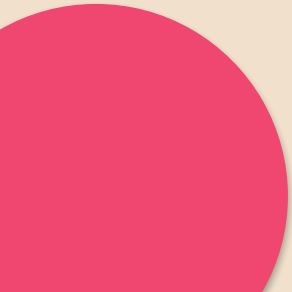




Les 17

Externe APIs in diepte

OAuth, webhooks, Stripe en Resend — productie-patronen



Terugblik

Productie-niveau externe APIs

Lessen 11-16: AI SDK, tool calling, agents, RAG, deploy, MCP.

Externe APIs in Les 15 (PokéAPI) was simpel. Productie is anders:

- Login met Google/GitHub (OAuth)
- Iemand doet betaling — server moet dat weten (webhooks)
- Betaalde APIs zoals Stripe en Resend
- Wat als de API faalt? Retry-logic
- Hoe doe je dat veilig en netjes?

Vandaag: de echte wereld. Stripe + Resend + webhooks + retry.

Planning

Vandaag — 180 minuten

Terugblik + API-types	15 min
Theorie: OAuth basics	20 min
Theorie: Webhooks + verifying	20 min
LIVE DEMO 1 — Stripe Checkout	25 min
LIVE DEMO 2 — Webhook ontvangen + verify	25 min
Pauze	15 min
LIVE DEMO 3 — Resend transactional email	25 min
LIVE DEMO 4 — Rate limit + retry	15 min
Productie checklist	10 min
Lesopdracht + Huiswerk	10 min

Drie soorten externe API integratie

Simple fetch

PokéAPI, Open-Meteo

URL + JSON (Les 15)

Key in header

OpenAI, Tavily, Stripe

Env vars, server-side

OAuth flow

Google, GitHub login

Redirect dance, tokens

Webhook

Stripe, GitHub events

POST endpoint + verify

Vandaag focus op de laatste twee — daar komen 80% van productie-API-bugs vandaan.

Theorie

OAuth — login met Google/GitHub

1. User klikt "Login met GitHub"
2. Redirect naar `github.com/login/oauth/authorize`
3. User logt in op GitHub + geeft toestemming
4. GitHub redirect terug naar JOUW app met code
5. JOUW server exchanged code voor `access_token`
6. JOUW server haalt user-data op met token
7. JOUW server zet sessie / cookie

In Next.js — drie populaire libraries:

Auth.js (NextAuth v5)

meest gebruikt

Better-Auth

moderne challenger

Clerk

paid, alles inclusief

Theorie

Webhooks — externe service belt JOUW app

Voorbeeld Stripe:

- 1 User doet betaling op Stripe Checkout
- 2 Stripe verwerkt betaling
- 3 Stripe stuurt POST request naar JOUW webhook URL
- 4 Jouw server update database, stuurt email, etc.

Cruciaal: signature verification

Zonder verify kan iedereen fake events sturen naar jouw URL.

```
const sig = request.headers.get("stripe-signature");
const event = stripe.webhooks.constructEvent(
  body, sig, process.env.STRIPE_WEBHOOK_SECRET
);
// Throws als signature niet klopt
```

Vandaag bouwen we

Subscription-dashboard

Mini-app: Premium subscription met Stripe + email confirmaties.

Tech:

- Next.js 16
- Stripe (test mode — geen echt geld)
- Resend (transactional email — gratis tier 3000/maand)
- Stripe CLI voor lokaal webhook testen

Flow:

```
[Pricing page] -> [Stripe Checkout] -> [Webhook] -> [Database + Email]
```

Endpoints:

POST /api/checkout

start Stripe sessie

POST /api/webhook/stripe

ontvang Stripe events

POST /api/email

verzend Resend mail (intern)

LIVE DEMO 1

Stripe Checkout — ~25 min

- 1 Stripe account in test mode + product + price
- 2 Pricing page met 'Subscribe' button
- 3 `app/api/checkout/route.ts` — Checkout session create
- 4 `success_url` + `cancel_url` + `customer_email`
- 5 Client redirect naar `session.url`
- 6 Test met card 4242 4242 4242 4242
- 7 Stripe Dashboard: transactie zichtbaar (test mode)

