Table of Contents

Directo MCP servera uzstādīšana — Claude Desktop un Gemini CLI 3
1. Ievads — kāpēc MCP? 3
Kas ir MCP? 3
Kāds ir ieguvums? 3
Arhitektūra 3
2. Priekšnosacījumu pārbaude 4
2.1 Nepieciešams visiem 4
2.2 Papildus Claude Desktop lietotājiem 4
2.3 Papildus Gemini CLI lietotājiem 5
2.4 Administratora tiesības — kad tās vajadzīgas? 5
3. Kopīgie soļi (jāizdara vienu reizi) 6
Solis 3.1 — Node.js instalēšana (obligāti) 6
Izvēlies savu ceļu 7
A CEĻŠ — ātrā instalācija ar skriptiem (ieteicams) 7
B CEĻŠ — manuāli, soli pa solim (alternatīva A ceļam) 9
Solis B1 — PowerShell izpildes politikas iestatīšana 9
Solis B2 — CredentialManager modulis 9
Solis B3 — MCP servera mapes izveide 10
Solis B4 — directo-mcp.js ievietošana 10
Solis B5 — package.json izveide 11
Solis B6 — Atkarību instalēšana 11
Solis B7 — API atslēgas droša saglabāšana 11
4. Claude Desktop uzstādīšana 12
Solis 4.1 — Konfigurācijas faila atrašanās vieta 12
Solis 4.2 — Konfigurācijas fails 13
Solis 4.3 — Claude Desktop restartēšana 14
Solis 4.4 — Pārbaude 14
5. Gemini CLI uzstādīšana 15
Solis 5.1 — Gemini CLI instalēšana 15
Solis 5.2 — Google API atslēgas uzstādīšana 16
Solis 5.3 — Konfigurācijas faila atrašanās vieta 16
Solis 5.4 — MCP servera pievienošana 16
Solis 5.5 — Mapes uzticēšana (Trusted Folders) 17
Solis 5.6 — Pārbaude 18
6. Problēmu novēršana 18
Biežākās kļūdas (abiem klientiem) 18
Claude Desktop specifiskās kļūdas 19
Gemini CLI specifiskās kļūdas 19
7. MCP rīki 20
8. Drošība 20
9. Atsauces 20

Directo MCP servera uzstādīšana — Claude Desktop un Gemini CLI

Šī pamācība skaidro, kā savienot Directo ERP sistēmu ar MI asistentu (Claude vai Gemini), izmantojot lokālu MCP (Model Context Protocol) serveri. Pamācība paredzēta gan Claude Desktop, gan Gemini CLI lietotājiem — kopīgie soļi jāizdara tikai vienu reizi, pēc tam izvēlies sadaļu, kas atbilst tavam MI klientam.

1. Ievads — kāpēc MCP?

Kas ir MCP?

MCP (Model Context Protocol) ir atvērts standarts, kas ļauj MI asistentiem sazināties tieši ar ārējām sistēmām — datubāzēm, API (lietojumprogrammu saskarnēm) un lietotnēm.

Bez MCP darbs ar MI asistentu izskatās šādi:

1. Tu nokopē datus no Directo
2. Tu ielīmē tos MI sarunā
3. MI atbild
4. Tu kopē atbildi atpakaļ Directo

Ar MCP tas izskatās šādi:

1. Tu pajautā dabiskā valodā: „Kuriem artikuliem atlikums ir zem minimuma?”
2. MI pats vaicā Directo API (lietojumprogrammu saskarnei)
3. MI analizē datus un atbild uzreiz

Kāds ir ieguvums?

Situācija	Bez MCP	Ar MCP
Noliktavas atlikuma pārbaude	Atver Directo → meklē → kopē → ielīmē MI	„Kuriem artikuliem noliktavā ir mazāk par 10 vienībām?”
Rēķinu analīze	Eksportē CSV → augšupielādē MI → jautā	„Kuri klienti nav maksājuši pēdējās 90 dienās?”
Pārdošanas atskaite	Veido manuāli → kopē → analizē	„Salīdzini šī mēneša pārdošanu ar pagājušo gadu”
Pasūtījumu izsekošana	Atver vairākus skatus → kopē informāciju	„Kuri pasūtījumi kavējas vairāk par 5 dienām?”

Arhitektūra

```
Tu (dabiskā valodā)
        ↕
MI klients (Claude Desktop vai Gemini CLI)
```

```
        ↕  MCP protokols (lokāli)
MCP serveris (tavā datorā, Node.js)
        ↕  HTTPS
Directo API (mākonis)
```

MCP serveris darbojas tavā datorā. Directo API atslēga nekad nepamet tavu datoru. Taču no Directo iegūtie dati tiek nosūtīti tālāk uz MI pakalpojuma sniedzēja mākoni, lai modelis tos varētu izlasīt un formulēt atbildi dabiskā valodā — tieši tāpat kā jebkurš cits sarunas saturs.



Lokāla instalācija — svarīgi saprast:

- MCP serveris tiek uzstādīts tikai tavā paša datorā
- Tas netiek koplietots — kolēģi nevar piekļūt tavam MCP serverim
- Katram lietotājam visi soļi jāizdara pašam, savā datorā
- MCP serveris darbojas tikai tad, kad tavs dators ir ieslēgts un MI klients ir palaists
- Ja vēlies MCP serveri koplietot vairākiem lietotājiem, nepieciešams atsevišķs mitināta servera risinājums (šajā pamācībā netiek aplūkots)

2. Priekšnosacījumu pārbaude

Pirms instalēšanas pārbaudi visus priekšnosacījumus. Neizpildīts priekšnosacījums ir visbiežākais neveiksmes iemesls.

