

Les 17 — Lesopdracht

Langfuse setup + eerste traces

Vak: AI-Assisted Development

Duur: 30 min in-class

Vak: AI-Assisted Development **Duur:** 30 min in-class

Doel

Aan het einde: Langfuse hangt aan je Polderfest-chat (of een willekeurige AI-app). Je ziet elke chat-call in het Langfuse-dashboard met prompt, response, tokens, cost, latency.

Stap 1 — Langfuse account

1. Ga naar <https://cloud.langfuse.com> — sign up (mag met Google/GitHub)
2. New project — naam: polderfest-chat
3. Settings → API Keys → kopieer **Public Key** + **Secret Key**

Stap 2 — Install

In je polderfest-chat repo (uit Les 12):

```
pnpm add langfuse-vercel langfuse
```

`.env.local`:

```
LANGFUSE_PUBLIC_KEY=pk-1f-...  
LANGFUSE_SECRET_KEY=sk-1f-...  
LANGFUSE_BASE_URL=https://cloud.langfuse.com
```

Stap 3 — Wrap je chat-route

app/api/chat/route.ts — voeg experimental_telemetry toe aan je streamText of generateText:

```
import { streamText, convertToModelMessages } from "ai";
import { openai } from "@ai-sdk/openai";

export async function POST(req: Request) {
  const { messages } = await req.json();

  const result = streamText({
    model: openai("gpt-4o-mini"),
    messages: convertToModelMessages(messages),
    tools: { /* je bestaande tools */ },
    stopWhen: stepCountIs(5),
    experimental_telemetry: {
      isEnabled: true,
      functionId: "polderfest-chat",
      metadata: {
        userId: "anonymous",
      },
    },
  });

  return result.toUIMessageStreamResponse();
}
```

Stap 4 — Telemetry exporter

instrumentation.ts in project-root:

```
import { registerOTel } from "@vercel/otel";
import { LangfuseExporter } from "langfuse-vercel";

export function register() {
  registerOTel({
    serviceName: "polderfest-chat",
    traceExporter: new LangfuseExporter(),
  });
}
```

pnpm add @vercel/otel @opentelemetry/api

next.config.ts — enable instrumentation:

```
export default {
  experimental: {
    instrumentationHook: true,
  },
};
```

Stap 5 — Test

```
pnpm dev
```

Open <http://localhost:3000/chat> — stel 3 vragen.

Open <https://cloud.langfuse.com> → jouw project → **Traces** tab.

Je ziet:

- 3 traces (één per vraag)
- Per trace: prompt, response, tool-calls, model, tokens, cost
- Latency in ms

Klik op een trace → zie de volledige conversation tree.

Eisen

- ☐ Langfuse account + project aangemaakt
- ☐ API keys in `.env.local`
- ☐ `experimental_telemetry` actief in chat-route
- ☐ OpenTelemetry exporter geconfigureerd
- ☐ Min 3 traces zichtbaar in dashboard
- ☐ Per trace: tokens + cost + latency zichtbaar

Tijdsindeling (30 min)

Stap	Tijd
1-2 — Setup + install	8 min
3-4 — Code-changes	12 min
5 — Test + debug	10 min

Veelvoorkomende problemen

Symptoom	Oplossing
Geen traces in dashboard	Wacht 30s — flush is async. Of <code>await langfuse.flushAsync()</code> toevoegen
"instrumentationHook is required"	<code>next.config.ts</code> aanpassen + restart <code>pnpm dev</code>
Cost = \$0	Klopt vaak voor mini-calls — check tokens is wel goed

Symptoom	Oplossing
Tool-calls niet zichtbaar	Update naar laatste AI SDK + langfuse-vercel versies
401 Unauthorized	Public + secret key check, regio (US vs EU) klopt

Klaar?

Verder met huiswerk:

- Eval-suite (10 cases voor je app)
- Prompt-injection bescherming
- Per-user rate limiting + cost-strategie
- POLISH.md met screenshots + cijfers

Zie Les17-Huiswerk.pdf.