

---

# **kawasemi Documentation**

***Release 1.2.0***

**Yusuke Miyazaki**

**Jul 17, 2017**



---

## Contents

---

<b>1</b>	<b>At a Glance</b>	<b>3</b>
<b>2</b>	<b>Contents</b>	<b>5</b>
2.1	Quickstart . . . . .	5
2.2	Sending Notification . . . . .	7
2.3	Backends . . . . .	9
2.4	License . . . . .	14
<b>3</b>	<b>Indices and tables</b>	<b>15</b>
<b>4</b>	<b>Links</b>	<b>17</b>





# Kawasemi

**kawasemi** is a Python library for sending notifications.

Kawasemi provides the following features:

- Very simple ways to send notification
  - GitHub, HipChat, Slack, Twitter, and Yo are supported
- Integrations with web application frameworks helps to load configurations
  - Django is supported



# CHAPTER 1

---

## At a Glance

---

After installation and configuration, you can send notifications with a following simple code:

```
# Python
from kawasemi import Kawasemi
kawasemi = Kawasemi(config)
kawasemi.send("Sample notification.")

# In Django application
from kawasemi.django import send
send("Sample notification.")
```

See [Quickstart](#) page for more details.





### Quickstart

### Requirements

#### Python

- Python 2.7+
- Python 3.3+
- PyPy
- PyPy3

#### Supported Frameworks

- Django 1.8
- Django 1.9
- Django 1.10
- Django 1.11

### Installation

Install kawasemi with your favorite package manager:

```
pip install kawasemi
```

Note: Please use the latest version of setuptools, pip, and wheel.

```
pip install -U setuptools pip wheel
```

## Configurations

You can write the configuration of kawasemi with a dictionary object.

- If you want to send notifications to HipChat:

```
config = {
    "CHANNELS": {
        "hipchat": {
            "_backend": "kawasemi.backends.hipchat.HipChatChannel",
            "api_id": "1234567",
            "token": "ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"
        }
    }
}
```

- You can use one or more channels. To send notifications to both HipChat and Slack:

```
config = {
    "CHANNELS": {
        "hipchat": {
            "_backend": "kawasemi.backends.hipchat.HipChatChannel",
            "api_id": "1234567",
            "token": "ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"
        },
        "slack": {
            "_backend": "kawasemi.backends.slack.SlackChannel",
            "url": "https://hooks.slack.com/services/ABCDEF/GHIJKLM/1234567890"
        }
    }
}
```

- Of course, you can send a message to two different rooms simultaneously. To send notifications to two different HipChat rooms:

```
config = {
    "CHANNELS": {
        "hipchat_first": {
            "_backend": "kawasemi.backends.hipchat.HipChatChannel",
            "api_id": "1234567",
            "token": "ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"
        },
        "hipchat_second": {
            "_backend": "kawasemi.backends.hipchat.HipChatChannel",
            "api_id": "3456789",
            "token": "abcdefghijklmnopqrstuvwxyz0987654321"
        }
    }
}
```

## Usage

You can send notifications with a following simple code:

```

from kawasemi import Kawasemi

config = {
    "CHANNELS": {
        "hipchat": {
            "_backend": "kawasemi.backends.hipchat.HipChatChannel",
            "api_id": "1234567",
            "token": "ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"
        }
    }
}

kawasemi = Kawasemi(config)
kawasemi.send("Sample notification.")
kawasemi.send("Another notification.")

```

## Integration with Django

You can load configurations of kawasemi from `settings.py` by using this integration.

1. Add 'kawasemi' to your `INSTALLED_APPS` setting:

```

INSTALLED_APPS = [
    # Other apps
    'kawasemi.django',
]

```

2. Add `KAWASEMI` to your project settings. You must obtain API keys or tokens from each service.

```

KAWASEMI = {
    "CHANNELS": {
        "hipchat": {
            "_backend": "kawasemi.backends.hipchat.HipChatChannel",
            "api_id": "1234567",
            "token": "ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"
        }
    }
}

```

3. You can send notifications with a following simple code:

```

from kawasemi.django import send

send("Sample notification.")

