Our Blog

← Hyper-V – Can't start Virtual Machine because it's backing up |
PowerShell – Returning one object from a function as an Array →

Meet Statusimo – PowerShell generated Status Page

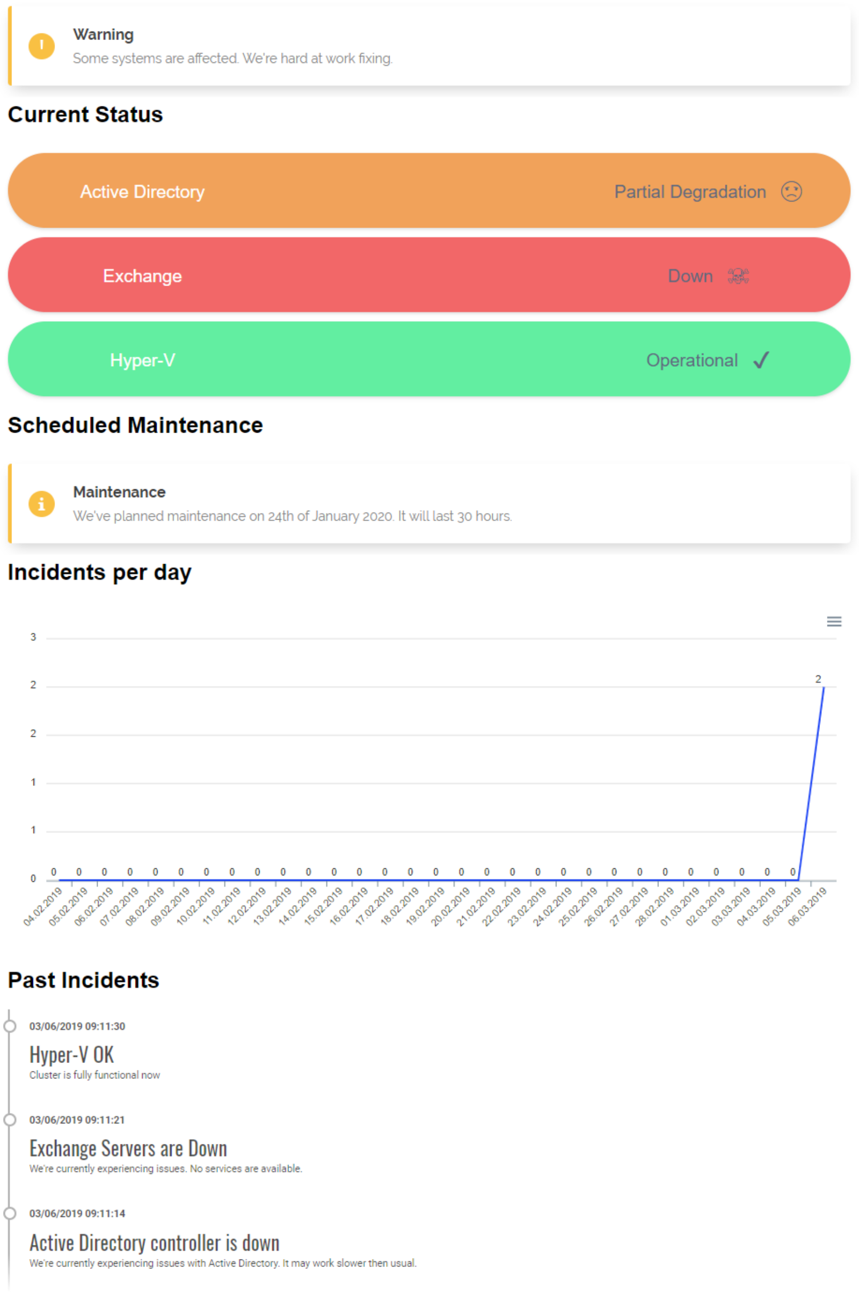
6th March 2019 *Standard* *0* *Przemyslaw Klys*

A few weeks ago, **Mateusz Czerniawski**, mentioned that he wants to build a **Status Page** for his company services. While I haven't needed for myself, it seems like an excellent idea to try and create one in **PowerShell**. Since I've been working on **PSWriteHTML** for a while, it wasn't that far fetched idea. While **PSWriteHTML** has a long way to go, to be in a state I want it to be, after a few days I had a **prototype** that didn't require much work to generate. If you're wondering what Status Page is it's a little summary page for your users to check what is the status of services they use. It has been popular in the last few years and is offered by many services (**Twitter**, **GitHub**, **Office 365** – they all have it). Companies are selling it **as a service** as well where you can host your status page for your users. This one is free.