2.1 Nepieciešams visiem

Priekšnosacījums	Kā pārbaudīt	Risinājums
Windows 10 vai 11 (64 bitu)	Sākt → Iestatījumi → Sistēma → Par	—
Node.js 22 vai jaunāka (LTS)	`node –version`	nodejs.org → LTS
npm	`npm –version` (darbojas tikai pēc izpildes politikas iestatīšanas, skat. zemāk)	Nāk komplektā ar Node.js
PowerShell izpildes politika	`Get-ExecutionPolicy` → jāatgriež `RemoteSigned`	Skat. zemāk
CredentialManager modulis	`Get-Module -ListAvailable CredentialManager`	Skat. zemāk
Directo API atslēga	Directo → Uzstādījumi → Sistēmas uzstādījumi → API	Jautā savam Directo administratoram

2.2 Papildus Claude Desktop lietotājiem

Priekšnosacījums	Kā pārbaudīt	Saite
Uzstādīts Claude Desktop	Claude parādās izvēlnē Sākt	https://claude.ai/download
Anthropic konts	Pieteikšanās darbojas	https://claude.ai

Oficiālā dokumentācija: https://docs.claude.ai

2.3 Papildus Gemini CLI lietotājiem



⚠ Gemini CLI bezmaksas līmenis joprojām pastāv, taču ir mainījies. Atkarībā no autentifikācijas veida:

- **Google konts** (Gemini Code Assist, individuāls lietotājs): ~60 pieprasījumi minūtē un ~1000 pieprasījumi dienā
- **Bezmaksas Gemini API atslēga** (neapmaksāta): šaurāk — ~10 pieprasījumi minūtē, ~250 pieprasījumi dienā, tikai Flash modeļi

2026. gada 18. jūnijā Gemini CLI neapmaksātajam līmenim un Google One lietotājiem tiks aizstāts ar Antigravity CLI — tas skar konkrētu autentifikācijas ceļu, nevis bezmaksas piekļuvi kopumā. Lielākam apjomam vai Pro modeļiem nepieciešams Google AI Pro/Ultra abonements vai Gemini Code Assist licence. Sava projekta faktiskos ierobežojumus pārbaudi AI Studio, jo tie bieži mainās un ir atkarīgi no reģiona un konta statusa.

Priekšnosacījums	Kā pārbaudīt	Saite
Google konts	Piekļuve gmail.com darbojas	https://accounts.google.com
Google AI Studio API atslēga	https://aistudio.google.com/apikey	Izveido API atslēgu
Uzstādīts Gemini CLI	`gemini –version`	`npm install -g @google/gemini-cli`

Oficiālā dokumentācija: https://geminicli.com/docs

2.4 Administratora tiesības — kad tās vajadzīgas?



Labā ziņa: gandrīz visus soļus var izdarīt bez administratora tiesībām. Administratora tiesības patiesi nepieciešamas tikai vienā vietā — Node.js instalēšanai.

Solis	Administratora tiesības?	Iemesls
Node.js instalēšana	Jā — to pieprasa instalētājs	Sistēmas līmeņa instalācija (Program Files)
PowerShell izpildes politika	Nē	`-Scope CurrentUser` raksta HKCU reģistrā
CredentialManager modulis	Nē	`-Scope CurrentUser` instalē lietotāja profilā
`npm install` (MCP serveris)	Nē	Lietotāja līmeņa instalācija
Claude Desktop konfigurācija	Nē	Lietotāja mapes
Gemini CLI konfigurācija	Nē	Lietotāja mapes
API atslēgas saglabāšana	Nē	Credential Manager, lietotāja līmeņa ieraksts

Riski un svarīgs brīdinājums:

- Node.js instalētājs — zems risks, oficiāla programmatūra
- **Nevienu no pārējiem soļiem neatver kā administrators**, pat ja iepriekšēja šīs pamācības versija vai internetā atrasti piemēri to iesaka — `-Scope CurrentUser` pastāv tieši tāpēc, lai administratora tiesības nebūtu vajadzīgas
- Ja tava ikdienas konta un administratora tiesību konts ir divi atsevišķi Windows konti (bieži uzņēmumu un domēna vidēs), atverot kaut ko kā administrators, darbība tiek veikta *otra* konta profilā — modulis, iestatījums vai atslēga pēc tam „pazūd" no tava parastā lietotāja skatpunkta. Sīkāk skat. problēmu novēršanas nodaļā.
- Nepalaid `npm install` kā administrators — tas rada atļauju konfliktus
- `npm install -g` (piemēram, Gemini CLI, solis 5.1) prasa administratora tiesības tikai tad, ja npm globālais prefikss norāda uz Program Files mapi — solī 5.1 ir aprakstīts risinājums, kas no tā izvairās

3. Kopīgie soļi (jāizdara vienu reizi)



Šos soļus izdari savā datorā. Ja kolēģis vēlas izmantot to pašu risinājumu, viņam šie soļi jāizdara patstāvīgi savā datorā.

Solis 3.1 — Node.js instalēšana (obligāti)



Šis ir vienīgais solis, kas nepieciešams abos zemāk aprakstītajos ceļos un ko nevar automatizēt (tam vajadzīgas administratora tiesības). Izdari to pirms turpināšanas neatkarīgi no tā, kuru ceļu izvēlies.

