

# SharePointClient — Teknisk not

## Problemet

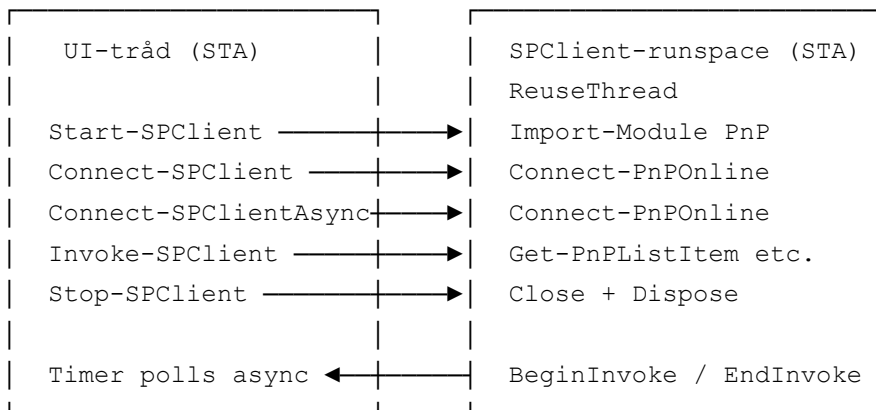
PnP.PowerShell lagrar anslutningsstate (token, session) per **runspace**. I den gamla koden använde knapp-handler en `BackgroundWorker` som kör `DoWork` på en `ThreadPool`-tråd. PowerShell 5.1 + WinForms skapar i det fallet en **ny implicit runspace** för varje `BackgroundWorker`-anrop. Resultatet:

1. `Connect-PnPOnline` i `DoWork` → anslutningen sparas i **bakgrunds-runspace**
2. `Get-PnPListItem` vid rapportbygge → körs i **huvud-runspace** → ser ingen anslutning
3. `$e.Error.Exception.ToString()` → `$e.Error` ÄR `Exception` (inte wrapper) → krasch

## Lösningen: SharePointClient.ps1

En ny modul som skapar **en enda dedikerad runspace** med `ReuseThread`. Alla PnP-anrop (import, connect, get-list) körs via denna runspace.

## Arkitektur



## Publika funktioner

Funktion	Beskrivning	Blockerar?
	Skapar runspace, NuGet bootstrap, importerar	

Start-SPClient	PnP	Ja (kort)
Connect-SPClient	Kör Connect-PnPOnline i runspace	Ja
Connect-SPClientAsync	Startar Connect-PnPOnline med BeginInvoke	Nej
Poll-SPClientAsync	Kollar om async-anropet är klart, uppdaterar state	Nej
Invoke-SPClient	Kör godtycklig ScriptBlock i runspace	Ja
Get-SPClientStatus	Returnerar Connected/Connecting/Error/RunspaceOk	Nej
Stop-SPClient	Stänger och städar runspace	Nej

## Flöden

### AutoConnect (startup under splash):

```
Start-SPClient → Connect-SPClient → $global:SpConnected = $true
```

UI fryser inte — splash visas redan.

### Manuell anslutning (knapp-klick):

```
Klick → Start-SPClient (synkront, ~1s)
      → Connect-SPClientAsync (startar BeginInvoke)
      → Timer pollar var 200 ms
      → Poll-SPClientAsync → klart! → uppdatera knapp + state
```

UI fryser INTE under Connect-PnPOnline (~10-15 sek). Splash visas.

### Rapportbygge:

```
Get-SPClientStatus → Connected?
      → Invoke-SPClient { Get-PnPListItem ... } → extraherar fält → returnerar hashta
```

ListItem-objekt stannar i runspace. Bara enkla hashtables/strings korsar runspace-gränsen.

### Varför inte Get-PnPListItem direkt?

PnP ListItem -objekt implementerar en FieldValues -indexer ( \$item['field'] ). När

objekt korsar runspace-gränser via `PowerShell.Invoke()` serialiseras de till `PSObject` — indexern försvinner. Därför körs hela frågan + fält-extraktion inuti `Invoke-SPClient`, och bara en enkel `@{ Ok=$true; Found=$true; Data=@{field='value'} }` returneras till UI-tråden.

## Global state

Variabel	Sätts av	Läses av
<code>\$global:SpConnected</code>	SharePointClient	Main.ps1 (knapp, build)
<code>\$global:SpConnecting</code>	SharePointClient	Main.ps1 (knapp guard)
<code>\$global:SpError</code>	SharePointClient	Main.ps1 (logg)
<code>\$global:SpEnabled</code>	Main.ps1 (config)	Main.ps1
<code>\$global:SpAutoConnect</code>	Main.ps1 (config)	Main.ps1

## Ändrade filer

Fil	Ändring
<b>Modules/SharePointClient.ps1</b>	NY — hela modulen
<b>Main.ps1 rad 32</b>	Import av SharePointClient.ps1
<b>Main.ps1 rad 82–108</b>	Startup: ersatt direkt PnP med Start-SPClient + Connect-SPClient
<b>Main.ps1 rad 519–580</b>	Knapp: ersatt BackgroundWorker med SPClientAsync + Timer
<b>Main.ps1 rad 3366–3370</b>	Rapportbygge: ersatt Get-PnPConnection med Get-SPClientStatus
<b>Main.ps1 rad 3442–3526</b>	Rapportbygge: ersatt Get-PnPListItem med Invoke-SPClient
<b>Main.ps1 rad 3958</b>	Cleanup: Stop-SPClient vid avslut