Opsgenie Provider

The Opsgenie provider is used to interact with the many resources supported by Opsgenie. The provider needs to be configured with the proper credentials before it can be used.

Use the navigation to the left to read about the available resources.

Example Usage

```
# Configure the Opsgenie Provider
provider "opsgenie" {
  api_key = "key"
  api_url = "api.eu.opsgenie.com" #default is api.opsgenie.com
}

# Create a user
resource "opsgenie_user" "test" {
  # ...
}
```

Configuration Reference

The following arguments are supported:

- `api_key` - (Required) The API Key for the Opsgenie Integration. If omitted, the `OPSGENIE_API_KEY` environment variable is used.
- `api_url` - (Optional) The API url for the Opsgenie.

You can generate an API Key within Opsgenie by creating a new API Integration with Read/Write permissions.

Testing and Development

In order to run the Acceptance Tests for development, the following environment variables must also be set:

- `OPSGENIE_API_KEY` - The API Key used for the Opsgenie Integration.

opsgenie_escalation

Manages an Escalation within Opsgenie.

Example Usage

```
data "opsgenie_escalation" "test" {
  name = "existing-escalation"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the escalation.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Escalation.
- `rules` - Escalation rules
- `description` - Escalation Description
- `repeat` - Escalation repeat preferences
- `owner_team_id` - If owner team exist the id of the team is exported

opsgenie_heartbeat

Manages existing heartbeat within Opsgenie.

Example Usage

```
data "opsgenie_heartbeat" "test" {
  name = "genieheartbeat-existing"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the heartbeat

Attributes Reference

The following attributes are exported:

- `description` - An optional description of the heartbeat
- `interval_unit` - Interval specified as minutes, hours or days.
- `interval` - Specifies how often a heartbeat message should be expected.
- `enabled` - Enable/disable heartbeat monitoring.
- `owner_team_id` - Owner team of the heartbeat.
- `alert_message` - Specifies the alert message for heartbeat expiration alert. If this is not provided, default alert message is "HeartbeatName is expired".
- `alert_priority` - Specifies the alert priority for heartbeat expiration alert. If this is not provided, default priority is P3.
- `alert_tags` - Specifies the alert tags for heartbeat expiration alert.

opsgenie_schedule

Manages a Schedule within Opsgenie.

Example Usage

```
resource "opsgenie_schedule" "test" {
  name        = "genieschedule-%s"
  description = "schedule test"
  timezone    = "Europe/Rome"
  enabled     = false
}

resource "opsgenie_schedule" "test" {
  name          = "genieschedule-%s"
  description   = "schedule test"
  timezone      = "Europe/Rome"
  enabled       = false
  owner_team_id = "${opsgenie_team.test.id}"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the schedule.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Schedule.
- `rules` - A Member block as documented below.
- `description` - Timezone of schedule. Please look at Supported Timezone Ids (https://docs.opsgenie.com/docs/supported-timezone-ids) for available timezones - Defaults to "America/New_York".
- `timezone` - The description of schedule.
- `enabled` - Enable/disable state of schedule
- `owner_team_id` - Owner team id of the schedule.

opsgenie_team

Manages existing Team within Opsgenie.

Example Usage

```
data "opsgenie_team" "sre-team" {
  name = "sre-team"
}
```

Argument Reference

The following arguments are supported:

- `name` - The name associated with this team. Opsgenie defines that this must not be longer than 100 characters.

The following attributes are exported:

- `id` - The ID of the Opsgenie Team.
- `member` - A Member block as documented below.
- `description` - A description for this team.

opsgenie_user

Manages existing User within Opsgenie.

Example Usage

```
data "opsgenie_user" "test" {
  username = "user@domain.com"
}
```

Argument Reference

The following arguments are supported:

- `username` - (Required) The email address associated with this user. Opsgenie defines that this must not be longer than 100 characters.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie User.
- `full_name` - The Full Name of the User.
- `role` - The Role assigned to the User. Either a built-in such as 'Owner', 'Admin' or 'User' - or the name of a custom role.
- `locale` - Location information for the user. Please look at Supported Locale Ids (https://docs.opsgenie.com/docs/supported-locales) for available locales.
- `timezone` - Timezone information of the user. Please look at Supported Timezone Ids (https://docs.opsgenie.com/docs/supported-timezone-ids) for available timezones.

opsgenie_api_integration

Manages an API Integration within Opsgenie.

Example Usage