1. Ej uz https://nodejs.org un lejupielādē LTS versiju (22 vai jaunāku)
2. Palaid lejupielādēto `.msi` failu kā administrators
3. Instalē ar noklusējuma iestatījumiem — nemaini instalācijas atrašanās vietu

Pārbaudi instalāciju (parastā PowerShell, nevis administratora režīmā):

```
node --version
```

Gaidāmais rezultāts:

```
v22.11.0 vai jaunāka
```



Mēs apzināti šeit vēl nepārbaudām `npm –version` — npm ir PowerShell skripts (`npm.ps1`) un pirms izpildes politikas iestatīšanas (A ceļa skripts vai solis B1) dos kļūdu `running scripts is disabled`, pat ja Node.js un npm paši ir instalēti pareizi. Tā nav kļūda, bet gaidāma uzvedība šajā posmā. `npm –version` darbosies pēc A vai B ceļa pabeigšanas.



Ja PowerShell saka „not recognized” — aizver logu un atver to no jauna. Node.js pievieno sevi PATH mainīgajam tikai jaunās PowerShell sesijās.

Izvēlies savu ceļu



Zemāk ir divi alternatīvi ceļi uz vienu un to pašu rezultātu — strādājošu MCP serveri ar uzstādītu API atslēgu un MI klientu. Izdari tikai VIENU no tiem, nevis abus.

	A ceļš — skripti	**B ceļš — manuāli**
Laiks	~5 minūtes	~20 minūtes
Kam piemērots	Lielākajai daļai lietotāju	Vēlies pats izprast katru soli vai A ceļš tavā vidē nedarbojas
Aptver soļus	A1—A2	B1—B7

Ja izvēlies A ceļu un tas izdodas, B ceļu (B1—B7) izlaid pavisam — tie soļi manuāli apraksta tieši to, ko skripti jau ir izdarījuši tavā vietā. B ceļš ir problēmu novēršanas atsauces materiāls, nevis papildu obligāts solis.

A CEĻŠ — ātrā instalācija ar skriptiem (ieteicams)

Solis A1 — Failu lejupielāde

Lejupielādē `directo-mcp-files.zip` un atarhivē to:

directo-mcp-files.zip



Faili wiki ir pievienoti kā ZIP arhīvs, nevis kā atsevišķas saites — DokuWiki pēc noklusējuma bloķē `.js` un `.ps1` failu augšupielādi (drošības apsvērumu dēļ, skat. `conf/mime.conf`), bet `.zip` ir atļauts. Nepārsauc failus uz citu paplašinājumu (piemēram, `.txt`) kā „risinājumu” — tieši nepareizs paplašinājums vēlāk izraisa `MODULE_NOT_FOUND` tipa kļūdas.



⚠ **Pašreizējais ierobežojums:** šajā arhīvā esošie skripti savus statusa ziņojumus joprojām izvada igauniski (piemēram, „Kontrollin Node.js-i”). Tie darbojas vienādi neatkarīgi no ziņojumu valodas — atšķiras tikai ekrānā redzamais teksts. Skriptu angļu valodas versija ir plānota; līdz tam izmanto šīs pamācības soļu aprakstus, lai saprastu, ko nozīmē katrs izvades posms.

Atarhivē to (labais klikšķis uz ZIP → Extract All, vai PowerShell):

```
Expand-Archive "$env:USERPROFILE\Downloads\directo-mcp-files.zip" -
DestinationPath "$env:USERPROFILE\Downloads\directo-mcp-files"
cd "$env:USERPROFILE\Downloads\directo-mcp-files"
```

Pirms skriptu palaišanas jānoņem divi Windows noklusējuma drošības ierobežojumi — abi ir obligāti, nevis pēc izvēles:

1) Atļaut skriptu izpildi PowerShell vispār:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```



Bez tā katrs nākamais skripts beigsies ar kļūdu `running scripts is disabled on this system` / `UnauthorizedAccess`. `-Scope CurrentUser` neprasa administratora tiesības.

2) Noņemt Windows atzīmi „lejupielādēts no interneta” (Mark of the Web):

```
Get-ChildItem "$env:USERPROFILE\Downloads\directo-mcp-files" -Recurse |
Unblock-File
```



Arī tas ir obligāti, nevis pēc izvēles — caur pārlūku lejupielādēti faili pēc noklusējuma tiek bloķēti pat tad, ja izpildes politika jau ir pareiza. Windows atzīmē no interneta lejupielādētos failus ar atsevišķu drošības marķieri, kas pārnesas arī uz failiem, kas izvilkti no ZIP arhīva — `RemoteSigned` šādiem failiem pieprasa ciparparakstu, kura mūsu skriptiem nav. Bez šī soļa redzēsi `is not digitally signed` / `UnauthorizedAccess`, kas mulsinoši atgādina 1. punkta kļūdu, bet tai ir cits cēlonis un cits risinājums.

Arhīvā ir četri faili: `directo-mcp.js` (pats MCP serveris), `set-directo-apikey.ps1`, `install-claude-mcp.ps1`, `install-gemini-mcp.ps1`.

Solis A2 — Skriptu palaišana

Tajā pašā PowerShell logā (ne kā administrators), tikko atarhivētajā mapē:

```
# API atslēga (kopīga abiem klientiem, dari to vispirms)
.\set-directo-apikey.ps1

# Izvēlies savu MI klientu (vai abus):
.\install-claude-mcp.ps1
.\install-gemini-mcp.ps1
```

Katrs skripts beigās izvada kopsavilkumu — kas ir izdarīts (□) un kas vēl jādara manuāli (□).



