
El Géo Paso

Version 1.5.0

Julien M. (guts@github, geojulien@twitter)

déc. 12, 2022

1	Introduction	3
2	Histoire du Projet	5
3	Gestion du projet	7
4	Pré-production	9
5	Git Flow	11
6	Utiliser Apache pour servir le site	15
7	Tâches planifiées	19
8	elgeopaso	21
9	Contributing Guidelines	59
10	Développement	61
11	Documentation	67
	Index des modules Python	69
	Index	71

Description : Simple web application performing statistical analysis on job offers published on Geo-Rezo.**Auteur/ice(s) et contributeur/ice(s) :** Julien M. (guts@github, geojulien@twitter)**Version du projet :** 1.5.0**Code source :** <https://github.com/Guts/elgeopaso/>**Dernière mise à jour :** 12 December 2022

CHAPITRE 1

Introduction

Documentation technique du projet El Géo Paso, statistiques dynamiques sur les offres d'emploi en géomatique publiées sur le forum francophone de géomatique [GeoRezo](#).

Base de connaissances techniques sur le projet El Géo Paso. Cette base, enrichie au gré du temps disponible et de l'envie (autant dire qu'elle est incomplète), a vocation à éviter que le projet ne soit une boîte noire et à faciliter les phases de reprise du développement, celui-ci étant discontinu et irrégulier (bénévolat mon amour).

1.1 Description fonctionnelle

1.1.1 Récupération à partir de GeoRezo

1. Toutes les heures, les dernières offres publiées sont récupérées à partir du [flux RSS du forum Job de GeoRezo](#) et stockées en brut dans une table dédiée ;
2. Chaque nouvelle offre est analysée en s'appuyant sur le kit de traitement du langage naturel NLTK et des correspondances personnalisables en base de données via l'interface d'administration ;

1.1.2 Représentation des données

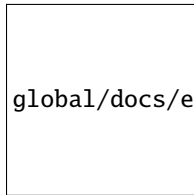
Les données sont ensuite représentées dans différents modes :

- métriques globales ;
 - valeurs absolues du nombre d'offre par période ;
 - valeurs proportionnelles selon le différents critères (types de contrats...)
-

1.2 Description technique

Pour dupliquer le projet, la documentation est dans [le wiki du dépôt](#).

1.2.1 Base de données



global/docs/elpaso_db_models_graph.png

Modèle généré automatiquement par Django Extensions graph-models (pydot)

1.2.2 Briques logicielles

Le projet est développé en Python 3.5.x avec le framework Django et des extensions :

- [feedparser](#) pour la consommation du flux RSS
- [NLTK](#) pour l'analyse sémantique
- [DRF \(Django REST Framework\)](#) et [drf-yasg](#) pour la mise en place de l'API REST et sa documentation automatisée
- [Django Extensions](#) pour la boîte à outils de dev Django

Du côté de l'interface du site web, on retrouve les classiques :

- [Bootstrap](#)
- [D3.js](#), via la surcouche [NVD3](#)
- [Django Suit](#) pour l'habillage de l'interface d'administration

Le site est servi sur le Web par [gunicorn](#) et [nginx](#) ou [Apache](#) selon les plateformes (dev ou production).

2.1 Auteurs

L'idée initiale est celle de [Pierre Vernier](#) et de [Julien Moura](#), portés que nous étions par la regrettée dynamique de [Geotribu](#) !

Les bénévoles de GeoRezo ont aussi contribué à relancer la dynamique lorsqu'elle s'éteignait, que ce soit [Yves Jacolin](#) pour les aspects techniques, [Marc Isenmann](#) pour son intérêt jamais démenti pour l'analyse des offres qu'il modère depuis tant d'années et Bruno Iratchet pour son soutien ponctuel mais non moins fidèle.

2.2 Pourquoi El Paso

Pourquoi El Paso ? Depuis des temps immémoriaux, le choix d'un nom de projet informatique est cornélien (le mot est sûrement trop faible). Né autour de fajitas et de rhum, le nom El Paso s'est imposé comme une évidence. Bien traduit, il représente le petit pas qui sépare un chercheur d'emploi d'un poste. Une simple offre, un simple pas... non, en fait, rien de tout cela, c'est juste parce-que les fajitas étaient bonnes !



3.1 Nommage des versions

Le projet respecte SemVer, pour *Semantic Versioning*.

Consulter :

- le [manifeste officiel de SemVer](#)
- le [paragraphe Wikipédia](#) dédié à cette convention de nommage

Exemples :

1.0.0

CHAPITRE 4

Pré-production

Le site est déployé en pré-production sur [Heroku](#).

4.1 Prérequis

- compte Heroku : le niveau gratuit suffit pour un déploiement basique
- outil en ligne de commande Heroku : [CLI Heroku](#)

4.2 Déployer

4.2.1 Déploiement automatisé

L'application peut être déployée automatiquement via le fichier `app.json`.

Plus d'informations : <https://devcenter.heroku.com/articles/heroku-button>

4.2.2 Déploiement pas à pas

Commandes lancées sous Windows 10 avec WSL activé.

```
# authenticate
heroku login

# add convenient plugin to pull/push from/to `.env` files
# https://github.com/xavdid/heroku-config
heroku plugins:install heroku-config
```

(suite sur la page suivante)

```
# create app in Europe
heroku create elgeopaso-dev --region eu

# add PostgreSQL database and schedule backup
heroku addons:create heroku-postgresql:hobby-dev --version=12
heroku pg:backups schedule --at "02:00 Europe/Paris" DATABASE_URL
# On bash, use simple quotes for the time zone: heroku pg:backups schedule --at '02:00
↪Europe/Paris' DATABASE_URL

# set some environment variables pointing to application's settings
heroku config:set PYTHONHASHSEED=random
heroku config:set WEB_CONCURRENCY=4
heroku config:set DJANGO_DEBUG=False
heroku config:set DJANGO_SETTINGS_MODULE=elgeopaso.settings.production
heroku config:set DJANGO_ALLOWED_HOSTS=elgeopaso-dev.herokuapp.com
heroku config:set DJANGO_ADMIN_URL=admin
heroku config:set DJANGO_SECRET_KEY=$(openssl rand -base64 64)
# on bash: heroku config:set DJANGO_SECRET_KEY="$(openssl rand -base64 64)"

# now, let's deploy
git push heroku master:master
heroku run python manage.py migrate
heroku run python manage.py createsuperuser --email elpaso@georezo.net
heroku run python manage.py check --deploy
heroku open
```

Le projet s'appuie sur l'intégration d'Heroku avec Github pour déployer des versions de test/développement et de pré-production :

- les *Review Apps* : des déploiements temporaires correspondant à une *Pull Request*
- une application gratuite : déploiement automatisé à partir de **master** sur <https://elgeopaso-dev.herokuapp.com/>.

5.1 Branches

- **origin/master** :
 - branche principale
 - correspond à la pré-production
 - *pull request* obligatoire : aucun commit ne peut être poussé directement
 - automatiquement déployée sur Heroku : <https://elgeopaso-dev.herokuapp.com/>
- **origin/develop** :
 - branche générique pour le développement actif
- **origin/housekeeping** :
 - branche dédiée aux opérations courantes de maintenance, mise à jour des dépendances, etc.

5.2 Processus type

1. Une nouvelle branche est créée ou une existante est utilisée
2. Des changements sont apportés dans cette branche et poussés vers la branche principale (master) via une pull-request. Un déploiement temporaire est effectué sur une URL mi-aléatoire. Exemple :
 - travail sur l'amélioration de la lecture du RSS pour gérer les problèmes d'encodage : <https://github.com/Guts/elgeopaso/pull/9>
 - déploiement temporaire correspondant : <https://el-geo-paso-rss-parser-xxsprem.herokuapp.com/> - l'URL est indiqué sur la pull request
3. Une fois les changements achevés et validés, ils sont fusionnés dans la branche principale (*merged*) qui est automatiquement déployée sur Heroku : <https://elgeopaso-dev.herokuapp.com/>

4. Lorsqu'une nouvelle version est finalisée, un *numéro de version* est ajouté via un `git tag`.

Pour comprendre l'étiquetage des commits, voir <https://git-scm.com/book/en/v2/Git-Basics-Tagging> ou Divers - Utilitaires.

5.3 Déploiement

5.3.1 Depuis le serveur de production

Configuration initiale de Git

1. S'ajouter aux utilisateurs

```
sudo adduser geotribu users
```

2. Générer une paire de clés SSH :

```
ssh-keygen -f ~/.ssh/git_elgeopaso_rsa -t rsa -b 4096 -C "elpaso@georezo.net"
```

3. Ajouter la clé publique dans la partie **Deploy keys** du dépôt en lecture seule : <https://github.com/Guts/elgeopaso/settings/keys>.