Statusimo - What you get?

While I am reasonably sure that you understood the idea behind it, it's best to show you what you get before going into details on how to get there. What you see below is Statusimo Page. It's a **single HTML** file that you can host however you like.

> *What you hear, you forget; what you see, you remember; what you do, you understand.*



Warning
Some systems are affected. We're hard at work fixing.

Current Status

Active Directory — Partial Degradation ☹

Exchange — Down

Hyper-V — Operational ✓

Scheduled Maintenance

Maintenance
We've planned maintenance on 24th of January 2020. It will last 30 hours.

Incidents per day

Past Incidents

03/06/2019 09:11:30
Hyper-V OK
Cluster is fully functional now

03/06/2019 09:11:21
Exchange Servers are Down
We're currently experiencing issues. No services are available.

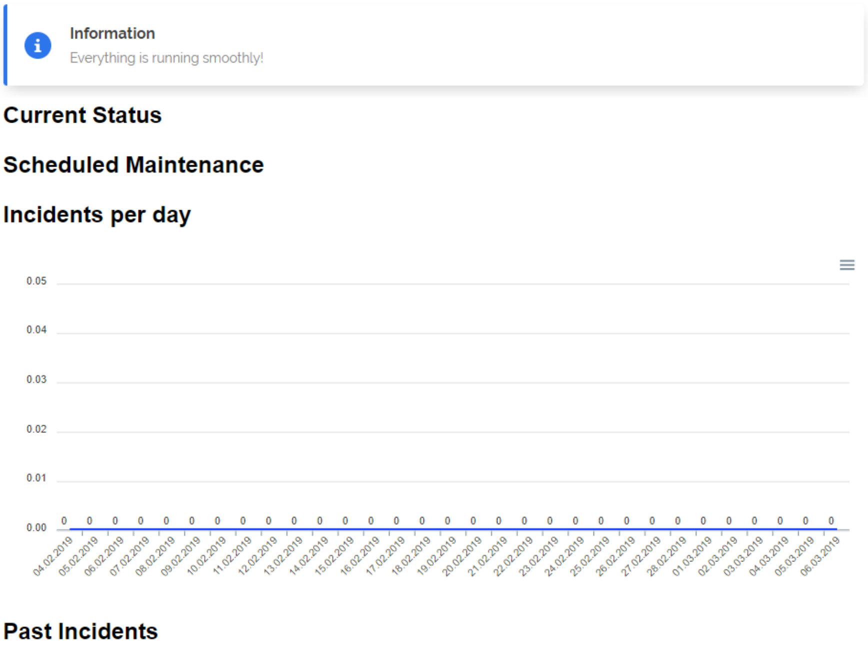
03/06/2019 09:11:14
Active Directory controller is down
We're currently experiencing issues with Active Directory. It may work slower then usual.

Looks nice? Are you interested? No? Oh, well, feel free to explore my **other PowerShell modules** which may be more to your likening. For those that things above look interesting, let's continue, shall we? All the above visual part is generated by a single PowerShell function.

```
Import-Module Statusimo # it may not be necessary if you've it installed on a system

```

```
New-StatusimoPage -FilePath
$PSScriptRoot\StatusPage.html -IncidentsPath
$PSScriptRoot\Incident -MaintenancePath
$PSScriptRoot\Maintenance
```

Function **New-StatusimoPage** takes three parameters. One is a path to where to save generated **HTML** file, and the other two are where all the incidents and maintenance are stored. You see, the way it's designed is that generation of the page is based on incidents. Each event is a simple JSON structured file. This function scanned the folder with all events and based on that data generates output. If there is no data or path is wrong output would look like below.



Information

Everything is running smoothly!