Ja skripti nostrādāja veiksmīgi, 3. nodaļa tev ir pabeigta. Ej uzreiz uz 4.3. nodaļu (Claude Desktop) un/vai 5. nodaļu (Gemini CLI) — B ceļš (zemāk) nav vajadzīgs.

B CEĻŠ — manuāli, soli pa solim (alternatīva A ceļam)



Šī ir alternatīva A ceļam, nevis papildinājums tam. Dari to tikai tad, ja A ceļa skripti nenostrādāja vai ja vēlies pats iziet cauri katram solim. Ja A ceļš jau izdevās, šo sadaļu izlaid.

Solis B1 — PowerShell izpildes politikas iestatīšana

Pārbaudi pašreizējo iestatījumu:

```
Get-ExecutionPolicy
```

Ja atbilde nav `RemoteSigned`, izlabo to parastā PowerShell logā (ne kā administrators):

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```



`-Scope CurrentUser` ieraksta iestatījumu tikai HKEY_CURRENT_USER reģistra zarā, nevis visai iekārtai kopīgā vietā — tāpēc administratora tiesības nav vajadzīgas. Tās būtu vajadzīgas tikai `-Scope LocalMachine` gadījumā, ko šī pamācība neizmanto.

Tiks prasīts apstiprinājums → nospied Y → Enter.

Solis B2 — CredentialManager modulis

Atver parastu PowerShell logu (ne kā administrators) — tieši ar to pašu lietotāja kontu, ar kuru vēlāk darbosies MCP serveris:

```
Install-Module -Name CredentialManager -Force -Scope CurrentUser
```



`-Scope CurrentUser` pastāv tieši tāpēc, lai administratora tiesības nebūtu vajadzīgas — modulis tiek instalēts tavā paša profilā (`Documents\WindowsPowerShell\Modules`). Ja komanda prasa apstiprināt NuGet nodrošinātāju vai „neuzticamu repozitoriju", atbildi Y — arī šie soļi neprasa administratora tiesības.

Pārbaudi, vai instalācija izdevās (tajā pašā parastajā PowerShell logā):

```
Get-Module -ListAvailable CredentialManager
```

Jāparāda moduļa nosaukums un versija.

⚠ Ja tavs ikdienas darba konts un administratora tiesības nāk no diviem atsevišķiem Windows kontiem (bieži uzņēmumu un domēna vidēs), pārliecinies, ka šis solis un vēlākais solis B7 tiek izpildīti ar to pašu kontu, ar kuru vēlāk darbosies node `directo-mcp.js`. PowerShell logs, kas atvērts ar paaugstinātām tiesībām, faktiski var darboties cita konta vārdā — tad modulis vai API atslēga paliks nepareizajā profilā un vēlākais solis neizdosies bez redzama paskaidrojuma.

Solis B3 — MCP servera mapes izveide

Atver parastu PowerShell logu (ne administratora) un palaid:

```
New-Item -ItemType Directory -Force -Path "$env:USERPROFILE\directo-mcp"
```

Vienmēr izmanto mainīgo `$env:USERPROFILE`, nevis manuāli raksti `C:\Users\TAVSVARDS`. Domēnam pievienotos datoros (uzņēmumu/AD vidēs) tava faktiskā profila mape var atšķirties no pieteikšanās vārda — piemēram, tajā var būt domēna sufikss (`TAVSVARDS.DOMENS`). `$env:USERPROFILE` vienmēr dod īsto, pareizo ceļu.

Solis B4 — directo-mcp.js ievietošana

Šajā failā ir visi 32 Directo API rīki, un tas ir pilnībā pārbaudīts pret shēmām — katrs parametrs ir salīdzināts ar Directo faktiskajām `ws_*.json` shēmām. Nekopē šo failu manuāli no Notepad vai kāda cita avota — vienmēr izmanto svaigu, oficiālu kopiju, lai nesanāktu novecojusi vai ar nepareiziem parametriem veidota versija.

Lejupielādē failu `directo-mcp.js` — tas ir iekļauts A ceļā saistītajā `directo-mcp-files.zip` arhīvā (atarhivē to, skat. soli A1) — un ievieto to šajā mapē:

```
$env:USERPROFILE\directo-mcp\directo-mcp.js
```

Vienkāršākais veids: ja fails jau ir atarhivētajā Downloads apakšmapē, nokopē to ar PowerShell:

```
Copy-Item "$env:USERPROFILE\Downloads\directo-mcp-files\directo-mcp.js"
"$env:USERPROFILE\directo-mcp\directo-mcp.js"
```

Pārliecinies, ka faila nosaukums ir tieši `directo-mcp.js`, nevis `directo-mcp.js.txt`. Pārbaudi:

```
Test-Path "$env:USERPROFILE\directo-mcp\directo-mcp.js"
```

Jāatgriež `True`. Ja `False`, pārbaudi precīzu faila nosaukumu:

```
Get-ChildItem "$env:USERPROFILE\directo-mcp"
```

Solis B5 — package.json izveide

Atver jaunu Notepad logu un ielīmē:

```
{
  "name": "directo-mcp",
  "version": "1.0.0",
  "type": "module",
  "description": "Directo ERP MCP Server",
  "main": "directo-mcp.js",
  "scripts": {
    "start": "node directo-mcp.js"
  },
  "dependencies": {
    "@modelcontextprotocol/sdk": "^1.0.0",
    "zod": "^3.22.0"
  }
}
```

Saglabā to tajā pašā mapē ar nosaukumu `package.json` (faila tips: All Files).