```
resource "opsgenie_api_integration" "example-api-integration" {
  name = "api-based-int"
  type = "API"

  responders {
    type = "user"
    id   = "${opsgenie_user.user.id}"
  }

  responders {
    type = "user"
    id   = "${opsgenie_user.fahri.id}"
  }
}

resource "opsgenie_api_integration" "example-api-integration" {
  name = "api-based-int-2"
  type = "Prometheus"

  responders {
    type = "user"
    id   = "${opsgenie_user.user.id}"
  }

  enabled                         = false
  allow_write_access              = false
  ignore_responders_from_payload = true
  suppress_notifications          = true
  owner_team_id                   = "${opsgenie_team_genies.id}"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the integration. Name must be unique for each integration.
- `type` - (Optional) Type of the integration. (API,Marid,Prometheus ...)
- `allow_write_access` - (Optional) This parameter is for configuring the write access of integration. If write access is restricted, the integration will not be authorized to write within any domain. Defaults to true.
- `enabled` - (Optional) This parameter is for specifying whether the integration will be enabled or not. Defaults to true

- `ignore_responders_from_payload` - (Optional) If enabled, the integration will ignore recipients sent in request payloads. Defaults to false.
- `suppress_notifications` - (Optional) If enabled, notifications that come from alerts will be suppressed. Defaults to false.
- `owner_team_id` - (Optional) Owner team id of the integration.
- `responder` - (Optional) User, schedule, teams or escalation names to calculate which users will receive the notifications of the alert.

`responder` supports the following:

- `type` - (Required) The responder type.
- `id` - (Required) The id of the responder.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie API Integration.
- `api_key` - (Computed) API key of the created integration

Import

API Integrations can be imported using the `id`, e.g.

```
$ terraform import opsgenie_team.team1 812be1a1-32c8-4666-a7fb-03ecc385106c
```

opsgenie_email_integration

Manages an Email Integration within Opsgenie.

Example Usage

```
resource "opsgenie_email_integration" "test" {
  name           = "genieintegration-name"
  email_username = "fahri"
}

resource "opsgenie_email_integration" "test" {
  name = "genieintegration-%s"

  responders {
    type = "user"
    id   = "${opsgenie_user.test.id}"
  }

  responders {
    type = "schedule"
    id   = "${opsgenie_schedule.test.id}"
  }

  responders {
    type = "escalation"
    id   = "${opsgenie_escalation.test.id}"
  }

  responders {
    type = "team"
    id   = "${opsgenie_team.test2.id}"
  }

  email_username                = "test"
  enabled                       = true
  ignore_responders_from_payload = true
  suppress_notifications        = true
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the integration. Name must be unique for each integration.
- `email_username` - (Required) The username part of the email address. It must be unique for each integration.
- `enabled` - (Optional) A Member block as documented below.

- `ignore_responders_from_payload` - (Optional) If enabled, the integration will ignore recipients sent in request payloads. Defaults to false.
- `suppress_notifications` - (Optional) If enabled, notifications that come from alerts will be suppressed. Defaults to false.
- `owner_team_id` - (Optional) Owner team id of the integration.
- `responder` - (Optional) User, schedule, teams or escalation names to calculate which users will receive the notifications of the alert.

`responder` supports the following:

- `type` - (Required) The responder type.
- `id` - (Required) The id of the responder.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Email based Integration.

Import

Email Integrations can be imported using the `id`, e.g.

```
$ terraform import opsgenie_email_integration.test 812be1a1-32c8-4666-a7fb-03ecc385106c
```

opsgenie_escalation

Manages an Escalation within Opsgenie.

Example Usage

```
resource "opsgenie_escalation" "test" {
  name = "genieescalation-%s"

  rules {
    condition   = "if-not-acked"
    notify_type = "default"
    delay       = 1

    recipient {
      type = "user"
      id   = "${opsgenie_user.test.id}"
        }
    }
}

resource "opsgenie_escalation" "test" {
  name          = "genieescalation-%s"
  description   = "test"
  owner_team_id = "${opsgenie_team.test.id}"

  rules {
    condition   = "if-not-acked"
    notify_type = "default"
    delay       = 1

    recipient {
      type = "user"
      id   = "${opsgenie_user.test.id}"
    }

    recipient {
      type = "team"
      id   = "${opsgenie_team.test.id}"
    }

    recipient {
      type = "schedule"
      id   = "${opsgenie_schedule.test.id}"
    }
  }

  repeat  {
    wait_interval          = 10
    count                  = 1
    reset_recipient_states = true
    close_alert_after_all  = false
  }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the escalation.

