Django Presence Documentation

Release 0.1.2

synw

May 09, 2017

Contents

Contents:

CHAPTER 1

Install

Dependencies:

- Centrifugo : the websockets server
- Django Instant : the Centrifugo <-> Django layer

Install Django Instant according to this documentation

`pip install django-presence`

Add `"presence",` to installed apps

Generate the config and install the worker:

```
python manage.py installpres
```

Settings

Centrifugo config: be sure to have presence enabled in `config.json` (cf django-instant docs):

```
{
 "secret": "70b651f6-775a-4949-982b-b387b31c1d84",
 "anonymous": true,
 "presence":true
 }
```

Settings:

```
# required
SITE_SLUG = "mysite"
# frequency of updates: optional: default is 10
PRESENCE_FREQUENCY = 30
```

Templates

Add a instant/extra_clients.js template with this content:

```
{% include "presence/js/client.js" %}
```

Add a instant/extra_handlers.js template with this content:

```
{% include "presence/js/handlers.js" %}
```

Where you want the presence widget to be put:

```
{% include "presence/widget.html" %}.
```

You can tweak `presence/js/handlers.js` to make your own client-side event handlers.

CHAPTER 2

Workers

The presence data is automaticaly updated from the time based worker asking Centrifugo who is on the socket. This data is broadcasted to the clients over the websocket.

Default worker

The default worker for presence updates is a go module. Install with the command `python manage.py installpres`: it will generate two files in your main project directory:

- `centpres`: the executable
- `centpres_config.json`: the config file

To run the worker: `./centpres`. This will start updating presence info.

Note: you can also compile the go module from the source in *presence/go/src/*

Alternate workers

For those who want a more traditional way two python async backends are supported: Celery and Huey.

For Celery in settings.py:

```
PRESENCE_ASYNC_BACKEND = "celery"

from datetime import timedelta
CELERYBEAT_SCHEDULE = {
    'update-presence': {
        'task': 'presence.tasks.update_presence',
        'schedule': timedelta(seconds=30),
    },
}
```

Then:

```
celery -A project_name beat  -l info --broker='redis://localhost:6379/0'
celery -A project_name worker  -l info --broker='redis://localhost:6379/0'
# or whatever broker or settings
```

For Huey in settings.py:

```
PRESENCE_ASYNC_BACKEND = "huey"

from huey import RedisHuey
HUEY = RedisHuey('your_project_name')
```

Then:

```
python manage.py run_huey
```

Note: Huey is limited: it will update presence info every minute