```

## Sending Notification

### Simple Notification

You can send a notification to all configured channels with a following code:

```

# Python
from kawasemi import Kawasemi
kawasemi = Kawasemi(config)
kawasemi.send("Sample notification.")

```

```
# With Django
from kawasemi.django import send
send("Sample notification.")
```

## Options

You can set some options for the each of backends:

```
kawasemi.send("Sample notification.", options={
    "hipchat": {
        "color": "green"
    },
    "slack": {
        "attachments": [
            {
                "fallback": "Attachment 1",
                "text": "Attachment 1",
                "color": "#36a64f"
            }
        ]
    }
})
```

Please refer to *Backends* for all available options.

## Exceptions

You can handle errors by using `try` statement:

```
try:
    kawasemi.send("Sample notification.")
except Exception as e:
    print("Error!!")
    print(e)
```

You can ignore errors with `fail_silently` parameter:

```
import kawasemi

kawasemi.send("Exceptions are ignored.", fail_silently=True)
```

## Send to a Specific Channel

By default, notifications are sent to all configured channels. You can send notifications to a channel with `channel` parameter.

Example settings:

```
config = {
    "CHANNELS": {
        "channel_1": {
            # ...
```

```

    },
    "channel_2": {
        # ...
    }
}

```

Send a notification to channel\_1:

```
kawasemi.send("sample notification", channel="channel_1")
```

## Backends

Here is a list of available backends and documentation on how to use each backend.

### GitHub

GitHub is online source code hosting service for Git projects.

kawasemi uses one of the [REST API v3](#) for sending notification to GitHub.

### Settings

Please refer to [this page](#) on how to obtain a token.

```

config = {
    "CHANNELS": {
        "github": {
            "_backend": "kawasemi.backends.github.GitHubChannel",
            # Token
            "token": "ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890",
            # Owner of the repository
            "owner": "ymyzk",
            # Repository
            "repository": "test-repo"
        }
    }
}

```

### Options

You can specify all options available in the [REST API v3](#). For instance:

```

kawasemi.send("Issue Title", options={
    "github": {
        "body": ""## ToDo
- [ ] Introduce A
- [ ] Refactor B"",
        "milestone": 123,
        "labels": ["enhancement"],
        "assignees": ["ymyzk"]
    }
})

```

```
}  
})
```

## HipChat

HipChat is hosted group chat and video chat for companies and teams.

kawasemi uses one of the [Room API](#) for sending notification to HipChat.

### Settings

You can obtain a Room API ID and a Room Notification Token from [HipChat Rooms Page](#).

```
config = {  
    "CHANNELS": {  
        "hipchat": {  
            "_backend": "kawasemi.backends.hipchat.HipChatChannel",  
            # Required  
            # Room API ID  
            "api_id": "1234567",  
            # Room Notification Token  
            "token": "ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890",  
            # Optional  
            # Only for HipChat Server  
            "base_url": "https://api.hipchat.com/v2/",  
            # Background color for message  
            # Valid values: yellow, green, red, purple, gray, random  
            "color": "random",  
            # Whether this message should trigger a user notification  
            "notify": True,  
            # Determines how the message is treated by HipChat server and rendered  
            ↪inside HipChat applications  
            # Valid values: html, text  
            "message_format": "html"  
        }  
    }  
}
```

### Options

You can specify all options available in the [Room API](#). For instance:

```
kawasemi.send("Sample notification.", options={  
    "hipchat": {  
        "color": "green",  
        "notify": False,  
        "message_format": "text"  
    }  
})
```

## Slack

Slack is a platform for team communication.

kawasemi uses [Incoming WebHooks](#) which is one of the [Slack API](#) for sending notifications.