- `rules` - (Required) List of the escalation rules.
- `description` - (Optional) Description of the escalation.
- `owner_team_id` - (Optional) Owner team id of the escalation.
- `repeat` - (Optional) Repeat preferences of the escalation including repeat interval, count, reverting acknowledge and seen states back and closing an alert automatically as soon as repeats are completed

`rules` supports the following:

- `condition` - (Required) The condition for notifying the recipient of escalation rule that is based on the alert state. Possible values are: if-not-acked and if-not-closed. If not given, if-not-acked is used.
- `notify_type` - (Required) Recipient calculation logic for schedules. Possible values are: `default: on call users next: next users in rotation previous: previous users on rotation users: users of the team admins: admins of the team all: all members of the team`
- `recipient` - (Required) Object of schedule, team, or users which will be notified in escalation. The possible values for participants are: user, schedule, team.
- `delay` - (Required) Time delay of the escalation rule. This parameter takes an object that consists timeAmount field that takes time amount in minutes.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Escalation.

Import

Escalations can be imported using the `id` , e.g.

```
$ terraform import opsgenie_escalation.test 812be1a1-32c8-4666-a7fb-03ecc385106c
```

opsgenie_heartbeat

Manages heartbeat within Opsgenie.

Example Usage

```
resource "opsgenie_heartbeat" "test" {
    name           = "genieheartbeat-%s"
    description    = "test opsgenie heartbeat terraform"
    interval_unit  = "minutes"
    interval       = 10
    enabled        = false
    alert_message  = "Test"
    alert_priority = "P3"
    alert_tags     = ["test","fahri"]
    owner_team_id  = "${opsgenie_team.test.id}"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the heartbeat
- `description` - (Optional) An optional description of the heartbeat
- `interval_unit` - (Required) Interval specified as minutes, hours or days.
- `interval` - (Required) Specifies how often a heartbeat message should be expected.
- `enabled` - (True) Enable/disable heartbeat monitoring.
- `owner_team_id` - (Optional) Owner team of the heartbeat.
- `alert_message` - (Optional) Specifies the alert message for heartbeat expiration alert. If this is not provided, default alert message is "HeartbeatName is expired".
- `alert_priority` - (Optional) Specifies the alert priority for heartbeat expiration alert. If this is not provided, default priority is P3.
- `alert_tags` - (Optional) Specifies the alert tags for heartbeat expiration alert.

Attributes Reference

Only the arguments listed above are exposed as attributes.

Import

Heartbeat Integrations can be imported using the `id`, e.g.

```
$ terraform import opsgenie_heartbeat.test 812be1a1-32c8-4666-a7fb-03ecc385106c
```

opsgenie_maintenance

Manages a Maintenance within Opsgenie.

Example Usage

```
resource "opsgenie_maintenance" "test" {
  description = "geniemaintenance-%s"

  time {
    type       = "schedule"
    start_date = "2019-06-20T17:45:00Z"
    end_date   = "2019-06-20T17:50:00Z"
  }

  rules{}
}

resource "opsgenie_maintenance" "test" {
  description = "geniemaintenance-%s"

  time {
    type       = "schedule"
    start_date = "2019-06-20T17:45:00Z"
    end_date   = "2019-06-%dT17:50:00Z"
  }

  rules {
    state = "enabled"