LIVE DEMO 2

Webhook ontvangen + verify — ~25 min

```
// app/api/webhook/stripe/route.ts
export async function POST(req: Request) {
  const body = await req.text(); // RAW body, niet json!
  const sig = (await headers()).get("stripe-signature")!;

  let event;
  try {
    event = stripe.webhooks.constructEvent(
      body, sig, process.env.STRIPE_WEBHOOK_SECRET!
    );
  } catch (err) {
    return new Response("Bad signature", { status: 400 });
  }

  if (event.type === "checkout.session.completed") {
    // handle...
  }
  return Response.json({ received: true });
}
```

Lokaal testen:

```
stripe listen --forward-to localhost:3000/api/webhook/stripe
```



Pauze

15 minuten

LIVE DEMO 3

Resend transactional email — ~25 min

```
// emails/WelcomeEmail.tsx
export function WelcomeEmail({ name }: { name: string }) {
  return (
    <div style={{ fontFamily: "sans-serif" }}>
      <h1>Welkom {name}!</h1>
      <p>Bedankt voor je aanmelding bij Premium.</p>
    </div>
  );
}
```

```
// In webhook handler
await resend.emails.send({
  from: "onboarding@resend.dev",
  to: session.customer_email!,
  subject: "Welkom bij Premium!",
  react: WelcomeEmail({ name: session.customer_email!.split("@")[0] }),
});
```

LIVE DEMO 4

Rate limit + retry — ~15 min

Pattern 1 — Exponential backoff retry:

```
async function retry<T>(fn: () => Promise<T>, attempts = 3): Promise<T> {
  for (let i = 0; i < attempts; i++) {
    try { return await fn(); }
    catch (err) {
      if (i === attempts - 1) throw err;
      await new Promise(r => setTimeout(r, 2 ** i * 1000));
    }
  }
  throw new Error("unreachable");
}
```

Pattern 2 — Rate limit met Upstash:

```
const ratelimit = new Ratelimit({
  redis: Redis.fromEnv(),
  limiter: Ratelimit.slidingWindow(10, "10s"),
});
const { success } = await ratelimit.limit(userId);
if (!success) return new Response("Too many", { status: 429 });
```

Productie checklist

API keys

- ✓ Test-keys vs prod-keys
- ✓ Vercel env scoping per env
- ✓ Rotate eens per kwartaal

Errors

- ✓ Exponential backoff retry
- ✓ Rate limit eigen endpoints
- ✓ Monitor met Sentry/Vercel

Webhooks

- ✓ Signature verification ALTIJD
- ✓ Idempotency check
- ✓ Return 200 snel — verwerking async

Security

- ✓ Keys nooit in client
- ✓ Webhook = verified signatures only
- ✓ User-input via Zod

Lesopdracht + Huiswerk

Lesopdracht (30 min)

- Stripe test account + product
- Pricing page + Checkout
- Test-betaling met 4242 card
- Success URL werkt

Huiswerk (~2 uur)

- A: Webhook + signature verify
- B: Resend transactional email
- C: Retry logic
- D: APIS.md met eindopdracht-plan
- Bonus: OAuth, idempotency, rate limit

Beoordeling (10 pt — voldoende 6+):

- | | |
|------------------------------|------|
| • A — Webhook + verify werkt | 3 pt |
| • B — Resend email werkt | 2 pt |
| • C — Retry helper + bewijs | 2 pt |
| • D — APIS.md compleet | 3 pt |

Afsluiting

Vragen?

Vandaag gezien:

- ✓ Drie soorten externe API integratie
- ✓ OAuth flow basics
- ✓ Webhooks ontvangen + signature verify
- ✓ Stripe Checkout productie-patroon
- ✓ Resend transactional email
- ✓ Rate limit + retry
- ✓ Productie checklist

Volgende les (Les 18 — DE LAATSTE): Supabase Auth + RLS

- Multi-user apps met login
- Magic link / password / social
- RLS policies — per-user data isolation
- Protected routes + middleware

Vragen?