



Les 12




Tool Calling

Laat AI zelf kiezen welke functie aan te roepen



Terugblik

Het schaalprobleem dat we vandaag oplossen

-  Les 11: Vercel AI SDK + Polderfest 2027
-  Chat met data — alle 500 bands meesturen als context
-  Werkt voor 500. Werkt NIET voor 50.000.

Vandaag — Tool Calling

AI kiest zelf welke functie te gebruiken om aan info te komen.

Schaalbaar. Real-time. Multi-step. + write-acties mogelijk.

Planning

Vandaag — 180 minuten

Welkom + Terugblik + schaalprobleem	10 min
Theorie: wat is Tool Calling?	30 min
LIVE DEMO 1 — Eerste tool: searchBands	20 min
LIVE DEMO 2 — Multi-step + meer tools	20 min
Pauze	15 min
LIVE DEMO 3 — Tool-calls in UI tonen	25 min
LIVE DEMO 4 — Edge cases + errors	15 min
Vergelijking + Lesopdracht + Huiswerk	25 min
Vragen + Afsluiting	15 min

Concept

Wat is Tool Calling?

AI kiest zelf welke functie te gebruiken op basis van de vraag.



Wat win je?

Schaalbaar · Real-time · Type-safe · Multi-step · Write-acties mogelijk

Anatomie

description + parameters + execute

```
import { tool } from "ai";
import { z } from "zod";

const searchBands = tool({
  description: "Zoek bands op dag, stage, of genre",
  parameters: z.object({
    day: z.enum(["Vrijdag", "Zaterdag", "Zondag"]).optional(),
    stage: z.string().optional(),
    genre: z.string().optional(),
  }),
  execute: async ({ day, stage, genre }) => {
    /* Supabase query */
  },
});
```

description

Wat doet de tool? AI leest dit om te kiezen.

parameters

Zod schema. Type-safe. AI weet wat hij mag invullen.

execute

Wat gebeurt er? Async functie. Supabase / API / logic.

Multi-step

Eén vraag = meerdere tool-calls met maxSteps

Voorbeeld: 'Vergelijk top headliner met drukst geplande opener'

Stap 1

searchBands({ tier: "headliner" }) → 50 bands

Stap 2

searchBands({ tier: "opener" }) → 100 bands

Stap 3

AI vergelijkt + antwoordt

```
const result = streamText({  
  model: openai("gpt-4o-mini"),  
  tools: { searchBands, getStats, getBandByName },  
  maxSteps: 5,  
  messages,  
});
```

Default: 1 stap. Voor multi-step expliciet maxSteps zetten.

Vandaag bouwen we

6 tools voor Polderfest

searchBands

Filter op dag, stage, genre, tier

Read

getBandByName

Exact lookup

Read

getStats

Aggregate per groep

Read

getScheduleByDay

Slot-overzicht per dag

Read

addFavorite

User favoriet opslaan

Write

listFavorites

User favorieten ophalen

Read

Eerste tool: searchBands

Refactor Les 11 chat-route — weg met alle bands meesturen. ~20 min.

- 1 Oude code wegsnijten** `alle .select("*") + context-string`
- 2 Tool importeren** `import { tool } + zod`
- 3 searchBands definiëren** `description + parameters + execute`
- 4 POST refactoren** `tools + maxSteps + kortere system prompt`
- 5 Browse /chat** `vraag testen — tool wordt aangeroepen`

Meer tools + multi-step in actie

Drie tools toevoegen en multi-step demonstreren. ~20 min.

- 1** **getStats tool** voor 'hoeveel jazz acts?'
- 2** **getBandByName tool** voor 'vertel me over X'
- 3** **Tools registreren** tools: { searchBands, getStats, getBandByName }
- 4** **Multi-step vraag** 'Hoeveel jazz fusion? En top 3 ervan?'
- 5** **Multi-step zien werken** AI roept 2 tools achter elkaar aan



Pauze

15 minuten



LIVE DEMO 3

Tool-calls in UI tonen

Transparantie + debug. ~25 min.

1 **messages.parts uitleggen**

text-parts vs tool-invocation-parts

2 **UI rendering refactoren**

map over parts i.p.v. content

3 **Tool-call chip**

`? toolName({args})`

4 **Tool-result collapsible**

`<details><summary>Toon</summary>`

5 **Multi-step zichtbaar**

meerdere chips per AI-bericht

Edge cases + error handling

Vier scenario's testen. ~15 min.

1

Ongeldige enum input

'Welke bands op Donderdag?' → AI weigert netjes

2

Lege resultaten

'Death metal bands?' → '0 gevonden, eerlijk'

3

Database error

Tool returnt { error } → AI legt uit

4

Write-tool

addFavorite — schrijft echt in DB

Vergelijking

Tool Calling > context-all

Aspect	Les 11 (context-all)	Les 12 (Tool Calling)
Tokens / call	~30.000	~2.000
Schaal	~1000 records	Duizenden makkelijk
Live data	Snapshot	Actueel per call
Write-acties	Niet mogelijk	Wel (addFavorite)
Multi-step	Beperkt	Native (maxSteps)
Cost	Hoger	Lager

Wanneer toch context-all?

Hele kleine dataset · snel prototype · geen schaal nodig

Productie: bijna altijd Tool Calling.

Lesopdracht

Refactor jouw thema-app naar Tool Calling

Bouw voort op je app uit Les 11. Vervang context-all door tools.

- [] Refactor chat-route — geen `.select("*")` aan begin meer
- [] Minstens 3 tools definiëren voor jouw dataset
- [] Eén tool met enum parameters (vaste keuze)
- [] `maxSteps: 5` in `streamText`
- [] System prompt aanpassen — 'gebruik tools'
- [] 3 vragen testen — minstens 1 multi-step

Tool-ideeën:

`searchX(filter)` · `getXById(id)` · `getStats(groupBy)` · thema-specifiek

Write-tool + UI visualisatie + reflectie

A

Write-tool

Nieuwe tabel + write-tool (addFavorite e.d.) + list-tool

B

UI visualisatie

Tool-calls tonen in chat — chips + collapsed result

C

TOOLS.md

Documentatie: tools-lijst, 3 vragen, 1 edge-case

Bonus: loading indicator • mooie kaartjes • confirmation UI write

Deadline: vóór Les 13 (Agents).

Volgende les

Agents — autonome multi-step

Vandaag

maxSteps: 5 — AI roept tools aan in 1 ronde, max 5 stappen
Werkt voor zoek, vergelijking, simpele acties

Volgende les — Agents

maxSteps: 20+ — AI plant, voert uit, evalueert, herhaalt
Voorbeeld: 'Plan mijn volledige Polderfest weekend'
30+ tool-calls in één request — autonoom werkende AI

Daarna:

- Les 14: RAG + embeddings (semantic search op grote corpora)
- Les 15-16: Testing + Deployment • Les 17-18: Eindopdracht + Pitch

Afsluiting

Vragen?

Vandaag gezien:

- Schaalprobleem van context-all opgelost met Tool Calling
- Anatomie van een tool: description + parameters + execute
- maxSteps voor multi-step workflows
- Zes tools voor Polderfest gebouwd (5 read + 1 write)
- Tool-invocations gevisualiseerd in de UI
- Edge cases + error handling
- Vergelijking: Tool Calling > context-all op bijna alles

Volgende les: Agents

Hoe ver kan AI autonoom? 30+ tool-calls in één request, met stop-condities.