    entity {
      id   = "${opsgenie_email_integration.test.id}"
      type = "integration"
    }
  }
}
```

Argument Reference

The following arguments are supported:

- `time` - (Required) Time configuration of maintenance. It takes a time object which has type, startDate and endDate fields
- `rules` - (Required) Rules of maintenance, which takes a list of rule objects and defines the maintenance rules over integrations and policies.
- `description` - (Optional) Description for the maintenance.

`times` supports the following:

- `type` - (Required) This parameter defines when the maintenance will be active. It can take one of for-5-minutes, for-30-minutes, for-1-hour, indefinitely or schedule.
- `start_date` - (Required) This parameter takes a date format as (yyyy-MM-dd'T'HH:mm:ssZ) (e.g. 2019-06-11T08:00:00+02:00).
- `end_date` - (Required) This parameter takes a date format as (yyyy-MM-dd'T'HH:mm:ssZ) (e.g. 2019-06-11T08:00:00+02:00).

`rules` supports the following:

- `entity` - (Required) This field represents the entity that maintenance will be applied. Entity field takes two mandatory fields as id and type.
  - `id` - (Required) The id of the entity that maintenance will be applied.
  - `type` - (Required) The type of the entity that maintenance will be applied. It can be either integration or policy.
- `state` - (Required) State of rule that will be defined in maintenance and can take either enabled or disabled for policy type rules. This field has to be disabled for integration type entity rules.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Maintenance Policy.

Import

Maintenance policies can be imported using the `id`, e.g.

```
$ terraform import opsgenie_maintenance.test 812be1a1-32c8-4666-a7fb-03ecc385106c
```

opsgenie_schedule

Manages a Schedule within Opsgenie.

Example Usage

```
resource "opsgenie_schedule" "test" {
  name        = "genieschedule-%s"
  description = "schedule test"
  timezone    = "Europe/Rome"
  enabled     = false
}

resource "opsgenie_schedule" "test" {
  name          = "genieschedule-%s"
  description   = "schedule test"
  timezone      = "Europe/Rome"
  enabled       = false
  owner_team_id = "${opsgenie_team.test.id}"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the schedule.
- `rules` - (Required) A Member block as documented below.
- `description` - (Optional) Timezone of schedule. Please look at Supported Timezone Ids (https://docs.opsgenie.com/docs/supported-timezone-ids) for available timezones - Defaults to "America/New_York".
- `timezone` - (Optional) The description of schedule.
- `enabled` - (Optional) Enable/disable state of schedule
- `owner_team_id` - (Optional) Owner team id of the schedule.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Schedule.

Import

Schedule can be imported using the `id`, e.g.

```
$ terraform import opsgenie_schedule.test 812be1a1-32c8-4666-a7fb-03ecc385106c
```

opsgenie_schedule_rotation

Manages a Schedule Rotation within Opsgenie.

Example Usage

```
resource "opsgenie_schedule_rotation" "test" {
  schedule_id = "${opsgenie_schedule.test.id}"
  name        = "test"
  start_date  = "2019-06-18T17:45:00Z"
  end_date    ="2019-06-20T17:45:00Z"
  type        ="hourly"
  length      = 6

  participant {
    type = "user"
    id   = "${opsgenie_user.test.id}"
  }

  time_restriction {
    type = "time-of-day"

    restriction {
      start_hour = 1
      start_min  = 1
      end_hour   = 10
      end_min    = 1
    }
  }
}
```

Argument Reference

The following arguments are supported:

- `schedule_id` - (Required) Identifier of the schedule.
- `name` - (Optional) Name of rotation.
- `start_date` - (Required) This parameter takes a date format as (yyyy-MM-dd'T'HH:mm:ssZ) (e.g. 2019-06-11T08:00:00+02:00). Minutes may take 0 or 30 as value. Otherwise they will be converted to nearest 0 or 30 automatically
- `end_date` - (Optional) This parameter takes a date format as (yyyy-MM-dd'T'HH:mm:ssZ) (e.g. 2019-06-11T08:00:00+02:00). Minutes may take 0 or 30 as value. Otherwise they will be converted to nearest 0 or 30 automatically
- `type` - (Required) Type of rotation. May be one of daily, weekly and hourly.
- `length` - (Required) Length of the rotation with default value 1.

- `participant` - (Required) List of escalations, teams, users or the reserved word none which will be used in schedule. Each of them can be used multiple times and will be rotated in the order they given. "user,escalation,team,none"
- `time_restriction` - (Required)

`participant` supports the following:

- `type` - (Required) The responder type.
- `id` - (Required) The id of the responder.

