

---

# **django-rechat Documentation**

***Release 0.1***

**synw**

**Jan 11, 2018**



---

## Contents

---

<b>1</b>	<b>Install</b>	<b>3</b>
1.1	Websockets server . . . . .	3
1.2	Settings . . . . .	3
1.3	Templates . . . . .	4
<b>2</b>	<b>Usage</b>	<b>5</b>
2.1	Create a room . . . . .	5
2.2	Authorizations . . . . .	5
2.3	Urls . . . . .	5
<b>3</b>	<b>Cache management</b>	<b>7</b>
<b>4</b>	<b>History management</b>	<b>9</b>
4.1	Install . . . . .	9
4.2	How to create a custom hook . . . . .	9
<b>5</b>	<b>Activity widget</b>	<b>11</b>



Contents:



### 1.1 Websockets server

Install Django Instant .

**Important note:** you have to use the `database_channels` branch from the `django-instant` repository for this module to work for the moment (it uses advanced features that have not yet landed mainstream).

Instructions are [here](#)

```
pip install redis django-mqueue
```

Add to `INSTALLED_APPS`;

```
'rechat',  
'mqueue',
```

Set the urls:

```
from instant.views import instant_auth  
  
urlpatterns = [  
    # ...  
    url(r'^centrifuge/auth/$', instant_auth, name='instant-auth'),  
    url(r'^chat/', include('rechat.urls')),  
]
```

Run the migrations

### 1.2 Settings

```
# Required  
  
SITE_SLUG = "mysite"
```

```
INSTANT_USERS_CHANNELS = [
    ["$" + SITE_SLUG + "_chat", ["/chat"]]
]

# Optional

# default: True
USE_CACHE = False
# default: 30
RECHAT_CACHE_ITEMS = 20
# default: 60*60*12 (12 hours)
CHAT_CACHE_TTL = 60*60
# default: localhost
RECHAT_REDIS_HOST = 'ip_here'
# default: 6379
RECHAT_REDIS_PORT = 4867
# default: 0
RECHAT_REDIS_DB = 1
```

You will need Redis to be installed to use the cache.

Only the logged in users can chat.

## 1.3 Templates

Create a

Create a `templates/instant/handlers/default.js` Fill it with this content:

```
{% include "rechat/js/handler.js" %}
```

To create a custom handler for a room create a `templates/instant/handlers/<room_name>.js`



### 2.1 Create a room

Go into the admin and create a room.

### 2.2 Authorizations

If public is checked all the logged in users will have access to the room.

A room can be reserved to some groups.

### 2.3 Urls

*/chat/rooms*: get a list of rooms matching user authorizations

*/chat/room/<room\_slug>*: the chatroom



## CHAPTER 3

---

### Cache management

---

By default the cache is enabled. You have to install and run Redis.

Optional: in settings.py:

```
# Redis host: default is localhost
RECHAT_REDIS_HOST = 'ip_here'
# Redis port: default is 6379
RECHAT_REDIS_PORT = 5555
# Redis db: default is 0
RECHAT_REDIS_DB = 1

# number of cached items: default is 30
RECHAT_CACHE = 20
# cache time to live in seconds: default is 60*60*12
RECHAT_CACHE_TTL = 60*60
```

The cache is used to load the last messages when a user reloads a page. The cache can be disabled so that the messages are only broadcasted to the socket, not stored. To disable the cache in settings.py:

```
RECHAT_USE_CACHE = False
```



## 4.1 Install

The chat messages can be stored in the database. You have to enable it in settings.py

```
MQQUEUE_HOOKS = {
    "rechat": {
        "path": "rechat.hook",
    }
}

MQQUEUE_NOSAVE = ["ChatMessage"]
```

## 4.2 How to create a custom hook

Use a *hook.py* file or whatever name with a *save* function that will receive a *MEvent* object:

```
def save(event, conf):
    if (event.event_class == "__chat_msg__"):
        do_something()
```

The in settings:

```
MQQUEUE_HOOKS = {
    "myhook": {
        "path": "myapp.hook",
    }
}

MQQUEUE_NOSAVE = ["ChatMessage"]
```

This way it is possible to implement any persistence layer or process for the chat messages.



## CHAPTER 5

---

### Activity widget

---

To use the graphical activity monitoring widget add this content in `templates/instant/extra_clients.js`:

```
{% include "rechat/js/activity_client.js" %}
```

Be sure to load `<script type="text/javascript" src="{% static 'rechat/js/smoothie.js' %}"></script>` before including the client.

Then you can use it in your templates:

```
<canvas id="chart" width="600" height="100"></canvas>
```