

Forbind en AI-agent til Jira og Confluence på 30 minutter

GitHub Copilot CLI + Atlassians MCP-server. Lad en agent læse din Confluence-dokumentation, forfine dine features og skrive direkte i dine Jira-tickets – fra din terminal.

PRAKTISK MINIGUIDE · TIL PRODUCT OWNERS OG SCRUM MASTERS

Stefano Vincenti er GenAI-strateg og -arkitekt, ekstern lektor ved ITU og DIS Copenhagen og co-founder af BotTellMe. Han hjælper nordiske ledelsesteams med at gå fra AI-nysgerrige til konkret implementering.

[Flere guides på aitrainer.dk](#)

[Tilmeld nyhedsbrevet](#)

■ HVEM ER DET TIL, OG HVAD FÅR DU

Dine udviklere blev hurtigere med AI. Gæt så hvem der nu er flaskehalsen. PO'en eller scrum masteren, der sidder i Jira og Confluence og forsøger at følge med et team, der pludselig bygger dobbelt så hurtigt.

TIP – TAG EN AI-WINGMAN MED

Følg trinene i rækkefølge, men du behøver ikke gøre det alene. Hold en AI-assistent åben ved siden af dig som wingman. Når en kommando fejler, så indsæt fejlbeskeden og spørg, hvad der gik galt. Bed den oversætte et Windows-trin til macOS eller Linux, forklare hvad en indstilling gør, eller tjekke din **mcp-config.json**, før du gemmer. En stærk, aktuel model klarer det nemt – fx **Claude Opus 4.8** eller **Claude Sonnet 4.6**. Guiden giver dig vejen; din wingman får dig fri, når du sidder fast.

Denne guide giver dig en terminal-baseret AI-agent (GitHub Copilot CLI), der kan:

- **Læse** din Confluence-dokumentation for kontekst
- **Søge og læse** dine Jira-sager
- **Forfine** feature-beskrivelser, så de er tydelige, konsistente og klar til review
- **Skrive direkte tilbage** i Jira – og opdatere tickets, når du siger “ja”

Samlet opsætningstid: cirka **30 minutter**. Det svære er ikke agenten. Den del findes allerede. Det kræver bare, at du tør arbejde i terminalen og give agenten adgang til at læse din dokumentation og skrive i dine tickets.

ÆRLIG MARKERING

Alle kommandoer i denne guide er testet på Windows 11 med GitHub Copilot CLI og den open source-baserede **mcp-atlassian**-server. De samme trin virker på macOS og Linux med små kommando-ændringer. Dette er ikke et officielt Atlassian-produkt.

■ FØR DU GÅR I GANG – FORUDSÆTNINGER

Du skal bruge **alt** af følgende:

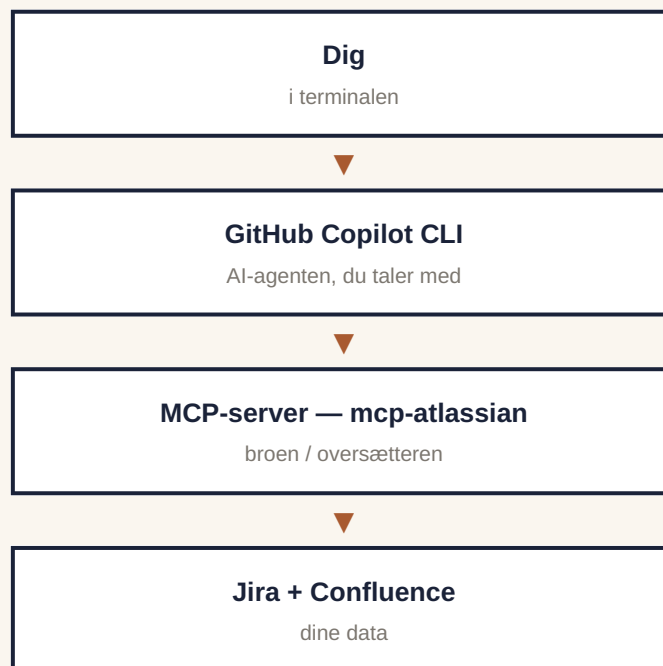
Forudsætning	Noter
En Windows 11-PC	Denne guide er skrevet og testet på Windows 11. De samme trin virker på macOS/Linux med små kommando-ændringer.

