

Les 16 — Huiswerk

MCP server uitbreiden + Supabase + MCP.md

Vak:	AI-Assisted Development
Deadline:	Voor Les 17 — Externe APIs in diepte
Inleveren:	GitHub repo + MCP.md in root
Tijd:	~2 uur

Doel

Bouw je MCP server uit naar productie-niveau: meerdere tools incl. een write-tool, een resource, geladen in Cursor en Claude Desktop, en gedocumenteerd.

A

Tools naar Supabase (verplicht)

Vervang mock-data door echte Supabase queries.

Eisen:

- 3 read-tools werken tegen Supabase (searchBands, getBandStats, getBandByName)
- 1 write-tool werkt (addFavorite) met 'alleen bij user-intent' in description
- Env vars worden gelezen vanuit MCP config
- Alle tools getest via MCP Inspector

Voorbeeld write-tool:

```
server.tool("addFavorite",
  "Voeg band toe aan favorieten. Alleen bij expliciete user-intent.",
  { userEmail: z.string().email(), bandName: z.string() },
  async ({ userEmail, bandName }) => {
    const { data: band } = await supabase.from("bands")
      .select("id").ilike("name", bandName).single();
    if (!band) return { content: [{ type: "text", text: "Niet gevonden" }], isError: true };
    const { error } = await supabase.from("user_favorites")
      .insert({ user_email: userEmail, band_id: band.id });
    if (error) return { content: [{ type: "text", text: error.message }], isError: true };
    return { content: [{ type: "text", text: `Toegevoegd: ${bandName}` }] };
  }
);
```

B Resource toevoegen (verplicht)

Minimaal 1 resource: bv. bands-list, today-schedule, of stages-info.

```
server.resource("bands-list", "bands://list", async () => {
  const { data } = await supabase.from("bands").select("*");
  return { contents: [{
    uri: "bands://list",
    mimeType: "application/json",
    text: JSON.stringify(data, null, 2),
  }]};
});
```

Eisen:

- 1 resource gedefinieerd
- Test via Inspector: resource lijst en preview werken
- Test in Cursor: @server:resource-name werkt

C Laden in twee clients (verplicht)

Server moet werken in **zowel Cursor als Claude Desktop**.

- Cursor toont server met groen vinkje
- Claude Desktop toont MCP-icon, server beschikbaar
- Test-prompt werkt in beide

D MCP.md documentatie (verplicht)

Vijf secties:

1. **Tools** — tabel met naam, description, type (read/write)
2. **Resources** — tabel met URI + inhoud
3. **Voorbeeld-prompts** — 5 prompts die tools triggeren, met verwacht antwoord
4. **Setup voor anderen** — installatie-instructies
5. **Eén observatie** — iets dat opviel (verschil Cursor/Claude, tool-keuze AI, etc)

Max 600 woorden. Concrete voorbeelden + tool-calls.

E**Bonus (optioneel)**

- **Bonus 1** — eigen server .prompt () template
- **Bonus 2** — npm publish (anderen kunnen via npx installen)
- **Bonus 3** — HTTP/SSE variant naast stdio
- **Bonus 4** — MCP server voor je eindopdracht-data (direct nuttig!)

F**Beoordeling (10 pt — voldoende 6+)**

Criterium	Punten
A — 3 read-tools + 1 write-tool werkend	3
B — 1 resource werkend	1
C — geladen in 2 clients	2
D — MCP.md compleet (5 secties)	3
Server runt zonder errors	1
Totaal	10

Veelvoorkomende valkuilen

- Env vars niet gelezen — check env block in Cursor mcp.json
- Server crash bij Supabase — check service key vs anon key
- AI roept verkeerde tool — description specifieker maken
- Resource niet zichtbaar in Cursor — check Claude Desktop ook

Tips

- Test in Inspector eerst — sneller dan via Cursor
- Descriptions zijn cruciaal — AI kiest op basis hiervan
- `isError: true` bij failures
- Eindopdracht-tip: MCP server voor eigen data — direct in Cursor laden