu trata decodarea JWT ca verificare a semnăturii. Validează emitentul, audiența, expirarea și algoritmi conform bibliotecii și protocolului. Pentru OAuth, păstrează state și celelalte protecții cerute de fluxul ales. Nu implementa manual criptografia.

Exemplu de lucru

Întrebări de alegere: cine deleagă accesul; ce operații; ce durată; cum se revocă; unde se păstrează secretul?

Cheie server: nu ajunge în browser.

Token: nu se pune în URL sau log.

JWT: semnătură și claims verificate, nu doar payload citit.

OAuth: flux documentat de furnizor.

Interpretarea rezultatului

Un token valid criptografic poate fi destinat altei aplicații sau altui scop. Verificarea trebuie să includă contextul, nu numai semnătura. Pentru un client public, regulile de păstrare a secretelor diferă față de un server; nu copia același flux peste tot.

Exercițiu de consolidare

Describe un caz în care JWT-ul este semnat corect, dar trebuie totuși respins pentru audiență greșită.

2. Autentificare: key, OAuth, JWT

Describe un caz în care JWT-ul este semnat corect, dar trebuie totuși respins pentru audiență greșită.

Verificare orientativă

Un token valid criptografic poate fi destinat altei aplicații sau altui scop. Verificarea trebuie să includă contextul, nu numai semnătura. Pentru un client public, regulile de păstrare a secretelor diferă față de un server; nu copia același flux peste tot.

De evitat

JWT este un format de token, nu singur un flux complet de autentificare.

Erori HTTP legate de acțiuni concrete

Clasifică erorile înainte de retry. Unele cer corectarea cererii, altele reautentificare sau așteptare. Păstrează răspunsul tehnic minim necesar și un mesaj public fără secrete. Un furnizor poate folosi statusuri și coduri proprii; tabelul este o orientare, nu un înlocuitor pentru contract. Nu repeta la nesfârșit o cerere 400 și nu dezactiva verificarea TLS pentru o eroare de certificat.

Exemplu de lucru

400/422: verifică datele și schema.
401: verifică autentificarea.
403: verifică drepturile, nu roți chei la întâmplare.
429: respectă politica de limită și eventual Retry-After.
5xx: eroare furnizor, reluare controlată dacă operația permite.
Timeout: status necunoscut al efectului, verifică reconcilierea.

Interpretarea rezultatului

Pentru o operație de plată sau expediere, timeoutul poate apărea după acceptare. Nu considera automat că operația nu există. Păstrează cheia logică și identificatorul furnizorului pentru a verifica rezultatul înaintea repetării.

Exercițiu de consolidare

Construiește o decizie de retry pentru citirea unui catalog și una pentru crearea unei expedieri, explicând diferența.

3. Erori și rate limits

Construiește o decizie de retry pentru citirea unui catalog și una pentru crearea unei expedieri, explicând diferența.

Verificare orientativă

Pentru o operație de plată sau expediere, timeoutul poate apărea după acceptare. Nu considera automat că operația nu există. Păstrează cheia logică și identificatorul furnizorului pentru a verifica rezultatul înaintea repetării.

De evitat

Retry pe orice eroare poate agrava incidentul.

Idempotență pe operația logică

Cheia de idempotență trebuie să reprezinte aceeași operație logică, nu să fie generată nou la fiecare retry. Dacă schimbi payloadul sub aceeași cheie, contractul furnizorului poate refuza sau poate avea alt comportament. Păstrează cererea, statusul și identificatorul extern într-o evidență controlată. O constrângere unică ajută concurența, dar nu coordonează singură toate efectele externe. Testează răspunsul pierdut după succes.

Exemplu de lucru

Operație internă: creare expediere pentru comanda 42, versiunea 1.
Cheie: shipment-42-v1, conform regulilor furnizorului.
Primul apel: acceptat, id extern S-100.
Retry cu aceeași operație: nu creează S-101.
Schimbare adresă: operație distinctă sau flux de modificare, nu retry mascat.

Interpretarea rezultatului

Idempotența nu înseamnă că orice metodă sau endpoint este sigur de repetat. Verifică durata de păstrare a cheii și politica furnizorului. Pentru un job reluat după mult timp, poate fi necesară reconcilierea, nu o nouă cerere identică.

Exercițiu de consolidare

Simulează două procese care încearcă aceeași operație și verifică faptul că există o singură expediție logică.

4. Retry și idempotență

Simulează două procese care încearcă aceeași operație și verifică faptul că există o singură expediție logică.

Verificare orientativă

Idempotența nu înseamnă că orice metodă sau endpoint este sigur de repetat. Verifică durata de păstrare a cheii și politica furnizorului. Pentru un job reluat după mult timp, poate fi necesară reconcilierea, nu o nouă cerere identică.

De evitat

O cheie nouă la fiecare retry poate dubla operațiunea.

Webhook verificat pe corpul corect și deduplicat

Semnătura unui webhook se verifică după algoritmul furnizorului, adesea pe corpul brut al cererii. Nu transforma JSON-ul și apoi presupune că semnătura se verifică identic. Verifică timestampul sau protecția anti-replay dacă protocolul le cere. Evenimentul poate fi duplicat sau poate ajunge în altă ordine. Păstrează identificatorul și statusul procesării, iar acțiunea trebuie să respecte starea curentă a resursei.

