

icinga-meldung via Pushover

[/etc/icinga2/scripts/pushover-service.sh](#)

```
#!/bin/bash

#####
#####
#
# full dokumentation on https://pushover.net/api
#
# POST an HTTPS request to https://api.pushover.net/1/messages.json
with the following parameters:
# token (required) - your application's API token
# user (required) - the user/group key (not e-mail address) of your
user (or you),
#     viewable when logged into our dashboard (often referred to as
USER_KEY in our documentation and code examples)
# message (required) - your message
#
# Some optional parameters may be included:
# device - your user's device name to send the message directly to that
device,
#     rather than all of the user's devices (multiple devices may be
separated by a comma)
# title - your message's title, otherwise your app's name is used
# url - a supplementary URL to show with your message
# url_title - a title for your supplementary URL, otherwise just the
URL is shown
# priority - send as
#     -2 to generate no notification/alert,
#     -1 to always send as a quiet notification,
#     0 (default) to send notification with sound, vibration and
display(not in quiet hours)
#     1 to display as high-priority and bypass the user's quiet hours,
or
#     2 to also require confirmation from the user
#     retry
#     expire
# timestamp - a Unix timestamp of your message's date and time to
display to the user, rather than the time your message is received by
our API
# sound - the name of one of the sounds supported by device clients to
override the user's default sound choice
#
# That's it. Make sure your application is friendly to our API servers
and you're all set.
# For more information on each parameter, keep reading or jump to a
section at the left.
```

```
#
# Need help using our API or found an error in the documentation? Drop
# us a line.
#
#####
#####

logpath="/var/log/pushover_icinga.txt"
ICINGA2HOST="$(hostname)"
CURLPROXY=""
debug="1"

#####
#Übergebene Parameter
#
#PUSHOVERUSER = "$user.vars.pushover_user$"
#PUSHOVERTOKEN = "$user.vars.pushover_token$"
#PUSHOVERPRIORITY = "$user.vars.pushover_priority$"
#PUSHOVERDEVICE = "$user.vars.pushover_device$"
#PUSHOVERRETRY = "$user.vars.pushover_retry$"
#PUSHOVEREXPIRE = "$user.vars.pushover_expire$"
#
#NOTIFICATIONTYPE = "$notification.type$"
#HOSTDISPLAYNAME = "$host.display_name$"
#SERVICEDISPLAYNAME = "$service.display_name$"
#SERVICESTATE = "$service.state$"
#ICINGALONGDATETIME = "$icinga.long_date_time$"
#SERVICEOUTPUT = "$service.output$"
#SERVICENAME = "$service.name$"
#HOSTNAME = "$host.name$"
#####

PUSHOVERMESSAGE=$(cat << EOF
**** Service Monitoring on $ICINGA2HOST ****
$SERVICEDISPLAYNAME on $HOSTDISPLAYNAME is ${SERVICESTATE}!
Info:      $SERVICEOUTPUT
When:      $ICINGALONGDATETIME
Service:   $SERVICENAME
Host:      $HOSTNAME
EOF
)

PUSHOVERTITLE="$NOTIFICATIONTYPE - $HOSTDISPLAYNAME -
$SERVICEDISPLAYNAME is $SERVICESTATE"

#Wenn die Priorität 2 vergeben wurde, ist ein retry zwingend
erforderlich
#Sollte retry nicht gesetzt sein, wird er auf 30 gesetzt
if [ "$PUSHOVERPRIORITY" = "2" ] && [ "$PUSHOVERRETRY" = "" ]
then
```

```

    PUSHOVERRETRY="30"
fi

#Wenn die Priorität 2 vergeben wurde, ist ein expire zwingend
erforderlich
#Sollte expire nicht gesetzt sein, wird er auf 300 gesetzt
if [ "$PUSHOVERPRIORITY" = "2" ] && [ "$PUSHOVEREXPIRE" = "" ]
then
    PUSHOVEREXPIRE="300"
fi

#Kommando, um per curl die Pushover-message zu verschicken
failstate=$(curl \
    --silent \
    --insecure --proxy "$CURLPROXY" \
    --form-string "token=$PUSHOVERTOKEN" \
    --form-string "user=$PUSHOVERUSER" \
    --form-string "message=$PUSHOVERMESSAGE" \
    --form-string "title=$PUSHOVERTITLE" \
    --form-string "priority=$PUSHOVERPRIORITY" \
    --form-string "retry=$PUSHOVERRETRY" \
    --form-string "expire=$PUSHOVEREXPIRE" \
    --form-string "device=$PUSHOVERDEVICE" \
    --location https://api.pushover.net/1/messages.json)

#Wenn das debugging eingeschaltet ist, wird die folgende Meldung
ausgegeben
#$logpath sollte vorhanden sein und auf nagios:nagios gesetzt sein
if [ "$debug" = "1" ]
then
    cat << EOF >> "$logpath"
    #####
    Debugging-Tool
    #####
    DatumZeit:      $(date)
    PUSHOVERTOKEN:   $PUSHOVERTOKEN
    PUSHOVERUSER:    $PUSHOVERUSER
    PUSHOVERTITLE:   $PUSHOVERTITLE
    PUSHOVERDEVICE:  $PUSHOVERDEVICE
    PUSHOVERPRIORITY: $PUSHOVERPRIORITY
    PUSHOVERRETRY:   $PUSHOVERRETRY
    PUSHOVEREXPIRE:  $PUSHOVEREXPIRE
    NOTIFICATIONTYPE: $NOTIFICATIONTYPE
    HOSTDISPLAYNAME: $HOSTDISPLAYNAME
    SERVICEDISPLAYNAME: $SERVICEDISPLAYNAME
    SERVICESTATE:    $SERVICESTATE
    ICINGALONGDATETIME: $ICINGALONGDATETIME
    ICINGA2HOST:      $ICINGA2HOST
    SERVICEOUTPUT:   $SERVICEOUTPUT