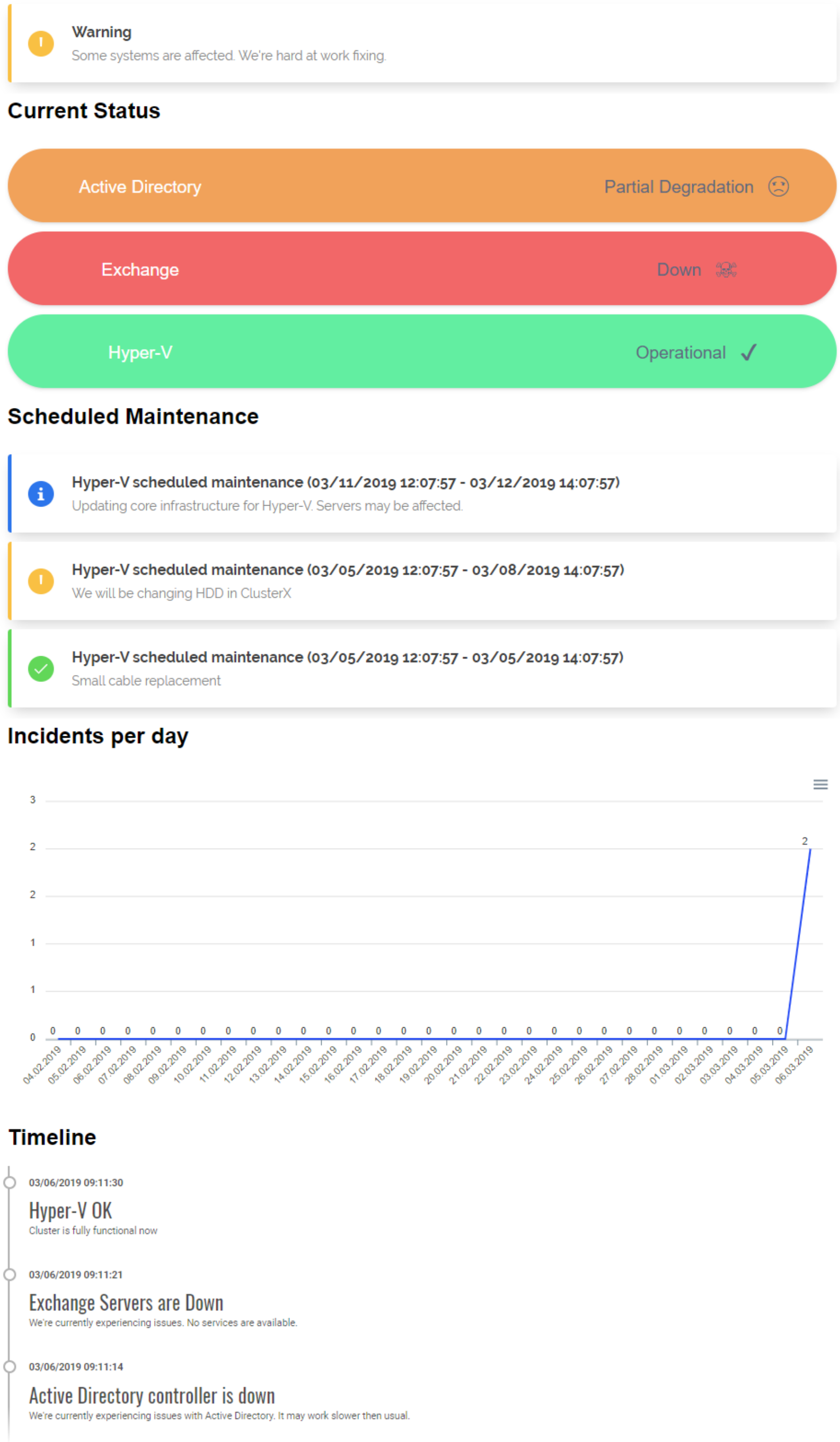
Current Status

Scheduled Maintenance

Incidents per day

Past Incidents

It's not pretty, but this is just a starter when there's nothing in it. There are no files in the Incident or Maintenance folder, so there's no way for this module to generate any data. Let's add some then.



Warning
Some systems are affected. We're hard at work fixing.

Current Status

Active Directory — Partial Degradation ☹

Exchange — Down

Hyper-V — Operational ✓

Scheduled Maintenance

Hyper-V scheduled maintenance (03/11/2019 12:07:57 - 03/12/2019 14:07:57)
Updating core infrastructure for Hyper-V. Servers may be affected.

Hyper-V scheduled maintenance (03/05/2019 12:07:57 - 03/08/2019 14:07:57)
We will be changing HDD in ClusterX

Hyper-V scheduled maintenance (03/05/2019 12:07:57 - 03/05/2019 14:07:57)
Small cable replacement

Incidents per day

Timeline

03/06/2019 09:11:30
Hyper-V OK
Cluster is fully functional now

03/06/2019 09:11:21
Exchange Servers are Down
We're currently experiencing issues. No services are available.

03/06/2019 09:11:14
Active Directory controller is down
We're currently experiencing issues with Active Directory. It may work slower then usual.