Solis B6 — Atkarību instalēšana

Parastā PowerShell (ne administratora):

```
cd "$env:USERPROFILE\directo-mcp"
npm install
```

Veiksmīga instalācija izveido šajā mapē `node_modules` mapi un `package-lock.json` failu.



Šī ir lokāla `npm install` instalācija (nonāk `directo-mcp` mapes paša `node_modules` apakšmapē), nevis globāla — tai nekad nav vajadzīgas administratora tiesības vai īpaša npm prefiksa konfigurācija neatkarīgi no tā, kā Node.js sākotnēji tika instalēts.

Solis B7 — API atslēgas droša saglabāšana

Directo API atslēga tiek saglabāta Windows Credential Manager — šifrēti un tikai tava lietotāja konta ietvaros. Neviens cits šajā datorā šo atslēgu neredz.

```
New-StoredCredential -Target DIRECTO_API_KEY -UserName directo -Password
```

```
YOUR_REAL_KEY_HERE -Persist LocalMachine
```

Pārbaudi, vai saglabāšana izdevās:

```
(Get-StoredCredential -Target
DIRECTO_API_KEY).GetNetworkCredential().Password
```

Jāparāda tava API atslēga.

API atslēgas nomaiņa nākotnē:

```
Remove-StoredCredential -Target DIRECTO_API_KEY
New-StoredCredential -Target DIRECTO_API_KEY -UserName directo -Password
NEW_KEY -Persist LocalMachine
```



Izpildi šo soli tieši ar to pašu lietotāja kontu, ar kuru vēlāk darbosies `node directo-mcp.js`. Skat. brīdinājumu solī B2 par atsevišķa administratora konta risku.



A ceļa skripts (`set-directo-apikey.ps1`) dara to pašu drošāk (prasa atslēgu, neattēlojot to ekrānā) un der arī atslēgas nomaiņai — izmanto to, ja vari.

4. Claude Desktop uzstādīšana



Vispirms pabeidz 3. nodaļu (solis 3.1 + A vai B ceļš). Claude Desktop uzstādīšana paredz, ka MCP serveris jau eksistē.



MCP serveris darbojas tikai **tavā paša datorā** un tikai **Claude Desktop lietotnē** — nevis claude.ai tīmekļa versijā un ne citu lietotāju datoros.



Ieteicams: ja A ceļā (3. nodaļa) palaidi skriptu `install-claude-mcp.ps1`, soļi 4.1—4.2 jau ir izdarīti — skripts pats atrod, papildina un dublē konfigurācijas failu. Ej uzreiz uz soli 4.3.

Solis 4.1 — Konfigurācijas faila atrašanās vieta

Claude Desktop Windows vidē ir divas iespējamās konfigurācijas atrašanās vietas atkarībā no tā, kā tas tika instalēts:

Instalācijas veids	Atrašanās vieta
Tiešais `.exe` instalētājs (parastais)	`%APPDATA%\Claude\`
Microsoft Store / MSIX	`%LOCALAPPDATA%\Packages\Claude_pzs8sxrjxfjjc\LocalCache\Roaming\Claude\`



Šī ir dokumentēta, pašlaik aktuāla Claude Desktop problēma: MSIX instalācijas gadījumā Windows caurspīdīgi virtualizē `%APPDATA%\Claude\` atrašanās vietu uz citu ceļu. Ja labo nepareizo (standarta) failu, izmaiņas lietotni nekad nesasniedz un MCP serveris klusējot netiek ielādēts, nedodot nekādu paskaidrojošu kļūdu.

Pārbaudi, kurš variants attiecas uz tavu datoru — atver File Explorer un izmēģini adreses joslā abus:

```
%APPDATA%\Claude
```

Ja tur jau ir fails `claude_desktop_config.json` (vai Claude Desktop tur jau kaut ko izveidojis), izmanto šo ceļu. Ja nē, izmēģini:

```
%LOCALAPPDATA%\Packages\Claude_pzs8sxrjxfjjc\LocalCache\Roaming\Claude
```



Ja neviena no mapēm neeksistē, palaid Claude Desktop vismaz vienu reizi — tas automātiski izveido vajadzīgās mapes (parasti standarta ceļā).

Solis 4.2 — Konfigurācijas fails

Izveido failu ar nosaukumu `claude_desktop_config.json` mapē, ko noteici solī 4.1.



Veidojot failu ar Notepad: File → Save As → faila tips All Files (*.*) → nosaukums: `claude_desktop_config.json`

Faila saturs:

```
{
  "mcpServers": {
    "directo": {
      "command": "node",
      "args": [
        "C:\\Users\\YOURNAME\\directo-mcp\\directo-mcp.js"
      ]
    }
  }
}
```



Aizstāj YOURNAME ar savas faktiskās profila mapes nosaukumu — nevis ar pieņemto lietotājvārdu. Domēnam pievienotos (AD) datoros faktiskā profila mape var atšķirties no pieteikšanās vārda (piemēram, saturēt domēna sufiksu). Pārbaudi precīzo vērtību:

```
$env:USERPROFILE
```

JSON failā izmanto tieši to, ko komanda atgriež (ar dubultajām slīpsvītrām). Piemērs ilustrācijai — ja `$env:USERPROFILE` atgriež `C:\Users\john.user.COMPANY`, tad ceļam JSON failā jābūt `C:\\Users\\john.user.COMPANY\\directo-mcp\\directo-mcp.js`. Vai domēna sufikss (`.COMPANY`) ir vai nav, atkarīgs no tava datora konfigurācijas — vienmēr izmanto savas komandas faktisko izvadi, nevis šo piemēru.