```

```
SERVICENAME:      $SERVICENAME
pushover json output:  $failstate
```

```
EOF
fi
```

[/etc/icinga2/scripts/pushover-host.sh](#)

```
#!/bin/bash

#####
#####
#
# full dokumentation on https://pushover.net/api
#
# POST an HTTPS request to https://api.pushover.net/1/messages.json
with the following parameters:
# token (required) - your application's API token
# user (required) - the user/group key (not e-mail address) of your
user (or you),
# viewable when logged into our dashboard (often referred to as
USER_KEY in our documentation and code examples)
# message (required) - your message
#
# Some optional parameters may be included:
# device - your user's device name to send the message directly to that
device,
# rather than all of the user's devices (multiple devices may be
separated by a comma)
# title - your message's title, otherwise your app's name is used
# url - a supplementary URL to show with your message
# url_title - a title for your supplementary URL, otherwise just the
URL is shown
# priority - send as
# -2 to generate no notification/alert,
# -1 to always send as a quiet notification,
# 0 (default) to send notification with sound, vibration and
display(not in quiet hours)
# 1 to display as high-priority and bypass the user's quiet hours,
or
# 2 to also require confirmation from the user
# retry
# expire
# timestamp - a Unix timestamp of your message's date and time to
display to the user, rather than the time your message is received by
our API
# sound - the name of one of the sounds supported by device clients to
override the user's default sound choice
#
```

```

# That's it. Make sure your application is friendly to our API servers
and you're all set.
# For more information on each parameter, keep reading or jump to a
section at the left.
#
# Need help using our API or found an error in the documentation? Drop
us a line.
#
#####

logpath="/var/log/pushover_icinga.txt"
ICINGA2HOST="$(hostname)"
CURLPROXY=""
debug="1"

#####
#Übergebene Parameter
#
#PUSHOVERUSER = "$user.vars.pushover_user$"
#PUSHVERTOKEN = "$user.vars.pushover_token$"
#PUSHOVERPRIORITY = "$user.vars.pushover_priority$"
#PUSHOVERDEVICE = "$user.vars.pushover_device$"
#PUSHOVERRETRY = "$user.vars.pushover_retry$"
#PUSHOVEREXPIRE = "$user.vars.pushover_expire$"
#
#NOTIFICATIONTYPE = "$notification.type$"
#HOSTDISPLAYNAME = "$host.display_name$"
#ICINGALONGDATETIME = "$icinga.long_date_time$"
#HOSTNAME = "$host.name$"
#HOSTSTATE = "$host.state$"
#HOSTOUTPUT = "$host.output$"

#####

PUSHOVERMESSAGE=$(cat << EOF
**** Host Monitoring on $ICINGA2HOST ****
$HOSTDISPLAYNAME is $HOSTSTATE!
Info:      $HOSTOUTPUT
When:      $ICINGALONGDATETIME
Host:      $HOSTNAME
EOF
)

PUSHOVERTITLE="$NOTIFICATIONTYPE - Host $HOSTDISPLAYNAME is
$HOSTSTATE!"

#Wenn die Priorität 2 vergeben wurde, ist ein retry zwingend
erforderlich
#Sollte retry nicht gesetzt sein, wird er auf 30 gesetzt
if [ "$PUSHOVERPRIORITY" = "2" ] && [ "$PUSHOVERRETRY" = "" ]