Forudsætning	Noter
En aktiv GitHub Copilot-licens	Copilot CLI er inkluderet i alle planer – Free, Pro, Pro+, Max og Business/Enterprise. På en arbejds-styret Business/Enterprise-licens skal jeres org-admin have slået Copilot CLI til. Forbruget trækker på planens månedlige AI Credits.
Adgang til dine Jira-projekter	Mindst ét Jira-projekt, du kan læse/redigere. Denne guide bruger Jira Data Center, men Jira Cloud virker også (auth-trinnet er anderledes – se Trin 3).
Adgang til dine Confluence-spaces	Læseadgang til de spaces, der rummer din dokumentation. Data Center her, men Cloud virker også.
Tilladelse til at oprette et Personal Access Token (PAT)	Din admin skal have aktiveret PAT'er. Jira/Confluence Data Center 8.14+ / 6.0+ understøtter PAT'er.

Sikkerhedsnote med det samme

Du genererer et token, der handler på dine vegne. Behandl det som et password. Vi dækker, hvor det gemmes, og hvordan du roterer det, til sidst.

■ DET STORE BILLEDE – SÅDAN HÆNGER DELENE SAMMEN



- **Copilot CLI** er agenten, du taler med.
- **MCP** (Model Context Protocol) er en åben standard, der lader agenten bruge eksterne værktøjer.
- **mcp-atlassian** er en færdig MCP-server, der ved, hvordan man taler med Jira og Confluence.

Du bygger ikke integrationen fra bunden. Du installerer og konfigurerer den bare.

■ OPSÆTNING I 7 TRIN

1 INSTALLÉR

Installér GitHub Copilot CLI

Åbn **PowerShell** (søg efter den i Start-menuen).

Mulighed A – WinGet (anbefales på Windows 11):

```
winget install GitHub.Copilot
```

Mulighed B – npm (hvis du allerede bruger Node.js 22+):

```
npm install -g @github/copilot
```

Når den er færdig, ser du *“Path environment variable modified; restart your shell.”* Luk PowerShell helt, og åbn et **nyt** vindue – kommandoen `copilot` genkendes ikke, før du gør det. (Det er den hyppigste “det virker ikke”-fælde.) Verificér derefter:

```
copilot --version
```

2 FØRSTE START

Første start og login

Gå ind i en arbejdsmappe (opret en, hvis du vil – **brug ikke systemmapper**), og start agenten:

```
cd $HOME
mkdir copilot-workspace
cd copilot-workspace
copilot
```

Den spørger **“Do you trust the files in this folder?”** Da det er din egen tomme arbejdsmappe, vælg **“Yes, and remember this folder.”** Giv aldrig tillid i en systemmappe som `C:\Windows\System32`. Log derefter ind ved at skrive `/login` og følg browser-prompten. Det gør du kun én gang. Skriv `/help` når som helst; afslut med `Ctrl+C`.

3

TOKENS

Hent dine Jira- og Confluence-PAT'er

Det er nøglen, der lader agenten handle på dine vegne. På Data Center opretter du **ét token i Jira** og **ét i Confluence** (de er separate apps).

1. Log ind på dit Jira Data Center-site i en browser.
2. Klik på din **avatar** (øverst til højre) → **Profile**.
3. Åbn **Personal Access Tokens** i menuen til venstre. *Kan du ikke se det?* Bed din admin om at aktivere PAT'er.
4. Klik **Create token**, giv det et navn (fx `copilot-mcp`), sæt en **udløbsdato**, opret det, og **kopier det med det samme** – du ser det ikke igen.

Gentag præcis de samme trin i **Confluence**. Du har nu **to tokens** gemt et sikkert sted (midlertidigt).

Bruger du Jira/Confluence Cloud i stedet?

Så opretter du ikke PAT'er – du opretter et **API-token** på `id.atlassian.com/manage-profile/security/api-tokens`. Til Cloud bruger du `JIRA_USERNAME` + `JIRA_API_TOKEN` (din e-mail + token) i stedet for `JIRA_PERSONAL_TOKEN`. Alt andet er det samme.