Ja failā jau ir cits saturs (paša Claude Desktop iestatījumi, piemēram, `preferences` bloks) — **nepārraksti to**. `mcpServers` jābūt faila saknes līmenī, tajā pašā līmenī kā `preferences`, nevis tā iekšpusē. Ja neesi drošs par pareizu figūriekavu izvietojumu, izmanto A ceļa skriptu, kas labo failu kā struktūru (`ConvertFrom-Json`/`ConvertTo-Json`), nevis kā tekstu — tas pilnībā novērš figūriekavu kļūdas.



API atslēga konfigurācijas failā netiek pievienota — serveris to automātiski nolasa no Credential Manager (A ceļa skripts vai solis B7).

Solis 4.3 — Claude Desktop restartēšana

1. Labais klikšķis uz Claude ikonas sistēmas teknē (pie pulksteņa, uzdevumjoslā) → Quit / Exit
2. Ar loga aizvēršanu vien nepietiek — Claude Desktop turpina darboties fonā
3. Ja Quit nav pieejams — atver uzdevumu pārvaldnieku → atrodi Claude procesu → End Task
4. Atver Claude Desktop no jauna un uzgaidi dažas sekundes

Solis 4.4 — Pārbaude

Claude Desktop sarunā ieraksti:

> Parādi visus noliktavas atlikumus

Claude automātiski izmantos rīku `get_stocklevels` un atgriezīs datus no Directo.

Žurnālfailu pārbaude, ja kaut kas nedarbojas:

Žurnālfaila atrašanās vietai ir tās pašas divas iespējas, kas solī 4.1:

```
# Tiešais .exe instalētājs:
Get-Content "$env:APPDATA\Claude\logs\mcp-server-directo.log" -Tail 30

# Microsoft Store / MSIX:
```

```
Get-Content
"$env:LOCALAPPDATA\Packages\Claude_pzs8sxrjxfjjc\LocalCache\Roaming\Claude\l
ogs\mcp-server-directo.log" -Tail 30
```

Oficiālā Claude Desktop MCP dokumentācija: modelcontextprotocol.io — Connect to local MCP servers

5. Gemini CLI uzstādīšana



Vispirms pabeidz 3. nodaļu (solis 3.1 + A vai B ceļš). Gemini CLI uzstādīšana paredz, ka MCP serveris jau eksistē.



MCP serveris darbojas tikai **tavā paša datorā** un tikai terminālī bāzētajā **Gemini CLI** lietotnē — nevis Gemini tīmekļa versijā un ne citu lietotāju datoros.



⚠ Pirms turpināšanas pārliecinies, ka tev ir derīga autentifikācija — vai nu Google konts (Gemini Code Assist), bezmaksas Gemini API atslēga (Flash modeļi), vai apmaksāts abonements. Ierobežojumi un modeļu izvēle ir atkarīgi no autentifikācijas veida (skat. soli 2.3).



Ieteicams: ja A ceļā (3. nodaļa) palaidi skriptu `install-gemini-mcp.ps1`, soļi 5.1 un 5.3—5.4 jau ir izdarīti. Jāizpilda tikai solis 5.2 (Google API atslēga — to skripts nedara, jo tā ir atsevišķa Google, nevis Directo autentifikācija) un soļi 5.5—5.6.

Solis 5.1 — Gemini CLI instalēšana



Šim solim nav vajadzīgas administratora tiesības — ar nosacījumu, ka npm globālās instalācijas atrašanās vieta norāda uz tavu paša profilu. Ja tas vēl nav konfigurēts, izdari to vienu reizi (parastā PowerShell, ne kā administrators):

```
mkdir "$env:APPDATA\npm-global" -Force
npm config set prefix "$env:APPDATA\npm-global"
[Environment]::SetEnvironmentVariable("Path", $env:Path +
";$env:APPDATA\npm-global", "User")
```

Aizver PowerShell pilnībā un atver jaunu logu (PATH atjaunojas tikai jaunā sesijā), pēc tam instalē Gemini CLI:

```
npm install -g @google/gemini-cli
```

Pārbaudi:

```
gemini --version
```



⚠ Ja komanda joprojām prasa administratora tiesības, visticamākais iemesls ir tas, ka npm prefikss jau bija iestatīts (piemēram, ar citu programmatūru) uz `C:\Program Files\…`. Pārbaudi pašreizējo iestatījumu ar `npm config get prefix` — pēc iepriekšējā soļa tam jārāda `%APPDATA%\npm-global`, nevis Program Files mape.

Solis 5.2 — Google API atslēgas uzstādīšana

Ej uz https://aistudio.google.com/apikey un izveido API atslēgu.

