django-billjobs Documentation

Release 1.0.0

Lionel Chanson

Jun 05, 2020

Contents:

A django billing app for coworking space

What is django-billjobs ?

Django-billjobs is a django app to manage coworkers and create their invoices. It uses Django admin site to manage coworkers account, services the space is providing and manage coworkers invoices.

From version *v1.x* Django-billjobs provides an API to let coworking spaces develop their own online services for their coworkers based on unique account.

Features

Account and Profile : from Django admin site you can create, update, delete coworkers account and their profile. As Django-billjobs is using Django authentication system you can also use *groups*.

Services : A service can be access to the coworking space for a month, a day, or whatever you want. It is just something with a name, a description and a unit price. We keep it simple, really !

Billing : You affect one or more services to one account. It creates an invoice and you can download a pdf of it.

Note: No tax management. This project is coming from non-profit organisation in France. We do not need to manage VAT for services.

Quickstart

```
pip install django-billjobs
```

in your django settings file:

```
INSTALLED_APPS = (
  'django.contrib.admin',
  'django.contrib.auth',
  'django.contrib.contenttypes',
  'django.contrib.sessions',
  'django.contrib.messages',
  'django.contrib.staticfiles',
  'rest_framework',
  'billjobs',
)
```

```
django-admin migrate
django-admin createsuperuser
django-admin runserver
```

3.1 Getting Started

3.2 Contributing

3.2.1 Clone repository

```
git clone https://github.com/ioO/billjobs.git
```

3.2.2 Create a virtualenv with python 3 binary

Billjobs works from **python 3.4 to 3.6**.

Read virtualenv documentation

```
mkvirtualenv django-billjobs --python=/path/to/python3.5
add2virtualenv path/to/django-billjobs
```

3.2.3 Install dependencies

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

3.2.4 Sample settings

The *core/* folder contains sample settings for development. Use **DJANGO_SETTINGS_MODULE** environment variables.

In your virtualenv *bin/postactivate*

```
export DJANGO_SETTINGS_MODULE=core.settings
```

In your virtualenv *bin/postdeactivate*

```
unset DJANGO_SETTINGS_MODULE
```

3.2.5 Database

Development use sqlite3 engine.

```
django-admin migrate
```

3.2.6 Git workflow

Create a feature branch when you develop a new feature, a hotfix and at the end rebase it with **master** branch.

```
git checkout -b new_feature
# do your commits
git checkout master
git pull
git checkout new_feature
git rebase master
git checkout master
git merge --no-ff new_feature
```

3.2.7 Fixtures

You can use development fixtures

```
django-admin loaddata billjobs/fixtures/dev_*
```

If you setup a super user it will be deleted by fixtures data.

- Login : bill
- Password : jobs

3.3 Indices and tables

- genindex
- modindex
- search