4

INSTALLÉR

Installér forudsætningen for MCP-serveren

`mcp-atlassian` -serveren kører via **uvx**, en lille, hurtig Python-værktøjskører. Én kommando:

```
winget install astral-sh.uv
```

Luk og genåbn PowerShell, og verificér:

```
uvx --version
```

`uvx` henter og kører `mcp-atlassian` on demand – du installerer ikke serveren separat eller styrer selv et Python-miljø.

Konfigurér MCP-serveren til Copilot CLI

Copilot CLI læser MCP-server-definitioner fra en config-fil i din hjemmemappe:

```
C:\Users\<<dit-brugernavn>\.copilot\mcp-config.json
```

Opret filen (hvis den ikke findes). **Til Jira/Confluence Data Center** (denne guides opsætning):

```
{
  "mcpServers": {
    "mcp-atlassian": {
      "type": "local",
      "command": "uvx",
      "args": ["mcp-atlassian"],
      "env": {
        "JIRA_URL": "https://jira.din-virksomhed.com",
        "JIRA_PERSONAL_TOKEN": "dit_jira_pat_her",
        "CONFLUENCE_URL": "https://confluence.din-virksomhed.com",
        "CONFLUENCE_PERSONAL_TOKEN": "dit_confluence_pat_her"
      }
    }
  }
}
```

Udfyld dine rigtige Data Center-URL'er og de to tokens fra Trin 3.

Virksomheds-certifikat (self-signed)?

Hvis din virksomhed bruger et internt certifikat, der ikke ligger i Windows' trust store, så tilføj `"JIRA_SSL_VERIFY": "false"` og `"CONFLUENCE_SSL_VERIFY": "false"` i `env`-blokken. Prøv uden dem først – virksomhedscertifikater, der allerede ligger i Windows Certificate Store, godkendes automatisk.

Til Jira/Confluence Cloud (alternativ), skift auth-nøglerne:

```
"env": {
  "JIRA_URL": "https://din-virksomhed.atlassian.net",
  "JIRA_USERNAME": "din.email@virksomhed.com",
  "JIRA_API_TOKEN": "dit_api_token",
  "CONFLUENCE_URL": "https://din-virksomhed.atlassian.net/wiki",
  "CONFLUENCE_USERNAME": "din.email@virksomhed.com",
  "CONFLUENCE_API_TOKEN": "dit_api_token"
}
```

Valgfrit, men anbefalet – afgræns den. Tilføj nogle af disse i `env` for at holde agenten fokuseret og sikker:

```
"JIRA_PROJECTS_FILTER": "PROJ,DEV",  
"CONFLUENCE_SPACES_FILTER": "TEAM,DOCS",  
"READ_ONLY_MODE": "false"
```

TIP TIL DIN FØRSTE SESSION

Start med **READ_ONLY_MODE** sat til **"true"**. Bekræft, at agenten kan læse dine data, og skift den så til **"false"** for at tillade opdateringer. Filtrene begrænser agenten til bestemte projekter/spaces.

6 FORBIND

Genstart og bekræft forbindelsen

Afslut Copilot CLI helt (**Ctrl+C**), og start igen med `copilot`. Tjek så, at serveren er forbundet:

```
/mcp
```

Du bør se **mcp-atlassian** listet som forbundet, med værktøjer som `jira_search` og `confluence_search`. Test med en rigtig prompt:

```
Vis mine åbne Jira-sager
```

Første gang agenten bruger et værktøj, beder den om din tilladelse – gennemgå og godkend. Hvis "åbne sager" ser forkert ud, så vær eksplicit:

```
Søg i Jira med JQL: assignee = currentUser()  
AND statusCategory != Done ORDER BY updated DESC
```

Ser du dine sager – **så er du færdig.**

Sæt den i arbejde

Nu den del, der fjerner flaskehalsen. Prøv prompts som disse.

Hent kontekst fra Confluence:

Søg i Confluence i TEAM-spacet efter vores retningslinjer for feature-definitioner, og opsummer den standard, vi bruger til at skrive feature-beskrivelser.

Forfin et sæt features:

Kig på de åbne features i projekt PROJ. For hver enkelt: læs eventuelle linkede Confluence-dokumenter for kontekst, og omskriv så beskrivelsen, så den er tydelig, konsistent og klar til review. Vis mig de foreslåede ændringer først.