Voir la documentation officielle de GitHub : <https://developer.github.com/v3/guides/managing-deploy-keys/#deploy-keys>

Configurer le dossier de destination

On utilise le fork du script de François Romain :

```
# récupérer le script
mkdir ~/scripts
cd ~/scripts
git clone git@gist.github.com:36672e8730244764b4a047f6584bd66d.git git-flow-deploy

# lancer le script
source git-flow-deploy/project-create elgeopaso

# modifier le git hook
cd /srv/git/elgeopaso.git/hooks/
sudo nano post-receive

# copier le contenu du fichier : .deploy/git-hooks/post-receive
```

Ressources :

- le fichier du hook post-receive dans le dépôt
- Voir le billet de blog lié : <https://medium.com/@francoisromain/vps-deploy-with-git-fea605f1303b>

5.3.2 Depuis la machine locale

Ajouter le dépôt distant correspondant au serveur :

```
git remote add deploy-prod ssh://geotribu@elgeopaso.georezo.net/srv/git/elgeopaso.git/
```

Pour publier (par exemple depuis master) :

```
git push --follow-tags deploy-prod master
```

Utiliser Apache pour servir le site

Pour servir l'application avec Apache, retenir ces quelques points de vigilance :

- par défaut, Apache ne support pas **WSGI**. Il faut donc utiliser le module `mod_wsgi` pour Apache.
- par défaut sur Ubuntu 18.04, ce module est compilé avec Python 3.6. **Or, il faut utiliser la version compilée avec la même version de Python que celle utilisée par l'application.**

6.1 Prérequis

```
# add repo with latest Apache version
sudo add-apt-repository ppa:ondrej/apache2
# install apache and dependencies
sudo apt install apache2 apache2-dev brotli
# enable brotli module
sudo a2enmod brotli
```

6.2 Déployer l'application Django avec le module Apache WSGI

6.2.1 1. Identifier la bonne version du module et le chemin Python

Installer et utiliser le module inclus dans l'environnement virtuel de l'application

```
# in project folder
cd /var/www/elgeopaso
source .venv/bin/activate

# install mod_wsgi Python module
python -m pip install mod-wsgi==4.7.*
```

(suite sur la page suivante)

(suite de la page précédente)

```
# run the config command to get the directives values
mod_wsgi-express module-config
> LoadModule wsgi_module "/var/www/elgeopaso/.venv/lib/python3.7/site-packages/mod_wsgi/
↳ server/mod_wsgi-py37.cpython-37m-x86_64-linux-gnu.so"
> WSGIPythonHome "/var/www/elgeopaso/.venv"
```

Installer et utiliser le module directement dans Apache

```
# open root input
sudo su -
# in project folder
cd /var/www/elgeopaso
source .venv/bin/activate

# install mod_wsgi Python module
python -m pip install mod-wsgi==4.7.*

# run the config command to get the directives values
mod_wsgi-express install-module
> LoadModule wsgi_module "/usr/lib/apache2/modules/mod_wsgi-py37.cpython-37m-x86_64-
↳ linux-gnu.so"
> WSGIPythonHome "/var/www/elgeopaso/.venv"
```

6.2.2 2. Mettre à jour la configuration Apache

```
# edit Apache module wsgi loader
sudo nano /etc/apache2/mods-available/wsgi.load

# paste the line output in the previous step. For example:
LoadModule wsgi_module "/usr/lib/apache2/modules/mod_wsgi-py37.cpython-37m-x86_64-linux-
↳ gnu.so"
```

6.2.3 3. Enable site, reload and restart

Validate configuration syntax :

```
sudo apache2ctl -t
```

Enable virtual hosts :

```
sudo a2ensite elpaso.conf
sudo a2ensite elpaso-ssl.conf
```

At the end, restart Apache server :

```
sudo service apache2 restart
```

6.2.4 4. Générer le certificat SSL avec Let's Encrypt

Il s'agit principalement de la reproduction de la doc officielle : <https://certbot.eff.org/lets-encrypt/ubuntubionic-apache>.

```
# travailler dans home
mkdir ~/letsencrypt
cd ~/letsencrypt

# enregistrer le dépôt des paquets de certbot - letsencrypt
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository universe
sudo add-apt-repository ppa:certbot/certbot
sudo apt update

# installer le certbot
sudo apt-get install certbot python3-certbot-apache

# lancer le processus en choisissant elgeopaso.georezo.net
sudo certbot --apache
```

Tester le renouvellement automatique :

```
sudo certbot renew --dry-run
```

6.3 Commandes habituelles

```
# check full version and compilation details
apache2ctl -V

# help
apache2ctl -h

# list enabled modules
apache2ctl -M
```

6.4 Resources

- [Django avec Apache et mod_wsgi](#)
- [Déployer Apache sur Ubuntu](#)
- [documentation officielle du module mod_wsgi](#)
- [Le package Python correspondant au module WSGI pour Apache](#)
- [Procédure de création d'un certificat SSL avec Let's Encrypt pour Apache](#)
- [Outil de génération de configuration Apache par Mozilla](#)

CHAPITRE 7

Tâches planifiées

Le projet repose sur certaines tâches récurrentes :

- la récupération des offres d'emploi depuis GeoRezo
- le vidage du cache
- la génération des rapports envoyés par mail
- la génération des fichiers GeoJSON pour les cartes interactives
- le renouvellement du certificat SSL par Let's Encrypt

En production, c'est `cron` qui est utilisé.

7.1 Paramètres de planification

Pour éditer les tâches planifiées lancées par cron avec nano :

```
export VISUAL=nano; crontab -e
```

Insérer :

```
# El Paso
@hourly cd /var/www/elgeopaso && /var/www/elgeopaso/.venv/bin/python /var/www/elgeopaso/
↳manage.py rss2db
@daily cd /var/www/elgeopaso && /var/www/elgeopaso/.venv/bin/python /var/www/elgeopaso/
↳manage.py clear_cache
30 23 * * 7 cd /var/www/elgeopaso && /var/www/elgeopaso/.venv/bin/python /var/www/
↳elgeopaso/manage.py report
05 00 * * 7 cd /var/www/elgeopaso && /var/www/elgeopaso/.venv/bin/python /var/www/
↳elgeopaso/manage.py map_builder

# Let's Encrypt
0 2 * * * root /bin/bash /home/geotribu/letsencrypt/scripts/cron.sh > /
↳home/geotribu/log/cron/letsencrypt.log
```

Pour la syntaxe, le site crontab.guru est une bonne ressource.

8.1 elgeopaso package

8.1.1 Subpackages

elgeopaso.accounts package

Subpackages

elgeopaso.accounts.migrations package

Submodules

elgeopaso.accounts.migrations.0001_initial module

```
class elgeopaso.accounts.migrations.0001_initial.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [('auth', '__first__')]
    initial = True
    operations = [<CreateModel name='Subscription', fields=[('id',
<django.db.models.fields.AutoField>), ('report_hour',
<django.db.models.fields.BooleanField>), ('report_week',
<django.db.models.fields.BooleanField>), ('user',
<django.db.models.fields.related.OneToOneField>)], options={'verbose_name':
'Abonnements mail'}>]
```

Submodules

elgeopaso.accounts.admin module

```
class elgeopaso.accounts.admin.SubscriptionInline(parent_model, admin_site)
    Bases : django.contrib.admin.options.StackedInline
    can_delete = False
    property media
    model
        alias de elgeopaso.accounts.models.Subscription
    verbose_name_plural = 'Options'

class elgeopaso.accounts.admin.UserAdmin(model, admin_site)
    Bases : django.contrib.auth.admin.UserAdmin
    inlines = (<class 'elgeopaso.accounts.admin.SubscriptionInline'>,)
    property media
```

elgeopaso.accounts.apps module

```
class elgeopaso.accounts.apps.AccountsConfig(app_name, app_module)
    Bases : django.apps.config AppConfig
    name = 'accounts'
    verbose_name = "Comptes d'utilisateurs"
```

elgeopaso.accounts.models module

```
class elgeopaso.accounts.models.Subscription(id, user, report_hour, report_week)
    Bases : django.db.models.base.Model
    exception DoesNotExist
        Bases : django.core.exceptions.ObjectDoesNotExist
    exception MultipleObjectsReturned
        Bases : django.core.exceptions.MultipleObjectsReturned
    id
        A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.
    objects = <django.db.models.manager.Manager object>
    report_hour
        A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.
    report_week
        A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.
    user
        Accessor to the related object on the forward side of a one-to-one relation.
        In the example :
```

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Restaurant.place` is a `ForwardOneToOneDescriptor` instance.