## Settings

You can obtain a Webhook URL from [this page](#).

```
config = {
  "CHANNELS": {
    "slack": {
      "_backend": "kawasemi.backends.slack.SlackChannel",
      # Required
      # Webhook URL
      "url": "https://hooks.slack.com/services/ABCDEF/GHIJKLM/1234567890",
      # Optional
      "username": "kawasemi",
      # You can set either icon_url or icon_emoji
      "icon_url": "https://slack.com/img/icons/app-57.png",
      "icon_emoji": ":ghost:",
      # Ex. 1 "channel": "#general"
      # Ex. 2 "channel": "@username"
      "channel": "#general"
    }
  }
}
```

## Options

You can specify `attachments` and `unfurl_links` in options.

## Attachments

You can send richly-formatted messages using attachments. For more information on attachments, please refer to [Attachments](#).

### Test message with attachments

#### Attachment 1

#### Attachment 2

##### Field 1

Field 1 value

##### Field 2

Field 2 value

##### Field 3

Field 3 value

```
kawasemi.send("Test message with attachments", options={
  "slack": {
    "attachments": [
```

```
{
  "fallback": "Attachment 1",
  "text": "Attachment 1",
  "color": "#36a64f"
},
{
  "fallback": "Attachment 2",
  "text": "Attachment 2",
  "color": "warning",
  "fields": [
    {
      "title": "Field 1",
      "value": "Field 1 value",
      "short": False
    },
    {
      "title": "Field 2",
      "value": "Field 2 value",
      "short": True
    },
    {
      "title": "Field 3",
      "value": "Field 3 value",
      "short": True
    }
  ]
}
]
```

## Unfurling

For more details on `unfurl_links`, please refer to [Unfurling](#).

```
kawasemi.send("Sample notification.", options={
  "slack": {
    "unfurl_links": True
  }
})
```

## Twitter

[Twitter](#) is an online social networking service that enables users to send and read short 140-character messages called “tweets”.

kawasemi uses [one](#) of the [REST APIs](#) for sending tweets.

## Settings

You can obtain keys and access tokens from [Twitter Application Management](#).



```

config = {
    "CHANNELS": {
        "twitter": {
            "_backend": "kawasemi.backends.twitter.TwitterChannel",
            # Required
            # Consumer Key (API Key)
            "api_key": "ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890",
            # Consumer Secret (API Secret)
            "api_secret": "ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890",
            # Access Token
            "access_token": "0123456789-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
            # Access Token Secret
            "access_token_secret": "ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"
        }
    }
}

```

## Options

You can specify all parameters available in the [API](#). For instance:

```

kawasemi.send("Sample tweet with location.", options={
    "twitter": {
        "lat": 37.7821120598956,
        "long": -122.400612831116
    }
})

```

## Yo

Yo is the simplest and most efficient communication tool in the world.

kawasemi uses one of the [API](#) for sending Yos.

## Settings

You can obtain an API token from [Yo Dashboard](#).

```

config = {
    "CHANNELS": {
        "yo": {
            "_backend": "kawasemi.backends.yo.YoChannel",
            # Required
            # Your API token
            "api_token": environ.get("CHANNELS_YO_API_TOKEN"),
            # Optional
            # The recipient's username
            "username": environ.get("CHANNELS_YO_USERNAME"),
        }
    }
}

```

### Text

You can send Yo with text (30 characters max) with Yo API v2.0.

```
# Yo with text
kawasemi.send("text")
```

### Options

You can send Yo Location or Yo Link by specifying options. For example:

```
# Yo Link
kawasemi.send(None, options={
    "yo": {
        "link": "http://docs.justyo.co/v1.0/docs/yo"
    }
})

# Yo Location
kawasemi.send(None, options={
    "yo": {
        "location": "35.0261581,135.7818476"
    }
})
```

Please note that you can only send link or location, but not both.

## License

MIT License. Please see [LICENSE](#).

## CHAPTER 3

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



## CHAPTER 4

---

### Links

---

- [Documentation](#)
- [GitHub](#)