`time_restriction` supports the following:

- `type` - (Required) This parameter should be set time-of-day
- `restriction` - (Required) It is a restriction object which is described below. In this case startDay/endDay fields are not supported.

  `restriction` supports the following:

  - `start_hour` - (Required) Value of the hour that frame will start
  - `start_min` - (Required) Value of the minute that frame will start. Minutes may take 0 or 30 as value. Otherwise they will be converted to nearest 0 or 30 automatically.
  - `end_hour` - (Required) Value of the hour that frame will end.
  - `end_min` - (Required) Value of the minute that frame will end. Minutes may take 0 or 30 as value. Otherwise they will be converted to nearest 0 or 30 automatically.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Schedule Rotation

Import

Schedule Rotations can be imported using the `id` , e.g.

```
$ terraform import opsgenie_schedule_rotation.test 812be1a1-32c8-4666-a7fb-03ecc385106c
```

opsgenie_team

Manages a Team within Opsgenie.

Example Usage

```
resource "opsgenie_user" "first" {
  username  = "user@domain.com"
  full_name = "name "
  role      = "User"
}

resource "opsgenie_user" "second" {
  username  = "test@domain.com"
  full_name = "name "
  role      = "User"
}

resource "opsgenie_team" "test" {
  name        = "example"
  description = "This team deals with all the things"

  member {
    id   = "${opsgenie_user.first.id}"
    role = "admin"
  }

  member {
    id   = "${opsgenie_user.second.id}"
    role = "user"
  }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name associated with this team. Opsgenie defines that this must not be longer than 100 characters.
- `description` - (Optional) A description for this team.
- `member` - (Optional) A Member block as documented below.

`member` supports the following:

- `id` - (Required) The UUID for the member to add to this Team.
- `role` - (Optional) The role for the user within the Team - can be either 'admin' or 'user', defaults to 'user' if not set.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Team.

Import

Teams can be imported using the `id`, e.g.

```
$ terraform import opsgenie_team.team1 812be1a1-32c8-4666-a7fb-03ecc385106c
```

opsgenie_user_contact

Manages a User Contact.

Example Usage

```
resource "opsgenie_user_contact" "sms" {
  user_id = "${opsgenie_user.exampleuser.id}"
  to      = "39-123"
  method  = "sms"
}

resource "opsgenie_user_contact" "email" {
  user_id = "${opsgenie_user.exampleuser.id}"
  to      = "fahri@opsgenie.com"
  method  = "email"
}

resource "opsgenie_user_contact" "voice" {
  user_id = "${opsgenie_user.exampleuser.id}"
  to      = "39-123"
  method  = "voice"
}
```

Argument Reference

The following arguments are supported:

- `username` - (Required) The username for contact.(reference)
- `to` - (Required) to field is the address of given method.
- `method` - (Required) This parameter is the contact method of user and should be one of email, sms or voice. Please note that adding mobile is not supported from API.
- `enabled` - (Optional) Enable contact of the user in OpsGenie. Default value is true.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Contact.

Import

Users can be imported using the `id`, e.g.

```
$ terraform import opsgenie_user_contact.testcontact da4faf16-5546-41e4-8330-4d0002b74048
```

opsgenie_user

Manages a User within Opsgenie.

Example Usage

```
resource "opsgenie_user" "test" {
  username  = "user@domain.com"
  full_name = "Test User"
  role      = "User"
  locale    = "en_US"
  timezone  = "America/New_York"
}
```

Argument Reference

The following arguments are supported:

- `username` - (Required) The email address associated with this user. Opsgenie defines that this must not be longer than 100 characters.
- `full_name` - (Required) The Full Name of the User.
- `role` - (Required) The Role assigned to the User. Either a built-in such as 'Owner', 'Admin' or 'User' - or the name of a custom role.
- `locale` - (Optional) Location information for the user. Please look at Supported Locale Ids (https://docs.opsgenie.com/docs/supported-locales) for available locales.
- `timezone` - (Optional) Timezone information of the user. Please look at Supported Timezone Ids (https://docs.opsgenie.com/docs/supported-timezone-ids) for available timezones.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie User.

Import

Users can be imported using the `id`, e.g.

```
$ terraform import opsgenie_user.user da4faf16-5546-41e4-8330-4d0002b74048s
```