Betriebsdokumentation

# Funktionen des Skripts

Mein Code ruft die Temperatur von einer Wetter-API ab, nimmt die Informationen aus der Abfrage und zeigt sie in einer Benachrichtigung auf dem Computer an.

# Grobbeschreibung

Mein Code überprüft zunächst, ob es einen bestimmten (im code bin es ich) Benutzer gibt. Danach prüft er, ob eine Internetverbindung mit Google besteht. Wenn dies gelingt, wird ein API-Aufruf durchgeführt, der mir die Temperatur aus einer JSON-Datei liefert. Diese Temperatur wird dann als Benachrichtigung angezeigt. Falls dies jedoch 10-mal fehlschlägt, wird eine Benachrichtigung angezeigt, dass keine Verbindung besteht. Im Falle eines API-Fehlers wird eine Fehlermeldung ausgegeben.

# Zielsystem

Ein Bild, das Text, Screenshot, Software, Display enthält.

Automatisch generierte BeschreibungDas Programm wurde für meinen Privatgebrauch erstellt. Das PowerShell-Programm wurde in Version 5.1.22621.3672 erstellt. Das Programm läuft auf Windows 11 Pro in der Version 23H2.

# Auslöser für das Skript

Für das Ausführen des Programms benötigt man den Task Scheduler. Um den Task Scheduler zu öffnen, drücken Sie «Win + R» und geben Sie *taskschd.msc* ein. Nach dem Öffnen erstellen Sie eine neue Aufgabe. Geben Sie ihr einen Namen und stellen Sie die Optionen wie in den Bildern gezeigt ein.

Ein Bild, das Text, Screenshot, Display, Software enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Software, Computersymbol enthält.

Automatisch generierte Beschreibung-NoProfile -ExecutionPolicy bypass File""/Speicherort\_des Proframmes/..ps1



## Benötigte Berechtigungen

Für das Ausführen des Programmes ist es wichtig, dass man **BurntToast** installiert, um die Benachrichtigungen zu bekommen. Am besten macht man das in dem   
Install-Module -Name BurntToast mit Adminrechte installiert.

Das Ausführen des Programmes braucht keine Adminrechte und ist dadurch nach der relativ ungefährlich für den Benutzer

## Und das Skript an sich

<https://github.com/ascii-phoenix/Powershell-wetter-api/blob/main/Powershell-weather-api.ps1>

<#

.SYNOPSIS

    My code Gets the temperatur from a api and displays it with a notifacation

.DESCRIPTION

    My code checks if there is a specific user.

    After that, it checks if there is an internet connection with Google.

    If it succeeds, it makes an API call that gives me the temperature from a JSON file, which I display as a notification.

    If it fails 10 times, it gives a notification that there is no connection. If the API fails, there will be an error message.

.NOTES

Author: Eloi Knecht

    Build-version: 1.2

    Last updated : 2023-06-24

    Download BurntToast

    Header template was copied from scriptrunner.com

    Build in PS1-version 5.1.22621.3672

.COMPONENT

   Please downlode BurntToast and change the Usersting in the 42 line to yours.

.LINK

https://github.com/Windos/BurntToast.

https://open-meteo.com/

https://www.scriptrunner.com/en/blog/powershell-script-header-parameters-in-scriptrunner/

.Parameter user

 is the user

.Parameter connectedAdapter

 is a bool that checks that tere is a conection

.Parameter counterOfRepetition

Counts how many trys there where to connect with google

.Parameter apiUrl

 is a string of my API-URL

.Parameter response

 it is the API respons as a json file

.Parameter temperature\_2m

 is the temperture as a numbre

.Parameter imagePath

 is the image used in my nodification

#>

Import-Module BurntToast # (Install-Module -Name BurntToast) do not forget to downlode "BurntToast".

do {# Checks who is the user

    $user = (Get-WmiObject -Class Win32\_ComputerSystem).UserName

    if ($user -ne "") {

        if (-not($user -eq "Studio2Eloi\eloik")) {exit}

    } else {

        Start-Sleep -Seconds 5

    }

} while ($null -eq $user)

$counterOfRepetition = 0;

do { #Checks if there is a connection to google.com (only 10 trys)

    $connectedAdapter = Test-Connection -ComputerName google.com -Count 1 -Quiet

    $counterOfRepetition++

    if (!$connectedAdapter) {

        Start-Sleep -Seconds 5

    }

} while (!$connectedAdapter -and ($counterOfRepetition -lt 10))

if ($connectedAdapter -and $counterOfRepetition -lt 10) {

    $apiUrl = "https://api.open-meteo.com/v1/forecast?latitude=47.3925&longitude=8.0442&current=temperature\_2m,rain&forecast\_days=1"

# try was created with Chat GPT to prevent that $response is not $null and that the programm does not crash.

    try {# Try to make an api-call

        $response = Invoke-RestMethod -Uri $apiUrl -Method Get

        if ( $null -ne $response) {# cheks if there is an answer

            $temperature\_2m = $response.current.temperature\_2m

            $imagePath = "C:\Users\eloik\Powershell\sun-8553511\_1280.jpg"

            New-BurntToastNotification -Text "Heute ist es : $($temperature\_2m)C" -AppLogo $imagePath

        } else {

            $imagePath = "C:\Users\eloik\Powershell\sun-8553511\_1280.jpg"

            New-BurntToastNotification -Text "Leider keine API Verbindung" -AppLogo $imagePath

        }

    } catch {# if there is a Error

        $imagePath = "C:\Users\eloik\Powershell\sun-8553511\_1280.jpg"

        New-BurntToastNotification -Text "Leider gabe es ein Problem" -AppLogo $imagePath

    }

} else {

    $imagePath = "C:\Users\eloik\Powershell\sun-8553511\_1280.jpg"

    New-BurntToastNotification -Text "Leider keine API Verbindung" -AppLogo $imagePath

}

## Verweis auf die Entwicklerdokumentation.

Falls Sie gerne mehr darüber wissen möchten, wie der PowerShell-Code funktioniert, ist es empfehlenswert, die Entwicklerdokumentation zu konsultieren, um mehr Informationen zu erhalten. Die Entwicklerdokumentation finden Sie im folgenden Link in einem GitHub-Repository: