Google API Documentation

alertcenter API Documentation

Version: v1

Manages alerts on issues affecting your domain. Note: The current version of this API (v1beta1) is available to all Google Workspace customers.

Google API Documentation

Endpoint: alerts.list

HTTP Method: GET

Path: v1beta1/alerts

Description: Lists the alerts.

AI-Generated Documentation

1. Friendly Technical Description with Common Use Cases

The `GET alertcenter.alerts.list` API endpoint allows you to retrieve a list of alerts from your Google Cloud account. Alerts are events that indicate a potential problem or security issue. You can use this endpoint to investigate alerts, triage them, and take action to resolve them.

Common use cases for this endpoint include:

* **Investigating security incidents:** You can use this endpoint to retrieve a list of alerts related to a specific security incident. This can help you understand the scope of the incident and identify the root cause.
* **Triage alerts:** You can use this endpoint to filter alerts by severity, type, or other criteria. This can help you prioritize which alerts to investigate first.
* **Taking action to resolve alerts:** You can use this endpoint to retrieve the details of a specific alert. This information can help you determine the appropriate action to take to resolve the alert.

2. Example Request with Placeholder Values

```
GET https://alertcenter.googleapis.com/v1beta1/projects/{project_number}/alerts
```

3. Common Parameters and Their Purposes

The following are some of the common parameters that you can use with the `GET alertcenter.alerts.list` endpoint:

* **project_number:** The numeric ID of the project that you want to retrieve alerts for.
* **filter:** A filter expression that you can use to narrow down the list of alerts that are returned. For example, you could filter alerts by severity, type, or status.
* **page_size:** The maximum number of alerts to return in a single page. The default page size is 100.
* **page_token:** A token that you can use to retrieve the next page of results.

Example Code

```python
def list_alerts(project_name, filter_str):
    """Lists alerts in the project.
    Args:
        project_name (str): The Google Cloud Project to use. The project name
            must be in the format - 'projects/<PROJECT_NAME>'.
```

```
        filter_str (str): Expression that defines the filter to apply across
            alerts. The expression is a list of zero or more restrictions combined
            via logical operators `AND` and `OR`. Parentheses are supported, and `OR`
            has higher precedence than `AND`.
            Restrictions have the form `<field> <operator> <value>` and may have a `-`
            character in front of them to indicate negation. The fields map to those
            defined in the corresponding resource.
            The supported operators are:
            * `=` for all value types.
            * `>`, `<`, `>=`, `<=` for integer values.
            * `:`, meaning substring matching, for strings.
            The supported value types are:
            * string literals in quotes.
            * integer literals without quotes.
            * boolean literals `true` and `false` without quotes.
            For example, `resource.type=gce_instance` is a valid filter string.
    """

    client = monitoring_v3.AlertPolicyServiceClient()
    alerts = client.list_alerts(request={"name": project_name, "filter": filter_str})
    for alert in alerts:
        print(alert.name)
        print(alert.display_name)
        print(alert.documentation.content)


```

Google API Documentation

Endpoint: alerts.get

HTTP Method: GET

Path: v1beta1/alerts/{alertId}

Description: Gets the specified alert. Attempting to get a nonexistent alert returns `NOT_FOUND` error.

AI-Generated Documentation

1. Friendly Technical Description with Common Use Cases

The `GET alertcenter.alerts.get` API endpoint retrieves a specific alert by its unique identifier. This endpoint is useful for obtaining detailed information about an alert, including its current status, severity, and associated resources.

Common use cases for this endpoint include:

- Troubleshooting and investigating alerts
- Verifying the resolution of alerts
- Gathering information for reporting and analysis

2. Example Request with Placeholder Values

```
GET https://alertcenter.googleapis.com/v1beta1/projects/{project_id}/alerts/{alert_id}
```

Placeholder Values:

- `{project_id}`: The Google Cloud project ID of the alert.
- `{alert_id}`: The unique identifier of the alert.

3. Common Parameters and Their Purposes

The following parameters are commonly used with the `GET alertcenter.alerts.get` endpoint:

- `name`: The full resource name of the alert.
- `filter`: A filter expression that filters alerts listed in the response. The expression must specify the field name, a comparison operator, and the value that you want to use for filtering. The value must be a string, a number, or a boolean. The comparison operator must be `=`, `!=`, `>`, or `<`. For example, if you are filtering alerts by their severity, you can use the following expression: `severity = "INFO"`.
- `order_by`: One or more fields in the response by which you want to sort the output. For example, you can sort the results by the `timestamp` field in descending order by specifying the `order_by` parameter as `timestamp desc`.
- `page_size`: The maximum number of results to return in a single response page. The server may return fewer results than this value. If unspecified, at most 50 results will be returned. The maximum value is 1000; values above 1000 will be coerced to 1000.
- `page_token`: A page token, received from a previous `ListAlerts` call. Provide this to retrieve the subsequent page. When paginating, all other parameters provided to `ListAlerts` must match the call that provided the page token.