```

```
then
    PUSHOVERRETRY="30"
fi

#Wenn die Priorität 2 vergeben wurde, ist ein expire zwingend erforderlich
#Sollte expire nicht gesetzt sein, wird er auf 300 gesetzt
if [ "$PUSHOVERPRIORITY" = "2" ] && [ "$PUSHOVEREXPIRE" = "" ]
then
    PUSHOVEREXPIRE="300"
fi

#Kommando, um per curl die Pushover-message zu verschicken
failstate=$(curl \
    --silent \
    --insecure --proxy "$CURLPROXY" \
    --form-string "token=$PUSHOVERTOKEN" \
    --form-string "user=$PUSHOVERUSER" \
    --form-string "message=$PUSHOVERMESSAGE" \
    --form-string "title=$PUSHOVERTITLE" \
    --form-string "priority=$PUSHOVERPRIORITY" \
    --form-string "retry=$PUSHOVERRETRY" \
    --form-string "expire=$PUSHOVEREXPIRE" \
    --form-string "device=$PUSHOVERDEVICE" \
    --location https://api.pushover.net/1/messages.json)

#Wenn das debugging eingeschaltet ist, wird die folgende Meldung ausgegeben
#$logpath sollte vorhanden sein und auf nagios:nagios gesetzt sein
if [ "$debug" = "1" ]
then
    cat << EOF >> "$logpath"
    #####
    Debugging-Tool
    #####
    DatumZeit:      $(date)
    PUSHOVERTOKEN:   $PUSHOVERTOKEN
    PUSHOVERUSER:    $PUSHOVERUSER
    PUSHOVERTITLE:   $PUSHOVERTITLE
    PUSHOVERDEVICE:  $PUSHOVERDEVICE
    PUSHOVERPRIORITY: $PUSHOVERPRIORITY
    PUSHOVERRETRY:   $PUSHOVERRETRY
    PUSHOVEREXPIRE:  $PUSHOVEREXPIRE
    HOSTDISPLAYNAME: $HOSTDISPLAYNAME
    ICINGALONGDATETIME: $ICINGALONGDATETIME
    NOTIFICATIONTYPE: $NOTIFICATIONTYPE
    ICINGA2HOST:      $ICINGA2HOST
    HOSTNAME:         $HOSTNAME
```

```
HOSTSTATE:      $HOSTSTATE
HOSTOUTPUT:     $HOSTOUTPUT
pushover json output:  $failstate
```