`user_id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

elgeopaso.accounts.views module

elgeopaso.api package

Submodules

elgeopaso.api.apps module

```
class elgeopaso.api.apps.ApiConfig(app_name, app_module)
    Bases: django.apps.config AppConfig
    name = 'api'
```

elgeopaso.api.serializers module

```
class elgeopaso.api.serializers.ContractSerializer(*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer
    class Meta
        Bases: object
        fields = '__all__'
        model
            alias de elgeopaso.jobs.models.Contract
class elgeopaso.api.serializers.JobSerializer(*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer
    class Meta
        Bases: object
        fields = '__all__'
        model
            alias de elgeopaso.jobs.models.JobPosition
class elgeopaso.api.serializers.OfferSerializer(*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer
    class Meta
        Bases: object
        depth = 1
        fields = ('id', 'id_rss', 'title', 'pub_date', 'contract', 'place',
            'technologies', 'jobs_positions', 'source', 'content', 'raw_offer')
```

```
    model
        alias de elgeopaso.jobs.models.Offer
class elgeopaso.api.serializers.PlaceSerializer(*args, **kwargs)
    Bases : rest_framework.serializers.ModelSerializer
    class Meta
        Bases : object
        fields = ('name', 'code', 'scale', 'variations')
    model
        alias de elgeopaso.jobs.models.Place
class elgeopaso.api.serializers.PlaceVariationsSerializer(*args, **kwargs)
    Bases : rest_framework.serializers.ModelSerializer
    class Meta
        Bases : object
        fields = ('label',)
    model
        alias de elgeopaso.jobs.models.PlaceVariations
class elgeopaso.api.serializers.TechnoSerializer(*args, **kwargs)
    Bases : rest_framework.serializers.ModelSerializer
    class Meta
        Bases : object
        fields = '__all__'
    model
        alias de elgeopaso.jobs.models.Technology
```

elgeopaso.api.urls module

Application URLs settings.

Learn more here :

- <https://docs.djangoproject.com/fr/2.2/topics/http/urls/>
- <https://docs.djangoproject.com/fr/2.2/ref/urls/>

elgeopaso.api.views module

```
class elgeopaso.api.views.ContractViewSet(**kwargs)
    Bases : rest_framework.viewsets.ReadOnlyModelViewSet
    API endpoint that allows contracts types to be viewed or edited.
    basename = None
    description = None
    detail = None
    name = None
    queryset
    serializer_class
        alias de elgeopaso.api.serializers.ContractSerializer
```

```

    suffix = None

class elgeopaso.api.views.JobViewSet(**kwargs)
    Bases : rest_framework.viewsets.ReadOnlyModelViewSet
    API endpoint that allows contracts types to be viewed or edited.
    basename = None
    description = None
    detail = None
    name = None
    queryset
    serializer_class
        alias de elgeopaso.api.serializers.JobSerializer
    suffix = None

class elgeopaso.api.views.OfferViewSet(**kwargs)
    Bases : rest_framework.viewsets.ReadOnlyModelViewSet
    API endpoint that allows offers to be viewed or edited.
    basename = None
    description = None
    detail = None
    name = None
    queryset
    serializer_class
        alias de elgeopaso.api.serializers.OfferSerializer
    suffix = None

class elgeopaso.api.views.PlaceVariationsViewSet(**kwargs)
    Bases : rest_framework.viewsets.ReadOnlyModelViewSet
    Places used as reference to parse raw offers.
    basename = None
    description = None
    detail = None
    name = None
    queryset
    serializer_class
        alias de elgeopaso.api.serializers.PlaceVariationsSerializer
    suffix = None

class elgeopaso.api.views.PlaceViewSet(**kwargs)
    Bases : rest_framework.viewsets.ModelViewSet
    Places used as reference to parse raw offers.
    basename = None
    description = None
    detail = None

```

```
name = None
queryset
serializer_class
    alias de elgeopaso.api.serializers.PlaceSerializer
suffix = None

class elgeopaso.api.views.TechnoViewSet(**kwargs)
    Bases: rest_framework.viewsets.ModelViewSet
    Places used as reference to parse raw offers.
    basename = None
    description = None
    detail = None
    name = None
    queryset
    serializer_class
        alias de elgeopaso.api.serializers.TechnoSerializer
    suffix = None
```

elgeopaso.cms package

Subpackages

elgeopaso.cms.migrations package

Submodules

elgeopaso.cms.migrations.0001_cms module

```
class elgeopaso.cms.migrations.0001_cms.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [('auth', '__first__')]
    initial = True
    operations = [<CreateModel name='Article', fields=[('id',
<django.db.models.fields.AutoField>), ('title',
<django.db.models.fields.CharField>), ('content', <ckeditor.fields.RichTextField>),
('created', <django.db.models.fields.DateTimeField>), ('updated',
<django.db.models.fields.DateTimeField>), ('author',
<django.db.models.fields.related.ForeignKey>)], options={'verbose_name': 'Contenu
éditorial', 'verbose_name_plural': 'Contenus éditoriaux', 'ordering': ['title'],
'get_latest_by': 'updated'}>, <CreateModel name='Category', fields=[('id',
<django.db.models.fields.AutoField>), ('name', <django.db.models.fields.CharField>),
('description', <django.db.models.fields.TextField>), ('slug_name',
<django.db.models.fields.SlugField>)], options={'verbose_name': 'Type de contenu',
'verbosename_plural': 'Types de contenu', 'ordering': ['name']}>, <AddField
model_name='article', name='category',
field=<django.db.models.fields.related.ForeignKey>>, <AddField model_name='article',
name='slug_title', field=<django.db.models.fields.SlugField>>, <AddField
model_name='article', name='ext_url', field=<django.db.models.fields.URLField>>]
```

```
replaces = [('cms', '0001_initial'), ('cms', '0002_auto_20180115_1329'), ('cms',
'0003_auto_20180115_1415'), ('cms', '0004_auto_20180117_0846'), ('cms',
'0005_article_ext_url')]
```

elgeopaso.cms.migrations.0002_auto_20180308_1417 module

```
class elgeopaso.cms.migrations.0002_auto_20180308_1417.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [('cms', '0001_cms')]
    operations = [<AlterField model_name='article', name='content',
field=<ckeditor.fields.RichTextField>>]
```

elgeopaso.cms.migrations.0003_auto_20180308_1427 module

```
class elgeopaso.cms.migrations.0003_auto_20180308_1427.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [('cms', '0002_auto_20180308_1417')]
    operations = [<AddField model_name='article', name='published',
field=<django.db.models.fields.BooleanField>>, <AlterField model_name='article',
name='content', field=<ckeditor_uploader.fields.RichTextUploadingField>>]
```

elgeopaso.cms.migrations.0004_auto_20180628_2357 module

```
class elgeopaso.cms.migrations.0004_auto_20180628_2357.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [('cms', '0003_auto_20180308_1427')]
    operations = [<AlterField model_name='article', name='author',
field=<django.db.models.fields.related.ForeignKey>>, <AlterField
model_name='article', name='category',
field=<django.db.models.fields.related.ForeignKey>>]
```

Submodules

elgeopaso.cms.admin module

```
class elgeopaso.cms.admin.ArticleAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    fieldsets = (('Métadonnées', {'fields': ('category', 'author')}), ('Titre',
{'fields': ('title', 'slug_title')}), ('Contenu', {'classes': ('full-width',),
'fields': ('content',)}), ('Divers', {'fields': ('ext_url',)}), ('Publication',
{'fields': ('published',)}))
    list_display = ('title', 'slug_title', 'category', 'created', 'updated')
    list_filter = ('category', 'author', 'published')
    property media
```

```
ordering = ('created',)
prepopulated_fields = {'slug_title': ('title',)}
search_fields = ('title', 'content')

class elgeopaso.cms.admin.CategoryAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    list_display = ('name', 'slug_name', 'description')
    list_filter = ('name',)
    property media
    ordering = ('name',)
    prepopulated_fields = {'slug_name': ('name',)}
    search_fields = ('name', 'description')
```

elgeopaso.cms.apps module

Application settings.

```
class elgeopaso.cms.apps.CmsConfig(app_name, app_module)
    Bases: django.apps.config.AppConfig
    name = 'cms'
    verbose_name = 'Contenu éditorial'
```

elgeopaso.cms.models module

```
class elgeopaso.cms.models.Article(id, author, category, title, slug_title, content, ext_url, published,
                                     created, updated)
```

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

author

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOne-Descriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToManyDescriptor instance.

author_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

category

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOne-Descriptor subclass) relation.