Saglabā atslēgu kā pastāvīgu vides mainīgo:

```
[System.Environment]::SetEnvironmentVariable("GEMINI_API_KEY",
"YOUR_GOOGLE_API_KEY", "User")
```

Aizver PowerShell un atver to no jauna. Pārbaudi:

```
$env:GEMINI_API_KEY
```

Solis 5.3 — Konfigurācijas faila atrašanās vieta

Gemini CLI konfigurācija Windows vidē atrodas šeit:

```
$env:USERPROFILE\.gemini\settings.json
```

Pārbaudi, vai mape eksistē:

```
Test-Path "$env:USERPROFILE\.gemini"
```

Ja atbilde ir `False`, izveido mapi:

```
New-Item -ItemType Directory -Force -Path "$env:USERPROFILE\.gemini"
```

Solis 5.4 — MCP servera pievienošana

Izveido vai labo failu `$env:USERPROFILE\.gemini\settings.json`:

```
{
  "mcpServers": {
    "directo": {
      "command": "node",
      "args": [
        "C:\\Users\\YOURNAME\\directo-mcp\\directo-mcp.js"
```

```
      ],
      "env": {
        "DIRECTO_API_KEY": "$DIRECTO_API_KEY"
      }
    }
  }
}
```



Aizstāj `YOURNAME` ar savas faktiskās profila mapes nosaukumu — precīzo vērtību pārbaudi ar `$env:USERPROFILE` (skat. brīdinājumu par to pašu tēmu solī 4.2).



Piezīme par API atslēgas loģiku: `directo-mcp.js` vispirms nolasa atslēgu no Credential Manager (A ceļa skripts vai solis B7) — tas Gemini gadījumā darbojas tieši tāpat kā Claude Desktop gadījumā, jo abi palaiž vienu un to pašu servera failu. Šeit redzamais env bloks ir rezerves variants, nevis vienīgais ceļš — ja Credential Manager jau ir uzstādīts, šis bloks nav obligāti nepieciešams, lai gan netraucē. Ja tomēr vēlies to izmantot, saglabā vērtību kā vides mainīgo:

```
[System.Environment]::SetEnvironmentVariable("DIRECTO_API_KEY",
"YOUR_DIRECTO_API_KEY", "User")
```

Gemini CLI atbalsta vides mainīgo izvēršanu env vērtībās ar $VAR_NAME sintaksi — tāpēc JSON failā ir "$DIRECTO_API_KEY", nevis pati atslēga.

Solis 5.5 — Mapes uzticēšana (Trusted Folders)

Gemini CLI ir drošības funkcija Trusted Folders, kas pēc noklusējuma neļauj pievienot MCP serverus no neuzticamas mapes. Ja šī funkcija ir ieslēgta, mapei jābūt atzīmētai kā uzticamai.

Vienkāršākais veids — uzticēties serverim tieši `settings.json` failā, pievienojot `"trust": true` `directo` blokam:

```
{
  "mcpServers": {
    "directo": {
      "command": "node",
      "args": [
        "C:\\Users\\YOURNAME\\directo-mcp\\directo-mcp.js"
      ],
      "env": {
        "DIRECTO_API_KEY": "$DIRECTO_API_KEY"
      },
      "trust": true
    }
  }
}
```



`"trust": true` izlaiž rīku izsaukumu apstiprinājuma dialogus. Izmanto to tikai tiem serveriem, kurus pilnībā pārvaldi pats — šis Directo serveris ir tieši tāds gadījums.

Alternatīva — ja Trusted Folders ir ieslēgts, Gemini CLI, pirmoreiz atverot šo mapi, parāda uzticēšanās dialogu. Tur izvēlies **Trust folder**. Izvēle tiek saglabāta failā `~/.gemini/trustedFolders.json` un tiek prasīta tikai vienu reizi katrai mapei. Lai vēlāk mainītu uzticēšanās līmeni, CLI iekšienē izmanto komandu `/permissions`.

Solis 5.6 — Pārbaude

Palaid Gemini CLI:

```
gemini
```

Pārbaudi MCP savienojumu:

```
/mcp list
```

Jāparāda:

```
 directo - Ready (32 tools)
```

Testa vaicājums:

> Parādi visus noliktavas atlikumus

Oficiālā Gemini CLI MCP dokumentācija: geminicli.com/docs/tools/mcp-server

6. Problēmu novēršana

Biežākās kļūdas (abiem klientiem)

Kļūdas ziņojums	Cēlonis	Risinājums
`DIRECTO_API_KEY` puudub *(burtisks žurnālfaila teksts, joprojām igauniski — skat. piezīmi zemāk)*	Neizdevās saglabāšana Credential Manager	Atkārto soli B7 (vai A ceļš: `.\set-directo-apikey.ps1`)
`Cannot find module`	Neizdevās `npm install` VAI `directo-mcp.js` ir nepareizā mapē / ar nepareizu nosaukumu	Atkārto soli B6; pārbaudi `Test-Path "$env:USERPROFILE\directo-mcp\directo-mcp.js"`
`SyntaxError`	Fails `directo-mcp.js` ir bojāts vai nepilnīgs	Lejupielādē failu vēlreiz, skat. soli B4 (vai A ceļš: A1)

Kļūdas ziņojums	Cēlonis	Risinājums
`Incorrect filter id`	Nederīgs API filtra nosaukums	Zināms ierobežojums — izmanto bez šī filtra
`401` / `403`	Directo API atslēga ir nederīga vai beigusies	Pārbaudi atslēgu Directo uzstādījumos
`running scripts is disabled`	PowerShell izpildes politika	`Set-ExecutionPolicy RemoteSigned -Scope CurrentUser` (solis A1 vai B1)
`is not digitally signed` / `UnauthorizedAccess`	Windows Mark-of-the-Web (fails lejupielādēts caur pārlūku)	`Get-ChildItem <mape> -Recurse | Unblock-File` (skat. soli A1)
`npm not recognized`	Node.js nav PATH mainīgajā	Aizver un atver PowerShell no jauna



Pirmā rinda tabulā („DIRECTO_API_KEY puudub”) rāda tieši to, kas parādīsies žurnālfailā — tas ir burtisks ziņojums, ko izvada `directo-mcp.js` un kas joprojām ir igauniski, kamēr nav pieejama šī faila angļu valodas versija. Ja redzi šo tekstu žurnālfailā, tas nozīmē „trūkst DIRECTO_API_KEY”, un risinājums ir tas pats, kas norādīts tabulā.

