



LOGDNA - ONPAGE
INTEGRATION

OnPage and LogDNA Integration

With the OnPage-LogDNA integration, teams can monitor systems and applications, and generate alerts from a saved view to OnPage. They can use "Presence" or "Absence" alerts to generate an OnPage alert to the on-call recipient.

The OnPage-LogDNA integration requires an enterprise administrator's permission for account authentication. If you do not have this permission, please contact your account owner or admin to configure the integration.

In OnPage

- 1- Log in to the OnPage Management Console
- 2- Click on the **Integrations** tab
- 3- Click on either **ALL** or **Information Technology**
- 4- Scroll down and find **OnPage Webhooks**
- 5- Click on **Settings**
- 6- Under Incoming Webhooks, click on **Create**
- 7- Give a Webhook name (i.e., LogDNA integration)
- 8- Copy **the Secret Key and Client ID** and store it in a secure place. **The Keys are only shown once.**

Create Web-hook

All credentials allow your app to access the OnPage API. They are secret. Please don't share your app credentials with anyone, include them in public code repositories, or store them in insecure ways.

Web-hook name

Client ID

Copy

Secret Key

Shown only once! Copy and store it in secured place.

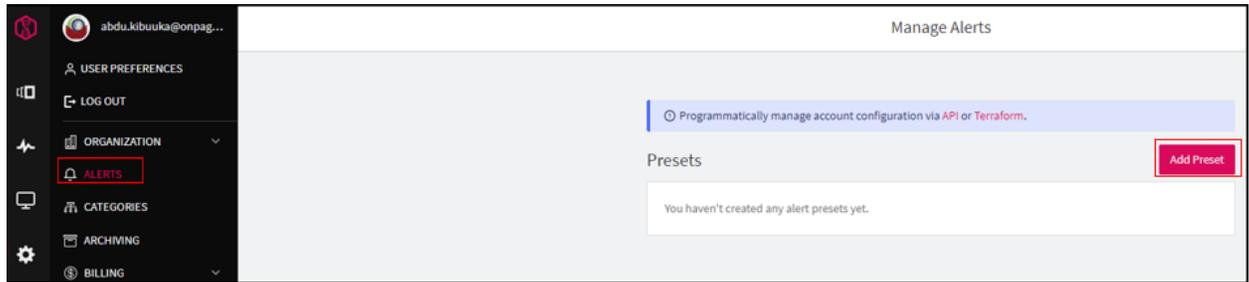
Copy

Save Cancel

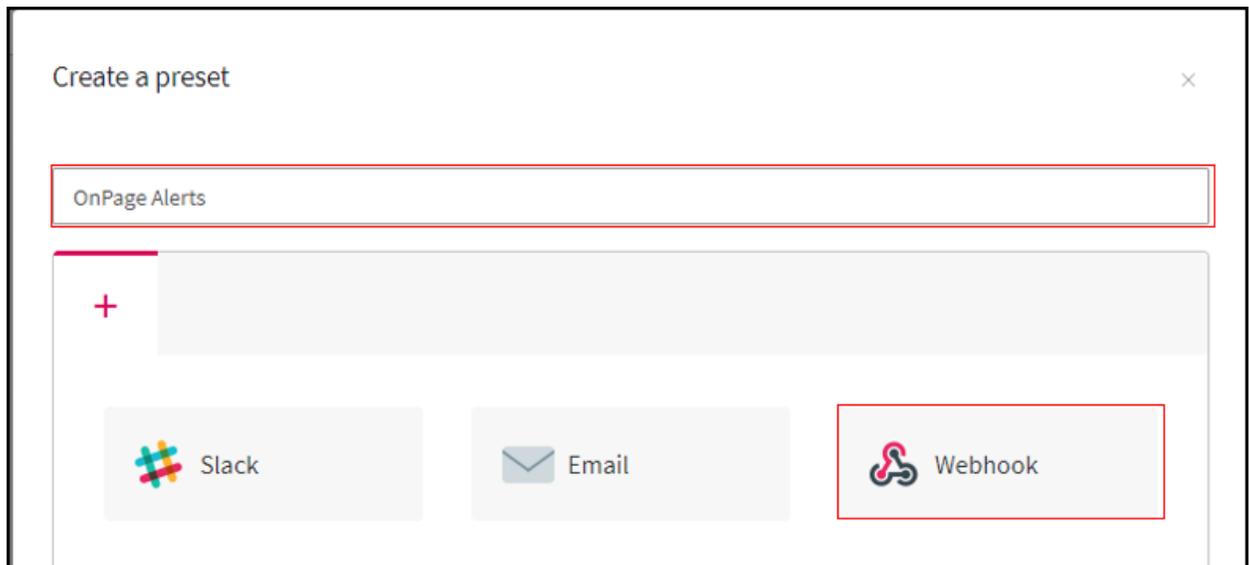
9- Click **Save**

In LogDNA

- 1- Click on **Alerts**
- 2- Click on **Add Preset**



- 3- Name the preset
- 4- Select **Webhook**



- 5- Select the type, either **Presence** or **Absence**
- 6- Select when log lines match for the view per alert interval
- 7- Select when to **send the alert**
- 8- You may choose to add a custom schedule to when the alerts are active
- 9- **Method and URL** should be **POST** and enter the OnPage Webhook URL:
<https://webhook.onpage.com/gw/v1/page>
- 10- Add the headers

Accept	application/json	x
Content-Type	application/json	x

11- Copy JSON payload for the integrations console

```
{
  "clientId": "-YOUR-CLIENT-ID-",
  "secretKey": "-YOUR-SECRET-KEY-",
  "message": {
    "subject": "-YOUR-ALERT-SUBJECT-",
    "body": "*-YOUR-ALERT-MESSAGE-",
    "recipients": ["OPIID1", "OPIID2", ...],
    "priority": "HIGH|LOW"
  }
}
```

12- Paste the payload in the **Body**

- i- Edit the script by inserting the **Client ID, Secret Key**

- ii- Add subject and body and or insert variables of what you would like to see as the subject and body of the message in the OnPage app.

- iii- Enter **Priority** of message. Either HIGH or LOW

- iv- Use the **Tokens** provided in the subject or body of the JSON to access specific information about the view or the matched lines.

Type **Presence** Absence

When Line appears within

Log lines from **Apr 28, 2021**

0
3:00 pm 3:15 pm 3:30 pm 3:45 pm 4:00 pm

Send an alert At the end of 30 seconds
 Immediately after 1 Line

Custom schedule off

Method & URL **POST**

Headers

Accept	application/json	✕
Content-Type	application/json	✕
		Add

Body

```
{
  "clientId":
  "a7f0b84de
  4277b25c1",
  "secretKey":
```

Validate JSON

Tokens

- View name
- Number of matched lines
- Raw output of matched lines
- Severity level
- View URL, with first matched line when possible

13- Click **Test** at the top and you should receive an OnPage test alert

Webhook

Type **Presence** Absence