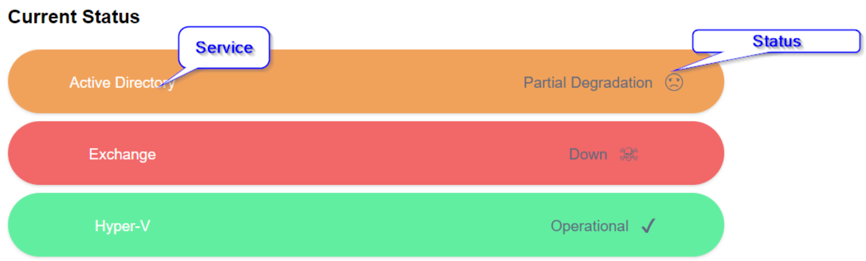
Statusimo - Adding incidents / events

There are two ways to add incidents. Create JSON file in Incidents folder or use PowerShell command.

```powershell
Import-Module Statusimo

$newStatusimoIncidentSplat = @{
    Date       = (Get-Date)
    Overview   = "We're currently experiencing issues. No services are available."
    Title      = 'Exchange Servers are Down'
    Status     = 'Down'
    Service    = 'Exchange'
    FolderPath = "$PSScriptRoot\Incidents"
}
New-StatusimoEvent @newStatusimoIncidentSplat

$newStatusimoIncidentSplat = @{
    Date = (Get-Date)
    Overview = "We're currently experiencing issues with Active Directory. It may work slower then usual."
    Title = 'Active Directory controller is down'
    Status = 'Partial Degradation'
    Service = 'Active Directory'
    FolderPath = "$PSScriptRoot\Incidents"
}
New-StatusimoEvent @newStatusimoIncidentSplat

New-StatusimoEvent -FolderPath $PSScriptRoot\Incidents -Date (Get-Date) -Service 'Hyper-V' -Status 'Operational' -Title 'Hyper-V OK' -Overview "Cluster is fully functional now"
```

As you can notice above there are three events created. You have five important parameters. **Date**, **Overview**, **Title**, **Status,** and **Service**. Of course, there's the **6th** parameter where to store the event but that's not that important at the moment.

- **ServiceName** – What's important to know is that when you write a service name, it can be anything you want (with reasonable limits). If it's the first event with **Service Name**, it will create a new status for it under the **Current Status** section. If it's second or later event, it will just update the status. If you will make a typo and have Active Directory and ActiveDirectory in two different events it will create two services. Being consistent is important.



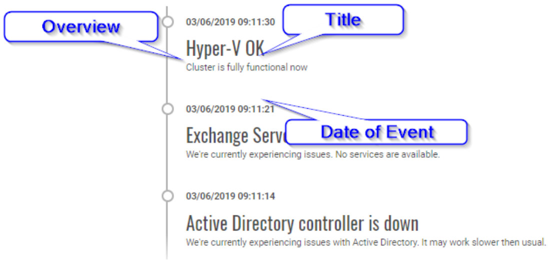
Current Status

Active Directory — Partial Degradation ☹

Exchange — Down ☠

Hyper-V — Operational ✓

- Status – There are three statuses **Down (color Red)**, **Partial Degradation (color Orange)** and **Operational (color Green)**. When the system generates HTML it takes the newest event and checks status of it and changes Current status accordingly.
- Title – Title is displayed in incidents status



03/06/2019 09:11:30

Hyper-V OK

Cluster is fully functional now

03/06/2019 09:11:21

Exchange Serv

We're currently experiencing issues. No services are available.

03/06/2019 09:11:14

Active Directory controller is down

We're currently experiencing issues with Active Directory. It may work slower then usual.

- ✓ Overview – this section is a longer description of what is happening.
- ✓ Date – this is a date for the event, incident to happen. Sorting is applied based on it.

When you execute **PowerShell** above each entry creates a single **JSON** file with a structure like below

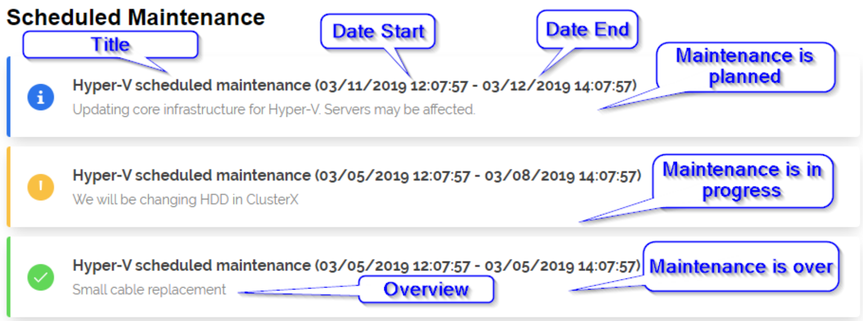
```
{
    "Date":  "\/Date(1551889816096)\/",
    "Service":  "Exchange",
    "Status":  "Down",
    "Title":  "Exchange Servers are Down",
    "Overview":  "We\u0027re currently experiencing issues. No services are available."
}
```