Google API Documentation

Example Code

```python
    from google.cloud import securitycenter

    client = securitycenter.SecurityCenterClient()

    # organization_id is the numeric ID of the organization. e.g.:
    # organization_id = "1234567777"
    # alert_id is the unique identifier for the alert. e.g.:
    # alert_id = "1234"
    alert_name = "organizations/{org_id}/alerts/{alert_id}".format(
        org_id=organization_id, alert_id=alert_id
    )

    try:
        alert = client.get_alert(request={"name": alert_name})
        print(alert)
    except Exception as e:
        print(e)
```

Google API Documentation

Endpoint: alerts.delete

HTTP Method: DELETE

Path: v1beta1/alerts/{alertId}

Description: Marks the specified alert for deletion. An alert that has been marked for deletion is removed from Alert Center after 30 days. Marking an alert for deletion has no effect on an alert which has already been marked for deletion. Attempting to mark a nonexistent alert for deletion results in a `NOT_FOUND` error.

AI-Generated Documentation

1. Friendly Technical Description with Common Use Cases

The `DELETE alertcenter.alerts.delete` endpoint in the Alert Center API allows you to permanently delete an alert from your account. This is useful when you no longer need an alert or if it is causing false positives.

2. Example Request with Placeholder Values

```
DELETE https://alertcenter.googleapis.com/v1beta1/projects/{project_id}/alerts/{alert_id}
```

3. Common Parameters and Their Purposes

The following parameters are commonly used with the `DELETE alertcenter.alerts.delete` endpoint:

* **project_id**: The Google Cloud project ID of the alert to be deleted.
* **alert_id**: The unique identifier of the alert to be deleted.

Example Code

```python
def delete_alert(alert_id):
    """Deletes an alert.

    Args:
        alert_id: The identifier of the alert to delete.
    """

    from google.cloud import securitycenter

    client = securitycenter.SecurityCenterClient()

    alert_name = client.alert_path(
        organization_id="YOUR_ORG_ID", alert_id=alert_id
    )

    client.delete_alert(name=alert_name)
```

```
    print(f"Deleted alert: {alert_name}")
```

Google API Documentation

Endpoint: alerts.undelete

HTTP Method: POST

Path: v1beta1/alerts/{alertId}:undelete

Description: Restores, or "undeletes", an alert that was marked for deletion within the past 30 days. Attempting to undelete an alert which was marked for deletion over 30 days ago (which has been removed from the Alert Center database) or a nonexistent alert returns a `NOT_FOUND` error. Attempting to undelete an alert which has not been marked for deletion has no effect.

AI-Generated Documentation

1. Friendly Technical Description with Common Use Cases

The `POST alertcenter.alerts.undelete` API endpoint allows you to restore a previously deleted alert. This can be useful if you accidentally deleted an alert or if you need to access the alert's data again.

Common use cases for this endpoint include:

* Restoring an alert that was accidentally deleted
* Accessing the data of a deleted alert
* Recovering an alert that was deleted as part of a bulk deletion operation

2. Example Request with Placeholder Values

The following is an example request for the `POST alertcenter.alerts.undelete` endpoint:

```
POST https://alertcenter.googleapis.com/v1beta1/projects/{project_id}/alerts/{alert_id}:undelete
```

where:

* `{project_id}` is the ID of the project that contains the alert
* `{alert_id}` is the ID of the alert to restore

3. Common Parameters and Their Purposes

The following are the common parameters used in the `POST alertcenter.alerts.undelete` endpoint:

* `name`: The name of the alert to restore. This is a required parameter.
* `etag`: The etag of the alert to restore. This is an optional parameter. If specified, the request will only be executed if the etag matches the current etag of the alert.

Google API Documentation

Endpoint: alerts.getMetadata

HTTP Method: GET

Path: v1beta1/alerts/{alertId}/metadata

Description: Returns the metadata of an alert. Attempting to get metadata for a non-existent alert returns `NOT_FOUND` error.

Google API Documentation

Endpoint: alerts.batchDelete

HTTP Method: POST

Path: v1beta1/alerts:batchDelete

Description: Performs batch delete operation on alerts.

Google API Documentation

Endpoint: alerts.batchUndelete

HTTP Method: POST

Path: v1beta1/alerts:batchUndelete

Description: Performs batch undelete operation on alerts.

Google API Documentation

Endpoint: alerts.feedback.create

HTTP Method: POST

Path: v1beta1/alerts/{alertId}/feedback

Description: Creates new feedback for an alert. Attempting to create a feedback for a non-existent alert returns `NOT_FOUND` error. Attempting to create a feedback for an alert that is marked for deletion returns `FAILED_PRECONDITION' error.

Google API Documentation

Endpoint: alerts.feedback.list

HTTP Method: GET

Path: v1beta1/alerts/{alertId}/feedback

Description: Lists all the feedback for an alert. Attempting to list feedbacks for a non-existent alert returns `NOT_FOUND` error.

Google API Documentation

Endpoint: v1beta1.getSettings

HTTP Method: GET

Path: v1beta1/settings

Description: Returns customer-level settings.

Google API Documentation

Endpoint: v1beta1.updateSettings

HTTP Method: PATCH

Path: v1beta1/settings

Description: Updates the customer-level settings.