In the example :


```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

category_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

content

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

ext_url

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_absolute_url()

get_next_by_created(* , field=<django.db.models.fields.DateTimeField : created>, is_next=True, **kwargs)

get_next_by_updated(* , field=<django.db.models.fields.DateTimeField : updated>, is_next=True, **kwargs)

get_previous_by_created(* , field=<django.db.models.fields.DateTimeField : created>, is_next=False, **kwargs)

get_previous_by_updated(* , field=<django.db.models.fields.DateTimeField : updated>, is_next=False, **kwargs)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

published

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property short_content

slug_title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

updated

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class elgeopaso.cms.models.**Category**(id, name, slug_name, description)

Bases : django.db.models.base.Model

exception DoesNotExist

Bases : django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases : `django.core.exceptions.MultipleObjectsReturned`

article_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

description

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_absolute_url()

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

slug_name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

elgeopaso.cms.sitemaps module

class `elgeopaso.cms.sitemaps.ArticleSitemap`

Bases : `django.contrib.sitemaps.Sitemap`

changefreq = 'monthly'

items()

lastmod(obj)

priority = 0.5

elgeopaso.cms.urls module

Application URLs settings.

elgeopaso.cms.views module**elgeopaso.cms.views.about**(*request*)

Displays global metrics about database on homepage.

elgeopaso.cms.views.docs(*request*)

Documentation content.

elgeopaso.cms.views.view_article(*request, slug, category*)**elgeopaso.cms.views.view_category**(*request, slug*)**elgeopaso.jobs package****Subpackages****elgeopaso.jobs.analyzer package****Subpackages****elgeopaso.jobs.analyzer.georezo package****Subpackages****elgeopaso.jobs.analyzer.georezo.parsers package****Submodules****elgeopaso.jobs.analyzer.georezo.parsers.content module**

Content parser.

class **elgeopaso.jobs.analyzer.georezo.parsers.content.ContentParser**(*offer_id, input_content*)Bases : **object**

Parse content of offers published on GeoRezo to extract informations.

Paramètres

- **offer_id** (*int*) – offer ID (for tracing purposes)
- **input_content** (*str*) – content to parse

parse_technology()

Identify technologies in content.

Type renvoyé **list**

elgeopaso.jobs.analyzer.georezo.parsers.title module

Title parser.

class elgeopaso.jobs.analyzer.georezo.parsers.title.**TitleParser**(*offer_id*, *input_title*)

Bases : **object**

Parse title of offers published on GeoRezo to extract informations.

Paramètres

- **offer_id** (*int*) – offer ID (for tracing purposes)
- **input_title** (*str*) – title to parse

parse_contract_type()

Extraction of types of contracts : CDI, CDD, mission, volontariat, etc.

In theory, offer's title is formatted to contain the type between [...]

Type renvoyé *Contract*

parse_jobs_positions()

Identify job position (“métier”) from passed string.

Type renvoyé *list*

parse_place(*mode=0*)

Extraction of types of contracts : CDI, CDD, mission, volontariat, etc. In theory, place information is within parenthesis “()”.

Paramètres **mode** (*int*) – 0 = STRICT regex (default) : only digits between () 1 = MEDIUM
 regex : alphanumeric between () 2 = SOFT regex : alphanumeric code outside ()

Type renvoyé *Place*

Submodules

elgeopaso.jobs.analyzer.georezo.georezo_analyzer module

Module in charge of analyzing raw offers from GeoRezo : extracting contract type, place, etc. from title and abstract.

class elgeopaso.jobs.analyzer.georezo.georezo_analyzer.**GeorezoOfferAnalyzer**(*li_offers_ids*,
opt_contracts=1,
opt_places=1,
opt_techs=1,
opt_skills=1,
opt_words=1,
source='GEOREZO_RSS',
new=1)

Bases : **object**

Analyze last offers published on GeoRezo and stored in the main table.

analysis()

Perform analysis on offers.

elgeopaso.jobs.crawlers package

Submodules

elgeopaso.jobs.crawlers.georezo_rss_parser module

Name : GeoRezo Jobs RSS Parser Purpose : Parse GeoRezo RSS Python : 3.7+

```
class elgeopaso.jobs.crawlers.georezo_rss_parser.GeorezoRssParser(feed_base_url='https://georezo.net/extern.php?fid=10',
                                                                feed_length_param='show',
                                                                items_to_parse=50,
                                                                user_agent='ElGeoPaso/DEV +https://elgeopaso.georezo.net/')

```

Bases : `object`

Handy module to parse GeoRezo job offers through RSS.

Paramètres

- **feed_base_url** (*str*) – URL to the feed. Defaults to : « <https://georezo.net/extern.php?fid=10> » - optional
- **feed_length_param** (*str*) – name of the URL parameter to specify the number of items. Defaults to : « show » - optional
- **items_to_parse** (*int*) – number of items to request to the feed. Defaults to : 50 - optional
- **user_agent** (*str*) – HTTP user-agent. Defaults to : « ElGeoPaso/DEV +https://elgeopaso.georezo.net/ » - optional

CRAWLER_LATEST_METADATA = 'crawler_georezo_rss_latest.json'

FEED_DATETIME_RAW_FORMAT = '%a, %d %b %Y %H:%M:%S %z'

FEED_DATETIME_RAW_FORMAT_ARROW = 'ddd, D MMM YYYY HH:mm:ss Z'

classmethod extract_offer_id_from_url(*in_url*)

Parse input URL to extract RSS item ID = job offer ID.

Paramètres in_url (*str*) – input URL as string. In GeoRezo RSS, it's : - in raw XML : “<guid isPermaLink= »true »>https://georezo.net/forum/viewtopic.php?pid=331081#p331081</guid>” - parsed by feedparser : entry.id = “https://georezo.net/forum/viewtopic.php?pid=331144#p331144”

Renvoie offer ID

Type renvoyé `int`

classmethod load_previous_crawler_metadata(*from_source*='./last_id_georezo.txt')

Retrieve last parsed item ID from specified source.

Paramètres from_source (*str*) – where to load the ID. Defaults to : « ./last_id_georezo.txt »

Lève

- `NotImplementedError` – [description]
- `ValueError` – [description]

Renvoie dictionary with previous crawler execution metadata

Type renvoyé `dict`

parse_new_offers(*ignore_encoding_errors=True, only_new_offers=True*)

Parse RSS feed, handle errors and filter on new offers.

Paramètres

- **ignore_encoding_errors** (*bool*) – option to ignore encoding exceptions. Defaults to : True
- **only_new_offers** (*bool*) – option to return only new offers basing on the previous crawler execution. If False, all of the feed items will be returned. Defaults to : True

Renvoie list with offers whose identifier is superior to the latest parsed

Type renvoyé *list*

save_parsing_metadata(*feed_parsed*, *save_type*='json')

Dumps some metadata from parsed feed to track behavior and enforce future usage into a structured JSON file.

Paramètres

- **feed_parsed** (*feedparser.FeedParserDict*) – parsed feed
- **save_type** (*str*) – type of save to perform. Defaults to : « json » - optional

Renvoie dictionary of saved data

Type renvoyé *dict*

Example

```
[
  {
    "encoding": "ISO-8859-1",
    "entries_required": 50,
    "entries_total": 50,
    "feed_updated_converted": "2020-03-10 13:07:06+01:00",
    "feed_updated_parsed": [
      2020,
      3,
      10,
      12,
      7,
      6,
      1,
      70,
      0
    ],
    "feed_updated_raw": "Tue, 10 Mar 2020 13:07:06 +0100",
    "latest_offer_id": 331132,
    "status": 200,
    "version": "rss20"
  }
]
```

elgeopaso.jobs.management package

Subpackages

elgeopaso.jobs.management.commands package

Submodules

elgeopaso.jobs.management.commands.csv2db module

Command used to import a CSV from a Georezo database export.

```
python manage.py csv2db --input-csv ./georezo/georezo_db_backup_2016-2017.csv
```

```
class elgeopaso.jobs.management.commands.csv2db.Command(stdout=None, stderr=None,
                                                         no_color=False, force_color=False)

Bases : django.core.management.base.BaseCommand
add_arguments(parser)
    Entry point for subclassed commands to add custom arguments.
args = '<foo bar ...>'
create_parser(*args, **kwargs)
    Create and return the ArgumentParser which will be used to parse the arguments to this command.
handle(*args, **options)
    The actual logic of the command. Subclasses must implement this method.
help = 'Import CSV data into the project database'
import_georezo_backup(input_csv_path)
    TO DOC
    Paramètres input_csv_path (path) –
```

elgeopaso.jobs.management.commands.map_builder module

```
class elgeopaso.jobs.management.commands.map_builder.Command(stdout=None, stderr=None,
                                                             no_color=False, force_color=False)

Bases : django.core.management.base.BaseCommand
args = '<foo bar ...>'
build_geojson_fr_departements(in_geojson)
    Parse input GeoJSON and update needed values to display maps.
    Paramètres in_geojson (Path) – Path to the input GeoJSON
external_downloader(overwrite=False)
    Download external files.
    Paramètres overwrite (bool) – option to overwrite existing files. Defaults to : False - optional
    Renvoie list of downloaded files
    Type renvoyé list
handle(*args, **options)
    Download required files and launch GeoJSON build.
help = '\n Commands to generate geojson files used for map visualization.\n '
url_to_file(url, final_file, timeout=60)
    Download file from URL to local storage.
    Source : https://stackoverflow.com/a/16696317/2556577
    Paramètres
    — url (str) – [description]
    — final_file (Path) – [description]
    — timeout (int) – [description]. Defaults to : 60 - optional
```

elgeopaso.jobs.management.commands.report module

```
class elgeopaso.jobs.management.commands.report.Command(stdout=None, stderr=None,
                                                         no_color=False, force_color=False)

    Bases : django.core.management.base.BaseCommand
    args = '<foo bar ...>'
    handle(*args, **options)
        TO DO
    help = '\n Commands to manage weekly report\n '
    now = <Arrow [2022-12-12T14:22:50.711287+01:00]>
```

elgeopaso.jobs.management.commands.reset_analisis module

```
class elgeopaso.jobs.management.commands.reset_analisis.Command(stdout=None, stderr=None,
                                                                no_color=False,
                                                                force_color=False)

    Bases : django.core.management.base.BaseCommand
    add_arguments(parser)
        Entry point for subclassed commands to add custom arguments.
    args = '<foo bar ...>'
    create_parser(*args, **kwargs)
        Create and return the ArgumentParser which will be used to parse the arguments to this command.
    handle(*args, **options)
        List commands to launch.
    help = 'Empty tables and launch GeorezoOfferAnalyzer from the whole georezo_rss
table.'
```

elgeopaso.jobs.management.commands.rss2db module

Custom Django management command to parse GeoRezo feed and launch analisis. See : <https://docs.djangoproject.com/fr/2.2/howto/custom-management-commands/>

```
class elgeopaso.jobs.management.commands.rss2db.Command(stdout=None, stderr=None,
                                                         no_color=False, force_color=False)

    Bases : django.core.management.base.BaseCommand
    Commands to manage offers sync and analisis.
    Two main steps :
    1. Crawl GeoRezo RSS to get new offers, analyze it and store into the database.
    2. Relaunch offer analisis on offers which have been manually modified (through the admin)

    Paramètres BaseCommand ([type]) – [description]
    Lève ValueError – [description]
    Renvoi [description]
    Type renvoyé [type]
```


add_arguments(*parser*)

Add arguments to the CLI.

Paramètres **parser** (*CommandParser*) – command parser

args = '<foo bar ...>'

create_parser(*args, **kwargs)

Super a command parser.

Renvoie [description]

Type renvoyé CommandParser

handle(*args, **options)

The actual logic of the command. Subclasses must implement this method.

help = 'Commands to manage offers sync and analisis. 2 main steps:\n 1. Crawl GeoRezo RSS to get new offers, analyze it and store into\n the database.\n\n 2. Relaunch offer analisis on offers which have been manually\n modified (through the admin)'

now = <Arrow [2022-12-12T14:22:50.866877+01:00]>

elgeopaso.jobs.migrations package

Submodules

elgeopaso.jobs.migrations.0001_initial module

class elgeopaso.jobs.migrations.0001_initial.**Migration**(*name*, *app_label*)

Bases: django.db.migrations.migration.Migration

dependencies = []

initial = True

```

operations = [<CreateModel name='Contract', fields=[('abbrv',
<django.db.models.fields.CharField>), ('name', <django.db.models.fields.CharField>),
('comment', <django.db.models.fields.TextField>), ('created',
<django.db.models.fields.DateTimeField>), ('updated',
<django.db.models.fields.DateTimeField>)], options={'verbose_name': 'Type de
contrat', 'verbose_name_plural': 'Types de contrats', 'ordering': ['abbrv']}>,
<CreateModel name='GeorezoRSS', fields=[('id_rss',
<django.db.models.fields.IntegerField>), ('title',
<django.db.models.fields.CharField>), ('content',
<django.db.models.fields.TextField>), ('pub_date',
<django.db.models.fields.DateTimeField>), ('created',
<django.db.models.fields.DateTimeField>), ('updated',
<django.db.models.fields.DateTimeField>), ('source',
<django.db.models.fields.BooleanField>), ('to_update',
<django.db.models.fields.BooleanField>)], options={'verbose_name_plural': "Offres
d'emploi brutes issues du RSS de GeoRezo", 'db_table': 'georezo_rss',
'get_latest_by': 'pub_date', 'unique_together': {( 'id_rss', 'pub_date',
'source')}}>, <CreateModel name='JobPosition', fields=[('name',
<django.db.models.fields.CharField>), ('comment',
<django.db.models.fields.CharField>), ('created',
<django.db.models.fields.DateTimeField>), ('updated',
<django.db.models.fields.DateTimeField>)], options={'verbose_name': 'Métier',
'verbose_name_plural': 'Métiers', 'ordering': ['name']}>, <CreateModel name='Place',
fields=[('name', <django.db.models.fields.CharField>), ('code',
<django.db.models.fields.CharField>), ('scale',
<django.db.models.fields.CharField>), ('created',
<django.db.models.fields.DateTimeField>), ('updated',
<django.db.models.fields.DateTimeField>)], options={'verbose_name': 'Lieu',
'verbose_name_plural': 'Lieux', 'ordering': ['code']}>, <CreateModel name='Source',
fields=[('id', <django.db.models.fields.AutoField>), ('name',
<django.db.models.fields.CharField>), ('url', <django.db.models.fields.URLField>),
('comment', <django.db.models.fields.TextField>), ('created',
<django.db.models.fields.DateTimeField>), ('updated',
<django.db.models.fields.DateTimeField>)], options={'verbose_name': "Source de
l'offre", 'verbose_name_plural': 'Sources'}>, <CreateModel name='Technology',
fields=[('id', <django.db.models.fields.AutoField>), ('name',
<django.db.models.fields.CharField>), ('license',
<django.db.models.fields.CharField>), ('type_soft',
<django.db.models.fields.CharField>), ('created',
<django.db.models.fields.DateTimeField>), ('updated',
<django.db.models.fields.DateTimeField>)], options={'verbose_name': 'Technologie',
'verbose_name_plural': 'Technologies', 'ordering': ['name']}>, <CreateModel
name='TechnologyVariations', fields=[('id', <django.db.models.fields.AutoField>),
('label', <django.db.models.fields.CharField>), ('name',
<django.db.models.fields.related.ForeignKey>)], options={'verbose_name': 'Variante
des technologies', 'verbose_name_plural': 'Variantes des technologies'}>,
<CreateModel name='PlaceVariations', fields=[('id',
<django.db.models.fields.AutoField>), ('label',
<django.db.models.fields.CharField>), ('name',
<django.db.models.fields.related.ForeignKey>)], options={'verbose_name': 'Variante
de lieu', 'verbose_name_plural': 'Variantes des lieux'}>, <CreateModel name='Offer',
fields=[('id', <django.db.models.fields.AutoField>), ('id_rss',
<django.db.models.fields.IntegerField>), ('title',
<django.db.models.fields.CharField>), ('content',
<django.db.models.fields.TextField>), ('pub_date',
<django.db.models.fields.DateTimeField>), ('week',
<django.db.models.fields.IntegerField>), ('created',
<django.db.models.fields.DateTimeField>), ('updated',
<django.db.models.fields.DateTimeField>), ('contract',
<django.db.models.fields.related.ForeignKey>), ('jobs_positions',

```

elgeopaso.jobs.migrations.0002_remove_offer_week module

```
class elgeopaso.jobs.migrations.0002_remove_offer_week.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [('jobs', '0001_initial')]
    operations = [<RemoveField model_name='offer', name='week'>]
```

elgeopaso.jobs.migrations.0003_variations_unique module

```
class elgeopaso.jobs.migrations.0003_variations_unique.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [('jobs', '0002_remove_offer_week')]
    operations = [<AlterField model_name='contractvariations', name='label',
field=<django.db.models.fields.CharField>>, <AlterField
model_name='jobpositionvariations', name='label',
field=<django.db.models.fields.CharField>>, <AlterField
model_name='placevariations', name='label',
field=<django.db.models.fields.CharField>>, <AlterField
model_name='technologyvariations', name='label',
field=<django.db.models.fields.CharField>>]
```

elgeopaso.jobs.templatetags package**Submodules****elgeopaso.jobs.templatetags.tpl_extras module**

```
elgeopaso.jobs.templatetags.tpl_extras.query_transform(context, **kwargs)
```

Returns the URL-encoded querystring for the current page, updating the params with the key/value pairs passed to the tag.

E.g : given the querystring ?foo=1&bar=2 {*% query_transform bar=3 %*} outputs ?foo=1&bar=3 {*% query_transform foo="baz" %*} outputs ?foo=baz&bar=2 {*% query_transform foo="one" bar="two" baz=99 %*} outputs ?foo=one&bar=two&baz=99

A RequestContext is required for access to the current querystring.

elgeopaso.jobs.templatetags.version module

Application views.

Learn more here : <https://docs.djangoproject.com/fr/2.2/topics/http/views/>

```
elgeopaso.jobs.templatetags.version.version_date(date_format='%d/%m/%Y')
```

Return the last modification date based on git. If git folder does not exist, an empty string is returned instead.

Renvoie last modification date

Type renvoyé str

Example

```
# from a Django template, first load the custom templatetag
{% load version %}

# in a text div
<p>
    My awesome project - Last updated: {% version_date %}
</p>
```

`elgeopaso.jobs.templatetags.version.version_number()`

Return the project version as number.

Renvoie version number

Type renvoyé `str`

Example

```
# from a Django template, first load the custom templatetag
{% load version %}

# in a text div
<p>
    My awesome project - version {% version_number %}
</p>
```

Submodules

`elgeopaso.jobs.admin` module

Application in administration panel.

class `elgeopaso.jobs.admin.ContractVariationsAdmin(model, admin_site)`

Bases: `django.contrib.admin.options.ModelAdmin`

list_display = ('label', 'name')

list_filter = ('name',)

property media

model

alias de `elgeopaso.jobs.models.ContractVariations`

search_fields = ('label',)

class `elgeopaso.jobs.admin.ContractVariationsInline(parent_model, admin_site)`

Bases: `django.contrib.admin.options.TabularInline`

list_display = ('name', 'label')

property media

model

alias de `elgeopaso.jobs.models.ContractVariations`

class `elgeopaso.jobs.admin.ContractsAdmin(model, admin_site)`

Bases: `django.contrib.admin.options.ModelAdmin`

```

    inlines = (<class 'elgeopaso.jobs.admin.ContractVariationsInline'>,)
    list_display = ('abbrv', 'name', 'comment')
    list_filter = ('abbrv',)
    property media
    ordering = ('abbrv',)
    readonly_fields = ('created', 'updated')
    search_fields = ('name', 'abbrv')

class elgeopaso.jobs.admin.GeorezoRSSAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    actions = [<function GeorezoRSSAdmin.offers_to_update>]
    date_hierarchy = 'pub_date'
    fieldsets = (('Modifier', {'fields': ('title', 'content', 'to_update')}), ('Date',
    {'fields': ('pub_date', 'created', 'updated')}), ('Référence', {'fields':
    ('id_rss',)}), ('Autres', {'fields': ('source', 'show_clean_offer')}))
    formfield_overrides = {<class 'django.db.models.fields.CharField'>: {'widget':
    <django.forms.widgets.TextInput object>}, <class
    'django.db.models.fields.TextField'>: {'widget': <django.forms.widgets.Textarea
    object>}}
    list_display = ('id_rss', 'title', 'short_content', 'pub_date', 'created',
    'updated')
    list_display_links = ('id_rss', 'title')
    list_filter = ('pub_date', 'created', 'updated', 'to_update')
    list_select_related = True
    property media
    offers_to_update(request, queryset)
    readonly_fields = ('created', 'id_rss', 'pub_date', 'source', 'updated',
    'show_clean_offer')
    search_fields = ('title', 'content')
    show_clean_offer(obj)

class elgeopaso.jobs.admin.JobPositionAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    inlines = (<class 'elgeopaso.jobs.admin.JobPositionVariationsInline'>,)
    list_display = ('name', 'comment')
    list_filter = ('name',)
    property media
    readonly_fields = ('created', 'updated')
    search_fields = ('name',)

class elgeopaso.jobs.admin.JobPositionVariationsAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    list_display = ('label', 'name')
    list_filter = ('name',)

```

```
property media
model
    alias de elgeopaso.jobs.models.JobPosition
search_fields = ('label',)

class elgeopaso.jobs.admin.JobPositionVariationsInline(parent_model, admin_site)
    Bases: django.contrib.admin.options.TabularInline
    list_display = ('name', 'label')
    property media
    model
        alias de elgeopaso.jobs.models.JobPositionVariations

class elgeopaso.jobs.admin.OfferAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    date_hierarchy = 'pub_date'

    fieldsets = (('Contenu', {'fields': ('title', 'content')}), ('Date', {'fields':
    ('pub_date', 'yearweek', 'created', 'updated')}), ('Informations extraites',
    {'fields': ('contract', 'technologies', 'place', 'jobs_positions')}), ('Autres',
    {'fields': ('show_raw_offer', 'source')}))

    list_display = ('id_rss', 'title', 'short_content', 'contract', 'place', 'pub_date')
    list_filter = ('raw_offer__to_update', 'pub_date', 'contract', 'technologies',
    'place')

    list_select_related = True
    property media
    ordering = ('-pub_date',)

    readonly_fields = ('content', 'contract', 'created', 'jobs_positions', 'place',
    'pub_date', 'show_raw_offer', 'source', 'technologies', 'title', 'updated',
    'yearweek')

    search_fields = ('title', 'content')
    show_raw_offer(obj)

class elgeopaso.jobs.admin.PlaceAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    inlines = (<class 'elgeopaso.jobs.admin.PlaceVariationsInline'>,)
    list_display = ('name', 'code', 'scale')
    list_filter = ('scale',)
    property media
    ordering = ('code',)
    readonly_fields = ('created', 'updated')
    search_fields = ('name', 'code')

class elgeopaso.jobs.admin.PlaceVariationsAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    list_display = ('label', 'name')
    list_filter = ('name',)
```

```

    property media
    model
    alias de elgeopaso.jobs.models.PlaceVariations
    search_fields = ('label',)

class elgeopaso.jobs.admin.PlaceVariationsInline(parent_model, admin_site)
    Bases: django.contrib.admin.options.TabularInline
    list_display = ('name', 'label')
    property media
    model
    alias de elgeopaso.jobs.models.PlaceVariations

class elgeopaso.jobs.admin.SourcesAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    list_display = ('name', 'url', 'comment')
    list_filter = ('name',)
    property media
    ordering = ('name',)
    readonly_fields = ('created', 'updated')

class elgeopaso.jobs.admin.TechnoVariationsInline(parent_model, admin_site)
    Bases: django.contrib.admin.options.TabularInline
    list_display = ('name', 'label')
    property media
    model
    alias de elgeopaso.jobs.models.TechnologyVariations

class elgeopaso.jobs.admin.TechnologyAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    inlines = (<class 'elgeopaso.jobs.admin.TechnoVariationsInline'>,)
    list_display = ('name', 'license', 'type_soft')
    list_filter = ('name', 'license', 'type_soft')
    property media
    ordering = ('name',)
    readonly_fields = ('created', 'updated')
    search_fields = ('name',)

class elgeopaso.jobs.admin.TechnologyVariationsAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    list_display = ('label', 'name')
    list_filter = ('name',)
    property media
    model
    alias de elgeopaso.jobs.models.TechnologyVariations
    search_fields = ('label',)

```

elgeopaso.jobs.apps module

Application settings.

```
class elgeopaso.jobs.apps.JobsConfig(app_name, app_module)
    Bases : django.apps.config.AppConfig
    name = 'jobs'
    verbose_name = "Offres d'emploi"
```

elgeopaso.jobs.decorators module

elgeopaso.jobs.decorators.**conditional_cache**(*decorator*)

Returns decorated view if user is not admin. Un-decorated otherwise

elgeopaso.jobs.filters module

Filters.

Learn more here : <https://django-filter.readthedocs.io/en/master/>

```
class elgeopaso.jobs.filters.OfferFilter(data=None, queryset=None, *, request=None, prefix=None)
    Bases : django_filters.rest_framework.filterset.FilterSet
    Filters related to search within offers.
    class Meta
        Bases : object
        fields = ['contract', 'place', 'technologies', 'pub_date', 'content', 'title',
                  'raw_offer__to_update']
        model
            alias de elgeopaso.jobs.models.Offer
    base_filters = {'content': <django_filters.filters.CharFilter object>, 'contract':
<django_filters.filters.ModelChoiceFilter object>, 'date':
<django_filters.filters.DateFromToRangeFilter object>, 'place':
<django_filters.filters.ModelChoiceFilter object>, 'pub_date':
<django_filters.filters IsoDateTimeFilter object>, 'raw_offer__to_update':
<django_filters.rest_framework.filters.BooleanFilter object>, 'technologies':
<django_filters.filters.ModelMultipleChoiceFilter object>, 'title':
<django_filters.filters.CharFilter object>}
    declared_filters = {'content': <django_filters.filters.CharFilter object>, 'date':
<django_filters.filters.DateFromToRangeFilter object>, 'raw_offer__to_update':
<django_filters.rest_framework.filters.BooleanFilter object>, 'title':
<django_filters.filters.CharFilter object>}
```


elgeopaso.jobs.models module

Application database models.

class elgeopaso.jobs.models.**Contract**(*abbrv, name, comment, created, updated*)

Bases : `django.db.models.base.Model`

exception **DoesNotExist**

Bases : `django.core.exceptions.ObjectDoesNotExist`

exception **MultipleObjectsReturned**

Bases : `django.core.exceptions.MultipleObjectsReturned`

abbrv

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

comment

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

contractvariations_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_next_by_created(**, field=<django.db.models.fields.DateTimeField : created>, is_next=True, **kwargs*)

get_next_by_updated(**, field=<django.db.models.fields.DateTimeField : updated>, is_next=True, **kwargs*)

get_previous_by_created(**, field=<django.db.models.fields.DateTimeField : created>, is_next=False, **kwargs*)

get_previous_by_updated(**, field=<django.db.models.fields.DateTimeField : updated>, is_next=False, **kwargs*)

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = `<django.db.models.manager.Manager object>`

offer_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

updated

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class elgeopaso.jobs.models.ContractVariations(*id, label, name*)

Bases : django.db.models.base.Model

exception DoesNotExist

Bases : django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases : django.core.exceptions.MultipleObjectsReturned

ND = 'UNDEFINED'

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

label

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

name_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

class elgeopaso.jobs.models.GeorezoRSS(*args, **kwargs)

Bases : django.db.models.base.Model

GeoRezo RAW offers.

exception DoesNotExist

Bases : django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases : django.core.exceptions.MultipleObjectsReturned

clean_offer

Accessor to the related object on the reverse side of a one-to-one relation.

In the example :

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

Place.restaurant is a ReverseOneToOneDescriptor instance.

content

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
get_next_by_created(* , field=<django.db.models.fields.DateTimeField : created>, is_next=True,
                    **kwargs)
```

```
get_next_by_updated(* , field=<django.db.models.fields.DateTimeField : updated>, is_next=True,
                    **kwargs)
```

```
get_previous_by_created(* , field=<django.db.models.fields.DateTimeField : created>, is_next=False,
                        **kwargs)
```

```
get_previous_by_updated(* , field=<django.db.models.fields.DateTimeField : updated>, is_next=False,
                        **kwargs)
```

id_rss

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
objects = <django.db.models.manager.Manager object>
```

```
property offre_traitee
```

pub_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
property short_content
```

source

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

to_update

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

updated

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class elgeopaso.jobs.models.JobPosition(name, comment, created, updated)
```

```
Bases : django.db.models.base.Model
```

```
exception DoesNotExist
```

```
Bases : django.core.exceptions.ObjectDoesNotExist
```

```
exception MultipleObjectsReturned
```

```
Bases : django.core.exceptions.MultipleObjectsReturned
```

comment

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
get_next_by_created(* , field=<django.db.models.fields.DateTimeField : created>, is_next=True,
                    **kwargs)
```

```

get_next_by_updated(*, field=<django.db.models.fields.DateTimeField : updated>, is_next=True,
                    **kwargs)

get_previous_by_created(*, field=<django.db.models.fields.DateTimeField : created>, is_next=False,
                       **kwargs)

get_previous_by_updated(*, field=<django.db.models.fields.DateTimeField : updated>, is_next=False,
                        **kwargs)

```

jobpositionvariations_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```

class Child(Model):
    parent = ForeignKey(Parent, related_name='children')

```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

offer_set

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example :

```

class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')

```

Pizza.toppings and Topping.pizzas are ManyToManyDescriptor instances.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

updated

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class elgeopaso.jobs.models.JobPositionVariations(id, label, name)
```

Bases : django.db.models.base.Model

exception DoesNotExist

Bases : django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases : django.core.exceptions.MultipleObjectsReturned

ND = 'UNDEFINED'

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

label

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

name_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

```
class elgeopaso.jobs.models.Offer(id, created, updated, id_rss, raw_offer, source, title, content, pub_date,
                                   contract, place)
```

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

content

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

contract

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

contract_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
get_next_by_created(*, field=<django.db.models.fields.DateTimeField : created>, is_next=True,
                    **kwargs)
```

```
get_next_by_updated(*, field=<django.db.models.fields.DateTimeField : updated>, is_next=True,
                    **kwargs)
```

```
get_previous_by_created(*, field=<django.db.models.fields.DateTimeField : created>, is_next=False,
                        **kwargs)
```

```
get_previous_by_updated(*, field=<django.db.models.fields.DateTimeField : updated>, is_next=False,
                        **kwargs)
```

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id_rss

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

jobs_positions

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example :

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

Pizza.toppings and Topping.pizzas are ManyToManyDescriptor instances.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

objects = <django.db.models.manager.Manager object>

place

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

place_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

pub_date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

raw_offer

Accessor to the related object on the forward side of a one-to-one relation.

In the example :

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

Restaurant.place is a ForwardOneToOneDescriptor instance.

raw_offer_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property short_content: str

Return the first 300 characters of the offer summary (content).

Renvoie 300 first characters of offer.content

Type renvoyé str

source

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

source_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

technologies

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example :

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

Pizza.toppings and Topping.pizzas are ManyToManyDescriptor instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

updated

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property yearweek: str

Return the week from the publication date (monday as first day). See : <https://docs.python.org/3/library/datetime.html#strptime-and-strptime-behavior>

Renvoie year and week (format : “YYYYWW”)

Type renvoyé str

```
class elgeopaso.jobs.models.Place(name, code, scale, created, updated)
```

Bases : `django.db.models.base.Model`

COUNTRY = 'COUNTRY'

DPT = 'DEPARTEMENT'

exception DoesNotExist

Bases : `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases : `django.core.exceptions.MultipleObjectsReturned`

SCALES = (('DEPARTEMENT', 'Département français'), ('TOM', 'Territoire français'), ('COUNTRY', 'Pays'), ('UNDEFINED', 'Indéfini'))

TOM = 'TOM'

UNDEFINED = 'UNDEFINED'

code

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
get_next_by_created(*, field=<django.db.models.fields.DateTimeField : created>, is_next=True,
                    **kwargs)
```

```
get_next_by_updated(*, field=<django.db.models.fields.DateTimeField : updated>, is_next=True,
                    **kwargs)
```

get_previous_by_created(**field*=<django.db.models.fields.DateTimeField : created>, *is_next*=False, ***kwargs*)

get_previous_by_updated(**field*=<django.db.models.fields.DateTimeField : updated>, *is_next*=False, ***kwargs*)

get_scale_display(**field*=<django.db.models.fields.CharField : scale>)

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

offer_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToManyDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

placevariations_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToManyDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

scale

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

updated

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class elgeopaso.jobs.models.PlaceVariations(*id*, *label*, *name*)

Bases : django.db.models.base.Model

exception DoesNotExist

Bases : django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases : django.core.exceptions.MultipleObjectsReturned

ND = 'UNDEFINED'

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

label

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

name_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

```
class elgeopaso.jobs.models.Source(id, name, url, comment, created, updated)
```

Bases : django.db.models.base.Model

exception DoesNotExist

Bases : django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases : django.core.exceptions.MultipleObjectsReturned

comment

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
get_next_by_created(*, field=<django.db.models.fields.DateTimeField : created>, is_next=True,
                    **kwargs)
```

```
get_next_by_updated(*, field=<django.db.models.fields.DateTimeField : updated>, is_next=True,
                    **kwargs)
```

```
get_previous_by_created(*, field=<django.db.models.fields.DateTimeField : created>, is_next=False,
                        **kwargs)
```