It means you're free to create those JSON files yourself via any means, or external applications and as long as you will keep it's the format you can feed data directly into **Statusimo**.

Statusimo - Adding maintenance

There are two ways to create maintenance. Add specially crafted JSON file (it's simple enough) generated by some of your systems, by your Service Desk agent or run this little **PowerShell** command. Below code adds 3 types of maintenance types. One that is planned, one that is already in progress and one that is already done. It's strictly based on **DateStart** and **DateEnd** parameters. There's no need to modify maintenance when there is time to start maintenance.

```
Import-Module Statusimo

$newStatusimoMaintenanceSplat = @{
    Title = 'Hyper-V scheduled maintenance'
    DateStart = (Get-Date).AddDays(5)
    DateEnd = (Get-Date).AddDays(6).AddHours(2)
    Service = 'Hyper-V'
    Overview = "Updating core infrastructure for Hyper-V. Servers may be affected."
    FolderPath = "$PSScriptRoot\Maintenance"
}

New-StatusimoMaintenance @newStatusimoMaintenanceSplat


$newStatusimoMaintenanceSplat = @{
    Title = 'Hyper-V scheduled maintenance'
    DateStart = (Get-Date).AddDays(-1)
    DateEnd = (Get-Date).AddDays(2).AddHours(2)
    Service = 'Hyper-V'
    Overview = "We will be changing HDD in ClusterX"
    FolderPath = "$PSScriptRoot\Maintenance"
}

New-StatusimoMaintenance @newStatusimoMaintenanceSplat


$newStatusimoMaintenanceSplat = @{
    Title = 'Hyper-V scheduled maintenance'
    DateStart = (Get-Date).AddDays(-1)
    DateEnd = (Get-Date).AddDays(-1).AddHours(2)
    Service = 'Hyper-V'
    Overview = "Small cable replacement"
    FolderPath = "$PSScriptRoot\Maintenance"
}

New-StatusimoMaintenance @newStatusimoMaintenanceSplat
```



Scheduled Maintenance

Hyper-V scheduled maintenance (03/11/2019 12:07:57 - 03/12/2019 14:07:57)
Updating core infrastructure for Hyper-V. Servers may be affected.

Hyper-V scheduled maintenance (03/05/2019 12:07:57 - 03/08/2019 14:07:57)
We will be changing HDD in ClusterX

Hyper-V scheduled maintenance (03/05/2019 12:07:57 - 03/05/2019 14:07:57)
Small cable replacement

Maintenance is solely based on DateStart and DateEnd. Maintenance will switch itself when the time is right from planning to progress to over, but only upon page regeneration.

```
Import-Module Statusimo

New-StatusimoPage -FilePath $PSScriptRoot\StatusPage.html -IncidentsPath $PSScriptRoot\Incidents -MaintenancePath $PSScriptRoot\Maintenance
```

The idea here is that page regeneration should be **triggered automatically** so one doesn't have to do it manually. There are many ways to achieve it – **Task Scheduler**, **Folder monitoring**, and **others**.

Statusimo - Removing maintenance

There are two ways to remove maintenance. Simply delete files from **Maintenance** folder and generate a new page or run PowerShell command that will delete any maintenance works based on how old **DateEnd** will be.

```
Import-Module Statusimo

Remove-StatusimoMaintenance -DaysOld 0 -MaintenancePath $PSScriptRoot\Maintenance
New-StatusimoPage -FilePath $PSScriptRoot\StatusPage.html -IncidentsPath $PSScriptRoot\Incidents -MaintenancePath $PSScriptRoot\Maintenance
```

Basic module description and module page is located at **Statusimo – PowerShell Module**.

Tags: css, html, javascript, powershell, status page, statusimo

IT & OUTSOURCING

🏠 Evotec
ul. Francuska 96B/23
Katowice, 40-507
Poland

☎ +48 502 469 760

✉ contact@evotec.pl

English polski (Polish)