Skriv tilbage til Jira (efter du har godkendt):

De ser gode ud. Opmåler hver Jira-sag med den forfinede beskrivelse.

Agenten viser, hvad den vil ændre, og **venter på dit "ja"**, før den skriver noget tilbage. Du har styringen; den taster.

■ FEJLFINDING

Symptom	Løsning
<code>copilot</code> genkendes ikke	Du genstartede ikke PowerShell efter installation. Luk og åbn vinduet igen.
<code>/mcp</code> viser ikke mcp-atlassian	Tjek, at <code>~/.copilot/mcp-config.json</code> findes og er gyldig JSON (ingen efterstillede kommaer). Afslut helt, og start <code>copilot</code> igen.
<code>uvx</code> blev ikke fundet	Genåbn PowerShell efter installation af <code>uv</code> ; kørs <code>uvx --version</code> for at bekræfte.
Auth-/401-fejl	Tjek token-værdier og URL'er igen. Bekræft, at PAT'en ikke er udløbet, og at du brugte det rigtige token til hver tjeneste.
SSL-/certifikatfejl	Tilføj <code>JIRA_SSL_VERIFY</code> / <code>CONFLUENCE_SSL_VERIFY</code> sat til "false" (Trin 5) – kun hvis din virksomhed bruger self-signed certifikater.

Symptom	Løsning
---------	---------

Agenten læser, men opdaterer ikke

`READ_ONLY_MODE` er "true". Sæt den til "false", og start igen.

■ SIKKERHED – GØR DET HER, NÅR DU ER SAT OP

Dine tokens gemmes i **klartekst** i `mcp-config.json`. Det er normalt for MCP-configs, men behandl filen med omtanke:

1. **Commit aldrig** `mcp-config.json` til et Git-repository, og indsæt aldrig tokens i chats/screenshots.
2. Sæt en **fornuftig udløbsdato** på hvert token.
3. **Rotér** et token med det samme, hvis du tror, det er blevet eksponeret: opret et nyt, opdatér config'en, og tilbagekald det gamle i Jira/Confluence.
4. Brug **filtrene** og **read-only-tilstanden** fra Trin 5 til at begrænse, hvad agenten kan røre.

■ HURTIGT OVERBLIK

1. Installér **Copilot CLI** → genstart shell.
2. `copilot` → giv din mappe tillid → `/login`.
3. Opret **Jira- + Confluence-PAT'er**.
4. Installér **uv / uvx**.
5. Tilføj **mcp-atlassian** til `~/.copilot/mcp-config.json`.
6. Genstart → `/mcp` → test.
7. Lad agenten forfine features og opdatere Jira.

Det er det. Ikke raketvidenskab – en halv times opsætning, og så arbejder den *med dig i stedet for*, at du arbejder for den.

Stefano Vincenti

GenAI-strateg og -arkitekt · 25 år i IT og digital transformation · Co-founder af BotTellMe · Ekstern lektor ved ITU og DIS Copenhagen · Partner hos TryZone.

LinkedIn [linkedin.com/in/stefanovincenti](https://www.linkedin.com/in/stefanovincenti) · **Nyhedsbrev** stefanovincenti.substack.com

Disclaimer

Dette er generel, praktisk vejledning – ikke juridisk, sikkerheds- eller compliance-rådgivning – og det er ikke et officielt Atlassian-, GitHub- eller Microsoft-produkt. At give en AI-agent læse- og skriveadgang til din Jira og Confluence rører din organisations data og adgangsnøgler: afklar det med din egen IT-, sikkerheds- og compliance-funktion, og følg din virksomheds politik for tredjepartsværktøjer, før du slår skrive- og læseadgang til i et produktionsmiljø. Personal Access Tokens handler på dine vegne – behandl dem som passwords; du er selv ansvarlig for, hvordan de gemmes, afgrænses og roteres. Værktøjer, planer, priser, vilkår og sikkerhedsadfærd ændrer sig hurtigt: bekræft altid de aktuelle forhold mod primære kilder (GitHub Docs, mcp-atlassian-repositoriet på github.com/sooper/mcp-atlassian og din egen Atlassian-admin), før du beslutter dig. Brug af tredjeparts- og open source-software sker på eget ansvar. Øjebliksbillede pr. juni 2026.

Stefano Vincenti · AI, Built Human