Exemplu de lucru

Flux: corp brut → verificare semnătură → validare schemă → deduplicare id eveniment → procesare/coadă → confirmare.
Teste: semnătură invalidă; eveniment repetat; eveniment întârziat; tip necunoscut; procesare eșuată.
Nu folosi: IP sau un câmp secret în JSON ca unic substitut pentru protocolul cerut.

Interpretarea rezultatului

Un eveniment „paid” nu trebuie să acorde acces unui utilizator arbitrar din URL; asociază resursa, suma și contextul conform contractului. Nu lăsa evenimentul mai vechi să inverseze o stare finală fără regulă. Păstrează loguri fără cheia de semnare.

Exercițiu de consolidare

Definește ce faci cu un eveniment valid, dar necunoscut, fără a bloca toate livrările ulterioare.

5. Webhook-uri

Definește ce faci cu un eveniment valid, dar necunoscut, fără a bloca toate livrările ulterioare.

Verificare orientativă

Un eveniment „paid” nu trebuie să acorde acces unui utilizator arbitrar din URL; asociază resursa, suma și contextul conform contractului. Nu lăsa evenimentul mai vechi să inverseze o stare finală fără regulă. Păstrează loguri fără cheia de semnare.

De evitat

Datele trimise de browser nu trebuie acceptate ca webhook de încredere.

Furnizor fals pentru cazuri greu de reprodus

Testarea unei integrări nu trebuie să consume operații reale sau bani pentru fiecare rulare. Izolează clientul furnizorului și folosește răspunsuri false pentru succes și erori. Păstrează câteva teste de contract în sandbox, separate de suita rapidă. Un fake care întoarce mereu succes nu verifică tratarea eșecurilor. Include răspuns valid JSON cu formă neașteptată, fiindcă parserul poate reuși și totuși datele să fie inutilizabile.

Exemplu de lucru

Matrice fake: 200 valid; 200 câmp lipsă; 422 validare; 429 limită; 500; timeout după acceptare; webhook duplicat.
Verificări: număr apeluri, payload permis, cheie idempotentă stabilă, status intern, mesaj public.
Sandbox: numai date și operații autorizate pentru test.

Interpretarea rezultatului

Testele trebuie să demonstreze și lipsa efectelor duplicate. Dacă adapterul furnizorului schimbă schema, un test de contract ar trebui să semnaleze diferența. Nu pune tokenuri reale în fixtures sau capturi salvate.

Exercițiu de consolidare

Scrie un scenariu de test pentru răspuns 200 fără identificatorul operației și definește statusul intern corect.

6. Testarea integrărilor

Scrie un scenariu de test pentru răspuns 200 fără identificatorul operației și definește statusul intern corect.

Verificare orientativă

Testele trebuie să demonstreze și lipsa efectelor duplicate. Dacă adapterul furnizorului schimbă schema, un test de contract ar trebui să semnaleze diferența. Nu pune tokenuri reale în fixtures sau capturi salvate.

De evitat

Un singur răspuns de succes nu acoperă ciclul real al integrării.

Atelier final: Webhook duplicat și timeout

Timp de lucru estimat: 40 de minute. Folosește datele de laborator de mai jos și salvează rezultatul separat de orice proiect real.

Date de pornire

Furnizor fictiv livrează evt-7 de două ori pentru aceeași operație. Înainte de webhook, cererea inițială a expirat și starea locală este necunoscută.

1. Pregătire — 5 minute

Extrage intrările și condițiile din enunț. Notează ce rezultat trebuie produs și ce informații nu sunt confirmate. Pregătește fișierele sau tabelul în care vei păstra dovada lucrului.

2. Construire — 15 minute

Definește stările pending, confirmed și failed, verificarea semnăturii și identificatorul unic. Simulează ambele livrări și reconcilierea stării.

3. Variație și verificare — 15 minute

Livrează evenimentele în ordine inversă și explică regula pentru actualizări vechi. Păstrează corpul fictiv și rezultatul fiecărei verificări.

4. Autoevaluare — 5 minute

Compară rezultatul cu reperele de mai jos. Înregistrează cel puțin o eroare identificată sau un caz-limită verificat, plus corectura ori explicația comportamentului observat.

Repere de rezolvare

Un singur efect de business este creat. Timeout-ul inițial nu dovedește eșecul definitiv. Semnătura invalidă nu modifică date, iar secretul nu apare în jurnal.

Ce predai

- Contract
- Autentificare
- Retry
- Idempotență
- Webhook
- Teste

Criterii de acceptare

- Semnătura este verificată
- Duplicatul are un singur efect
- Timeout-ul nu este tratat ca plată eșuată definitiv

Păstrează o versiune inițială și una revizuită, cu explicația diferențelor. Dacă un criteriu nu este îndeplinit, revino la lecția relevantă și repetă verificarea. Durata este orientativă; instalarea mediului și aprofundarea pot necesita timp suplimentar.

Registrul de lucru

Pentru fiecare etapă notează intrarea, rezultatul, dovada verificării, eroarea găsită și corectura. Nu introduce date personale sau secrete în fișa partajată.

Documentație și surse

- HTTP overview — MDN
- OWASP Top 10 — OWASP

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>

<https://owasp.org/projects/top-ten>