Claude Desktop specifiskās kļūdas

Problēma	Risinājums
Nav redzama āmura ikona	Pārbaudi žurnālfailu (skat. zemāk)
MCP serveris nestartē	Pārbaudi, vai konfigurācijas faila nosaukumam nav `.txt` paplašinājuma
Trūkst konfigurācijas mapes	Palaid Claude Desktop vienu reizi — mapes tiek izveidotas automātiski
Konfigurācija labota, bet nekas nemainās	Pārliecinies, ka labo PAREIZO failu — skat. soli 4.1 (divas iespējamās vietas, MSIX pret tiešo instalāciju)

Žurnālfails (izmēģini abus, skat. soli 4.1):

```
Get-Content "$env:APPDATA\Claude\logs\mcp-server-directo.log" -Tail 30
Get-Content "$env:LOCALAPPDATA\Packages\Claude_pzs8sxrjxfjjc\LocalCache\Roaming\Claude\logs\mcp-server-directo.log" -Tail 30
```

Gemini CLI specifiskās kļūdas

Problēma	Risinājums
`/mcp list` rāda `Disconnected`	Pievieno servera konfigurācijai `”trust”: true` (solis 5.4) vai uztici mapi ar dialoga / komandas `/permissions` palīdzību
`GEMINI_API_KEY` kļūda	Atkārto soli 5.2, atver jaunu PowerShell logu
Rate limit / `429` kļūda	Pārsniegts bezmaksas līmeņa pieprasījumu limits — pagaidi limita atjaunošanos vai apsver apmaksātu abonementu (skat. soli 2.3)

Gemini CLI diagnostika:

```
/mcp list
```

7. MCP rīki

Abi klienti (Claude Desktop un Gemini CLI) izmanto vienus un tos pašus rīkus:

Grupa	**Rīki**
Pārdošana	`get_invoices`, `get_orders`, `get_sales_quotations`, `get_sales_contracts`, `get_receipts`
Klienti	`get_customers`, `get_suppliers`, `get_contacts`
Artikuli / noliktava	`get_items`, `get_itemclasses`, `get_stocklevels`, `get_stocklevels_sn`, `get_allocations`, `get_priceformulas`
Iepirkumi	`get_purchase_orders`, `get_purchase_invoices`, `get_purchase_payments`, `get_stock_receipts`
Grāmatvedība	`get_transactions`, `get_accounts`, `get_financial_budgets`, `get_nettings`
Loģistika	`get_deliveries`, `get_movements`
Citi	`get_projects`, `get_objects`, `get_resources`, `get_events`, `get_users`, `get_recipes`, `get_replacementlogs`
Sinhronizācija	`get_deleted`

8. Drošība

Pasākums	**Skaidrojums**
Credential Manager (abiem klientiem)	API atslēga ir šifrēta un redzama tikai tavam Windows kontam — Claude un Gemini nolasa vienu un to pašu ierakstu
Vides mainīgais (Gemini rezerves variants)	Ja Credential Manager ir uzstādīts, env bloks nav obligāti nepieciešams — skat. piezīmi solī 5.4
Lokāls serveris	MCP serveris darbojas tikai tavā datorā un nav publiski sasniedzams
Minimālas tiesības	Ierobežo Directo API atslēgas tiesības tikai uz tiem resursiem, kas tev patiešām vajadzīgi
OneDrive/Dropbox	Izslēdz `directo-mcp` mapi no mākoņa sinhronizācijas
Administratora tiesības	Tikai Node.js instalēšanai (solis 3.1) — kopīgs priekšnosacījums abiem ceļiem — viss pārējais, tostarp CredentialManager modulis, darbojas ar `-Scope CurrentUser` bez administratora tiesībām (skat. 2.4)

9. Atsauces

Claude Desktop:

- Lejupielādēt Claude Desktop
- Claude Desktop MCP uzstādīšana
- Pamācība: lokālie MCP serveri Claude Desktop vidē
- Oficiālā MCP dokumentācija

Gemini CLI:

- Gemini CLI dokumentācija
- Gemini CLI MCP uzstādīšana
- Google AI Studio API atslēgas
- Gemini Code Assist dokumentācija

Directo:

- Directo API dokumentācija

Vispārīgi:

- Lejupielādēt Node.js
- CredentialManager PowerShell modulis

Ātrās instalācijas skripti (skat. A ceļu, 3. nodaļa):

- `directo-mcp.js` — MCP serveris (32 rīki, pārbaudīts pret shēmām)
- `set-directo-apikey.ps1` — API atslēgas pārvaldība (sākotnējā uzstādīšana + nomaiņa)
- `install-claude-mcp.ps1` — automātiska Claude Desktop uzstādīšana
- `install-gemini-mcp.ps1` — automātiska Gemini CLI uzstādīšana

From:
https://wiki.directo.ee/ - **Directo Help**

Permanent link:
https://wiki.directo.ee/lv/ask_directo?rev=1790062443

Last update: **2026/09/22 10:34**