EOF

fi

[/etc/icinga2/conf.d/commands/pushover-notification.conf](#)

```
object NotificationCommand "pushover-host-notification" {
import "plugin-notification-command"

command = [ SysconfDir + "/icinga2/scripts/pushover-host.sh" ]

env = {
    PUSHOVERUSER = "$user.vars.pushover_user$"
    PUSHOVERTOKEN = "$user.vars.pushover_token$"
    PUSHOVERPRIORITY = "$user.vars.pushover_priority$"
    PUSHOVERDEVICE = "$user.vars.pushover_device$"
    PUSHOVERRETRY = "$user.vars.pushover_retry$"
    PUSHOVEREXPIRE = "$user.vars.pushover_expire$"

    NOTIFICATIONTYPE = "$notification.type$"
    HOSTDISPLAYNAME = "$host.display_name$"
    ICINGALONGDATETIME = "$icinga.long_date_time$"
    HOSTNAME = "$host.name$"
    HOSTSTATE = "$host.state$"
    HOSTOUTPUT = "$host.output$"

}
}

object NotificationCommand "pushover-service-notification" {
import "plugin-notification-command"

command = [ SysconfDir + "/icinga2/scripts/pushover-service.sh" ]

env = {
    PUSHOVERUSER = "$user.vars.pushover_user$"
    PUSHOVERTOKEN = "$user.vars.pushover_token$"
    PUSHOVERPRIORITY = "$user.vars.pushover_priority$"
    PUSHOVERDEVICE = "$user.vars.pushover_device$"
    PUSHOVERRETRY = "$user.vars.pushover_retry$"
    PUSHOVEREXPIRE = "$user.vars.pushover_expire$"

    NOTIFICATIONTYPE = "$notification.type$"
    HOSTDISPLAYNAME = "$host.display_name$"
    SERVICEDISPLAYNAME = "$service.display_name$"
    SERVICESTATE = "$service.state$"
    ICINGALONGDATETIME = "$icinga.long_date_time$"
```

```
SERVICEOUTPUT = "$service.output$"  
SERVICENAME = "$service.name$"  
HOSTNAME = "$host.name$"  
}  
}
```

[/etc/icinga2/conf.d/templates/pushover.conf](#)

```
template Notification "pushover-host-notification" {  
    command = "pushover-host-notification"  
  
    states = [ Up, Down ]  
    types = [ Problem, Acknowledgement, Recovery, Custom,  
             FlappingStart, FlappingEnd,  
             DowntimeStart, DowntimeEnd, DowntimeRemoved ]  
  
    period = "24x7"  
}  
  
template Notification "pushover-service-notification" {  
    command = "pushover-service-notification"  
  
    states = [ OK, Warning, Critical, Unknown ]  
    types = [ Problem, Acknowledgement, Recovery, Custom, FlappingStart,  
             FlappingEnd, DowntimeStart, DowntimeEnd, DowntimeRemoved ]  
  
    period = "24x7"  
}
```

[/etc/icinga2/conf.d/users/alle.conf](#)

```
object User "patrick" {  
    import "generic-user"  
  
    display_name = "Patrick"  
    # groups = [ "icingaadmins" , "icingaadmins-pushover" ]  
    groups = [ "icingaadmins-pushover" ]  
  
    email = "root@localhost"  
  
    #Pushover Messages  
    vars.pushover_user = "user-Token"  
    vars.pushover_token = "API-Token"  
    vars.pushover_device = [ "device" ]  
    # vars.pushover_priority = "2"  
    # retry >= 30  
    # vars.pushover_retry = "30"  
    # expire <= 10800
```



```
# vars.pushover_expire = "300"

}
```

Kurze Beschreibung:

- device:
 - Devices können einzeln angegeben werden ([„device1“ , „device2“])
 - Wenn keine Device gesetzt ist, wird die Benachrichtigung an alle Devices gesendet
- Standard-Priority = 0
- Wenn Priorität = 2, dann muss retry und expire gesetzt sein. Sollte nichts gesetzt sein, wird via Script retry auf 30 und expire auf 300 gesetzt

[/etc/icinga2/conf.d/usergroups/alle.conf](#)

```
object UserGroup "icingaadmins-pushover" {
    display_name = "Icinga 2 Admin Group - Pushover"
}
```

[/etc/icinga2/conf.d/hosts/ein_host.conf](#)

```
object Host "foobar" {
    import "generic-host"

    display_name = "Foo bar"
    address = "<IP-Adresse>"

    #Benachrichtigung via email
    vars.notification["mail"] = {
        groups = [ "icingaadmins" ]
    }

    #Benachrichtigung via Pushover
    vars.notification["pushover"] = {
        groups = [ "icingaadmins-pushover" ]
    }
}
```

From:
<https://wiki.da-checka.de/> - PSwiki

Permanent link:
<https://wiki.da-checka.de/doku.php/wiki/programme/icinga2/pushover?rev=1516394243>

Last update: **2018/01/19 21:37**

