Alerts API

The Alerts API allows you to configure notifications about significant changes to your traffic based on continual tracking by Akamai's network monitoring platform. It allows you to create and modify alerts based on a wide range of criteria, both static and dynamic, and to configure reports on anomalies.
https://developer.akamai.com/api/luna/alerts/overview.html

Create an alert

1. `GET` **`/alerts/v2/alert-templates/`** to list templates, either based on **`STATIC`** values or **`ADAPTIVE`** modeling of dynamic traffic patterns:

```
{
    "data": [
        {
            "templateId": "a@1",
            "name": "Adaptive",
            "origin": "ADAPTIVE",
            ...
        },
        {
            "templateId": "s@1",
            "name": "High Traffic -- Content
Delivery",
            "origin": "STATIC",
            ...
        }
    ]
}
```

2. `GET` **`/alerts/v2/alert-templates/a@1`** to get a template.
3. Gather **`fields`** such as **`name`** of alert, **`email`** recipients, and relevant metrics. The template guides you for the expected data type.
4. Optionally gather dynamic access control data, such as available Content Provider (CP) codes. `GET` **`/alerts/v2/access-control-data?type=cpcode&templateId=<templateId>`**

```
{
    "data": [
        {
            "fields": { "timezone": null },
            "name": "Jabberwocky",
            "objectId": "44",
            "type": "cpcode"
        }
    ]
}
```

5. `POST` to **`/alerts/v2/alert-definitions`** to create a new alert instance:

```
{
    "definitionId": "",
    "origin": "STATIC",
    "fields": {
        "aca_cpcode": [ "111", "87525" ],
        "alertLowerBound": 5,
        "definitionId": "",
        "email": [ "you@example.com" ],
        "emailBrief": [ "them@example.com" ],
        "emailHtmlFormat": true,
        "isSum": true,
        "name": "myTestAlertName",
        "network": "myNetwork",
        "param": 20,
        "paramName": "cpercent",
        "templateId": "s@150"
    }
}
```

6. Store the **`definitionId`** from the response, such as s**`@123`**.

Find out when the alert fired

1. Use the definitionId: `GET` **`/alerts/v2/alert-definitions/s@123/alert-firings`**.
2. Along with start and end time, the response provides details on what caused the alert to fire:

```
{
    "data": [
        {
            "firingId": "9826198",
            "name": "Adaptive Alert",
            "definitionId": "a@12938",
            "startTime": "2001-01-21T00:01:82.123Z",
            "endTime": "2001-01-21T02:21:12.002Z",
            "fieldMap": {
                "Alert_Condition_(Mbits/sec)": "38.232",
                "Alert_Threshold_(Mbits/sec)": "32.12",
                "CP_Code": "1234",
                "Expected_Value_(Mbits/sec)": "25.792",
                "Message": "Mbps to User: Above model",
                "email": "joy@example.com,juan@example.com"
            }
        }
    ]
}
```

Alerts API: Monitor alert traffic

The Alerts API lets you create and configure notifications about changes to your traffic patterns. Use the API to report on fired alerts and your dynamically modeled traffic patterns.

Plot alert firings

1. `GET /alerts/v2/sparklines?inclRange=true&duration=P7D&definitionIds=a@456`
   - Use **definitionIds=a@456** to specify the alert you want to report on.
   - Use **inclRange=true** to include the expected range of data.
   - Use **duration=P7D** to specify the previous

2. The **anomalies** are periods when alerts fire:

```
"anomalies": [
    {
        "start": "2015-11-09T16:50:00Z",
        "end": "2015-11-11T12:15:00Z",
        "firingId": "9826198"
    },
    ...
]
```

3. The **points** capture observed data along with the expected range:

```
"points": [
    {
        "high": 5863.47,
        "low": 3074.6,
        "x": "2015-11-09T16:50:00Z",
        "y": 4764.86
    },
    ...
]
```

4. Set the **anomalies** over the **points** data to visualize: