

---

# **django-notifyAll Documentation**

*Release 0.1.3*

**Neeraj Dhiman**

**Feb 11, 2020**



---

# Contents

---

<b>1</b>	<b>Why?</b>	<b>3</b>
1.1	For SMS . . . . .	3
1.2	For Push . . . . .	3
1.3	For Email . . . . .	3
<b>2</b>	<b>What?</b>	<b>5</b>
<b>3</b>	<b>How?</b>	<b>7</b>
<b>4</b>	<b>Contents</b>	<b>9</b>
4.1	Overview . . . . .	9
4.2	Installation and Usage . . . . .	10
4.3	Providers . . . . .	11
4.4	Add providers . . . . .	15
4.5	Example . . . . .	15
4.6	Want to contribute ? . . . . .	16
4.7	license . . . . .	16
<b>5</b>	<b>Indices and tables</b>	<b>17</b>



A library which can be used for all types of notifications like SMS, Mail, Push.

- Supports Python2.7+

**Documentation :** <https://django-notifyall.readthedocs.io/en/latest/index.html>



Every application today is dependent on sending out some form of notification - SMS, email or Push. There is no single interface available to manage either the notification type or the notification service provider.

### 1.1 For SMS

- Plivo
- Twilio

both have their own SDK and API

### 1.2 For Push

- Apple Push
- Android Push

both have their own SDK and API (though Firebase could be used as a single interface)

### 1.3 For Email

- The same problem.





## CHAPTER 2

---

What?

---

This library aims to provide a uniform interface to all the developers to use any notification mechanism, from any service provider.



## CHAPTER 3

---

How?

---

Work in progress, the basic idea is to follow the lead of [django-allauth](#).



## 4.1 Overview

### 4.1.1 Requirements

- Python2.7.3.\*
- Other requirements are based on the Provider you choose.

### 4.1.2 Supported Providers

- Gmail
- SendGrid
- Plivo
- Twilio

### 4.1.3 Done (Suggestion/Feedback most welcome)

- Notification Provider Base Class
- Notification SMS Provider Base Class
- Notification Email Provider Base Class
- Notification Push Provider Base Class
- Notification SMS Service Provider - Plivo
- Notification Email Service Provider - Twilio
- Notification Email Service Provider - Gmail
- Notification Email Service Provider - SebdGrid

## 4.1.4 ToDo

- Notification Push Service Provider - ios / APNS
- Notification Push Service Provider - android
- Async Notifications - Celery Integration
- Notifications Reports
- Notification Push Service Provider - OneSignal
- Notification Push Service Provider - AirBop

**Note** : *Push* Notification Support is not enabled yet but it is in our future ToDo list.

## 4.2 Installation and Usage

### 4.2.1 Installation

#### Django

Python package:

```
pip install django-notifyAll
```

settings.py

- Add `notifyAll` in your `INSTALLED_APPS`

Further settings depends on the Provider you Choose. Visit [Provider](#) page for that.

### 4.2.2 Usage

- import notifier from notifyAll services as :

```
from notifyAll.services import notifier
```

- Then call `notifier.Notifier()`, there are some required params which are as follows:

#### source

This signifies who wants to send notification.

#### destination

To whom you want to send notification.

#### notification\_type

This signifies the Notification type available options are `sms`, `email` or `push`.

Note : *push* Notification support is not available yet,

## provider

Type of Provider e.g., plivo, twilio, gmail etc, for more info visit [Provider](#) page for that.

## context

All other information will come under context, it is of type dict. e.g., notification body, cc, bcc or attachment in case of email.

- Possible keys of context :

**body** : Body of Notification.

**cc** : For email provider only.

**bcc** : For email provider only.

**attachment** : For email provider only.

**html\_message** : For email provider only. If you want to send message including **HTML** then you need to send your notification body in the above key.

## Example Usage :

```

from notifyAll.services import notifier

def notify():
    """
    """
    context = {
        'subject': 'subject'
        'body': 'body'
        'html_message': '<h1>html message</h1>'
    }

    data = {
        'source': 'admin@example.com',
        'destination': 'me@example.com',
        'notification_type': 'email',
        'provider': 'gmail',
        'context': context,
    }

    notification = notifier.Notifier(**data)

    return notification.notify()

```

For more information about usage visit our [Example](#) project.

## 4.3 Providers

Most of the Providers needs you to register on their site and get some kind of token/secret in order to use their services.

I'll try to explain what settings you need in order to user Any below providers.

### 4.3.1 Email Providers

- Notification Type for all Email Provider is email

```
notification_type = 'email'
```

- **Configuration Settings can be passed as function arguments as well as *Environment Variable*. The main AIM is to provide possible flexibility to user to use Any *Provider* with any configuration.**

**Note :** Any Provider may need some extra settings (if any) those will be mentioned in respective Provider.

#### > Gmail

- Provider Type for all Gmail Provider is gmail

```
provider = 'gmail'
```

- You can use Gmail as your **SMTP** provider and send Emails from Your own *Gmail* account.
- For this you need below settings to configure in your Django Project.
- Sample settings for Gmail Provider are as follows:

#### As Environment Variable :

- *EMAIL\_USERNAE*
- *EMAIL\_PASSWORD*

#### As Function Arguments:

- *EMAIL\_USERNAE* as *username*.
- *EMAIL\_PASSWORD* as *password*.

#### Example Usage :

```
from notifyAll.services import notifier

def notify():
    """
    """
    context = {
        'subject': 'subject'
        'body': 'body'
        'html_message': '<h1>html message</h1>'
    }

    data = {
        'source': 'admin@example.com',
        'destination': 'me@example.com',
        'notification_type': 'email',
        'provider': 'gmail',
        'context': context,
    }

    notification = notifier.Notifier(**data)

    return notification.notify(username='myuser@gmail.com', password=
↪ 'mypassword')
```



## > SendGrid

- Provider Type for all Sendgrid Provider is `sendgrid`

```
provider = 'sendgrid'
```

- Use *Sendgrid* as your **SMTP** provider
- You need to register to sendgrid for using their services , from their you will get an

### SENDGRID\_API\_KEY :

you sendgrid api\_key, it is visible only once, so you need to copy it after creating.

- Sample settings for SendGrid Provider are as follows:

#### As Environment Variable :

- `SENDGRID_API_KEY`

#### As Function Arguments:

- `SENDGRID_API_KEY` as `sendgrid_api_key`.
- Usage is same as shown in *Gmail* provider example

## 4.3.2 SMS Providers

- Notification Type for all SMS Provider is `sms`

```
notification_type = 'sms'
```

- General Settings you need to configure in Django Project when using any email providers are mentioned below :
- **Some settings can be passed as function arguments as well as in *Django settings*. The main AIM is to provide all possible flexibility to user to use Any *Provider* with any configuration.**
- **If we Add settings in *Django settings* then in entire project those settings will be used But if you want every notification use different provider configuration then that is also possible here.**

## > Plivo

### Requirement

- you need to install python `plivo` package to use this provider.

```
pip install plivo
```

- Provider Type for all Plivo Provider is `plivo`

```
provider = 'plivo'
```

- When you register at *Plivo* it will give you two keys which you need to configure in your Django Project.

### As Django settings :

## PLIVO\_AUTH\_ID

## PLIVO\_AUTH\_TOKEN

### As Function Arguments:

- *PLIVO\_AUTH\_ID* as *auth\_id*
- *PLIVO\_AUTH\_TOKEN* as *auth\_token*

### Example Usage :

```
from notifyAll.services import notifier

def notify():
    """
    """
    data = {
        'source': '<source>',
        'destination': '<destination>',
        'notification_type': 'sms',
        'provider': 'plivo',
        'context': {
            'body': 'test message'
        },
    }

    notification = notifier.Notifier(**data)

    return notification.notify(auth_id='<plivo_auth_id>', auth_token='<plivo_
↪auth_token>')
```

## > Twilio

### Requirement

- you need to install python `twilio` package to use this provider.

```
pip install twilio
```

- Provider Type for all Twilio Provider is `twilio`

```
provider = 'twilio'
```

- When you register at *Twilio* it will give you two keys which you need to configure in your Django Project.

### As Django settings :

## TWILIO\_ACCOUNT\_SID

## TWILIO\_AUTH\_TOKEN

### As Function Arguments:

- *TWILIO\_ACCOUNT\_SID* as *account\_sid*

- `TWILIO_AUTH_TOKEN` as `auth_token`
- Usage is same as shown in *Plivo* provider example

<<<<<< HEAD Note :

Providers can be managed in two ways using *django settings* or as *function arguments*. See respective Providers docs for more info

>>>>>> develop

## 4.4 Add providers

If you want to contribute to this repo and wish to add more providers just follow below instructions:

- All providers (provider apps) comes under this folder `notifyAll/providers`.
- Naming convention of provider app and class is as follows :
  - Class *id* and provider *app\_name* must be same
  - e.g., For *Gmail* provider , *id* & *app\_name* both are same i.e., *gmail*
- If your provider require some settings from user then add them to `notifyAll/settings.py`
- Also update [Provider](#) documentation for your provider (Update respective doc only means email,sms have their own documentation).
- Update service configuration file to add your provider in `ALLOWED_SERVICES` at `notifyAll/settings.py` so that your provider can be enabled for outer world.

## 4.5 Example

Example project can be find out [here](#)

- Its a Django project.Follow below instructions to set up this project.
- Download above example project (if you set up virtual; env then that will good.)
- Go inside example directory from terminal.
- Install all requirements using below command :

```
pip install -r requirements.txt
```

- Add settings for provider you want to test in your `settings.py` , For What settings you need to put in `settings.py` read [Provider](#) page.
- run local server as

```
python manage.py runserver
```

- Access server in browser using `http://localhost:8000` . you will see a page with forms for every notification, Fill any one form and click on send, you will see a success message on successful notification send or an error message (if any)

**Note :** If you face any issue raise an issue on our [github](#) repo

## 4.6 Want to contribute ?

If anyone wish to contribute in improving this library then he is most welcome, due credit will be given to every individual,

### 4.6.1 How you can Contribute ?

- You can add more providers (follow [these](#) instructions)
- Improve/Add Functionality to existing Providers.
- Any Suggestion to improve performance of this library.

## 4.7 license

django-notifyAll

A library which you can use for all types of notifications like SMS, Mail, Push. Copyright (C) 2017 Neeraj Dhiman.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

## CHAPTER 5

---

### Indices and tables

---

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