```
get_previous_by_updated(*, field=<django.db.models.fields.DateTimeField : updated>, is_next=False,
                        **kwargs)
```

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

offer_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

updated

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

url

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class elgeopaso.jobs.models.Technology(id, name, license, type_soft, created, updated)
```

```
Bases : django.db.models.base.Model
```

```
DEV = 'LANGUAGE'
```

```
exception DoesNotExist
```

```
Bases : django.core.exceptions.ObjectDoesNotExist
```

```
exception MultipleObjectsReturned
```

```
Bases : django.core.exceptions.MultipleObjectsReturned
```

```
ND = 'UNDEFINED'
```

```
OSS = 'OSS'
```

```
PROPRIETARY = 'PROPRIETARY'
```

```
SOFTWARE = 'SOFTWARE'
```

```
TYPE_LICENSE = (('OSS', 'Libre'), ('PROPRIETARY', 'Propriétaire'), ('UNDEFINED',  
'Indéfini'))
```

```
TYPE_SOFT = (('LANGUAGE', 'Language de programmation'), ('SOFTWARE', 'Logiciel'),  
('UNDEFINED', 'Indéfini'))
```

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
get_license_display(*, field=<django.db.models.fields.CharField : license>)
```

```
get_next_by_created(*, field=<django.db.models.fields.DateTimeField : created>, is_next=True,  
**kwargs)
```

```
get_next_by_updated(*, field=<django.db.models.fields.DateTimeField : updated>, is_next=True,  
**kwargs)
```

```
get_previous_by_created(*, field=<django.db.models.fields.DateTimeField : created>, is_next=False,  
**kwargs)
```

```
get_previous_by_updated(*, field=<django.db.models.fields.DateTimeField : updated>, is_next=False,  
**kwargs)
```

```
get_type_soft_display(*, field=<django.db.models.fields.CharField : type_soft>)
```

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

license

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
objects = <django.db.models.manager.Manager object>
```

offer_set

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.
In the example :

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

technologyvariations_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.
In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

type_soft

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

updated

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class elgeopaso.jobs.models.TechnologyVariations(id, label, name)
```

Bases : `django.db.models.base.Model`

exception DoesNotExist

Bases : `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases : `django.core.exceptions.MultipleObjectsReturned`

ND = 'UNDEFINED'

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

label

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

name_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

elgeopaso.jobs.urls module

Application URLs settings.

Learn more here : <https://docs.djangoproject.com/fr/2.2/topics/http/urls/>

elgeopaso.jobs.views module

Application views.

Learn more here : <https://docs.djangoproject.com/fr/2.2/topics/http/views/>

`elgeopaso.jobs.views.calc_compare_week_by_year()`

Calculate variation in percentage between number of published offers this week and those published the same week last year.

Renvoie variation percentage (can be negative)

Type renvoyé `int`

`elgeopaso.jobs.views.get_contracts_by_technos(request)`

Count offers by contracts types and software passed.

`elgeopaso.jobs.views.get_countries_top5(request)`

Count offers by countries other than France.

`elgeopaso.jobs.views.get_fr_dpts_top10(request)`

Count offers by French departments, including DOM TOM.

`elgeopaso.jobs.views.get_offers_by_period(request)`

Get the number of offers per period (year, month, week). month and week TO DO Called via AJAX

`elgeopaso.jobs.views.get_types_contract_by_period(request)`

Get the number types of contract per period (year, month, week). month and week TO DO Called via AJAX

`elgeopaso.jobs.views.stats_contrats(request)`

Renders statistics by contracts type on contracts page.

`elgeopaso.jobs.views.stats_home(request)`

Displays global metrics about database on homepage.

`elgeopaso.jobs.views.timeline(request)`

Displays 50 latest offers.

elgeopaso.settings package

Submodules

elgeopaso.settings.base module

Base settings to build other settings files upon.

elgeopaso.settings.local module

Settings built upon base for local development.

elgeopaso.settings.production module

Settings built upon base for production.

elgeopaso.settings.test module

Settings built upon base for running tests.

elgeopaso.utils package

Submodules

elgeopaso.utils.custom_stopwords module

elgeopaso.utils.env_loader module

Project utilities.

`elgeopaso.utils.env_loader.find_and_load_environment_vars(start_dir='.')`

Find and load environment files.

Paramètres `start_dir` (*Path*) – folder where to look for env files. Defaults to : « . » - optional

elgeopaso.utils.text_toolbelt module

Tool.

class `elgeopaso.utils.text_toolbelt.TextToolbelt`

Bases : `object`

Tools to manipulate text : tokenize, clean, etc.

classmethod `remove_html_markup(html_text, cleaner='bs-lxml')`

Very basic cleaner for HTML markups.

Paramètres

— `html_text` (*str*) – text to be clean

— `cleaner` (*str*) – Which lib to use to clean the text : - « bs-lxml » : BeautifulSoup4 + LXML - Default. - « psl-only » : Python Standard Library only (html + regex)

Renvoie clean text

Type renvoyé `str`

classmethod `tokenize(input_content)`

Extraction of words mentioned into the offers. The goal is to perform a semantic analysis. Mainly based on NLTK : <https://www.nltk.org/>.

Paramètres `input_content` (*str*) – input text to parse and tokenize

Renvoie list of tokenized words

Type renvoyé `list`

8.1.2 Submodules

elgeopaso.urls module

Project URLs settings.

Learn more here : <https://docs.djangoproject.com/fr/2.2/topics/http/urls/>

To add a new path :

```
# first import the app
import jobs

# then add the new path:
path('jobs/', jobs.urls, name="Jobs offers")
```

elgeopaso.wsgi module

WSGI config for elgeopaso project.

It exposes the WSGI callable as a module-level variable named `application`.

For more information on this file, see <https://docs.djangoproject.com/en/2.1/howto/deployment/wsgi/>

Contributing Guidelines

First off, thanks for considering to contribute to this project !

These are mostly guidelines, not rules. Use your best judgment, and feel free to propose changes to this document in a pull request.

9.1 Git flow

9.1.1 Creating branches

A new branch should be always created from the main branch `master`, except in certain cases which require to be justified.

Naming pattern

The pattern is : `{category}/{slugified-description}`. Where :

- `category` is the type of work. Can be : `feature`, `bug`, `tooling`, `refactor`, `test`, `chore`, `release`, `hotfix`, `docs`, `ci`, `deploy` or `release-candidate`.
- `slugified-description` is the description of the work, slugified.

Example : `feature/improve-encoding`

9.1.2 Merge Requests workflow

Rules

- the code coverage must be increased or equal, never decreased. If you write some new code, write new tests.
- the code must run without any error on the CI.

Using the draft status

A draft Merge Request is a merge request that is not ready to be merged but the code is published to allow other team mates follow the development.

Comments are welcome but they must be global, about the conception, not the details (wait for the WIP status removal).

9.2 Code Style

Make sure your code *roughly* follows [PEP-8](#) and keeps things consistent with the rest of the code. Related tools :

- docstrings : [sphinx-style](#) is used to write technical documentation.
 - formatting : [black](#) is used to automatically format the code without debate.
 - sorted imports : [isort](#) is used to sort imports
 - static analysis (linter) : [flake8](#) is used to catch some dizziness and keep the source code healthy.
-

9.3 Git hooks

We use git hooks through [pre-commit](#) to enforce and automatically check some “rules”. Please install it before to push any commit : `pre-commit install`.

See the relevant configuration file : `.pre-commit-config.yaml`.

10.1 Prérequis

10.1.1 OS

Windows

Version 10 minimum

- activer le sous-système Linux (WSL) :
 - installer Debian ou Ubuntu (16.04 ou 18.04)
 - initialiser l'instance
 - installer les paquets souhaités, par exemple openssl : `sudo apt update && sudo apt upgrade && sudo apt install openssl libssl-dev`
- l'utilisation du nouveau Terminal est fortement recommandée

Linux

Distributions compatibles :

- Debian
- Ubuntu 16.04 ou 18.04

10.1.2 Logiciels

- Docker (Engine et Compose a minima)
 - Python 3.7 64 bits
 - PostgreSQL 12
-

10.2 Lancer en local

Commandes lancées sous Windows 10 avec WSL activé.

10.2.1 Installation

```
# create virtual env
# on Linux: python3.7 -m venv .venv
py -3.7 -m venv .venv

# enter into
.\.venv\Scripts\activate
# on Linux: source .venv/bin/activate

# upgrade install tooling
python -m pip install --upgrade pip

# install requirements
python -m pip install -U -r .\requirements\local.txt
# on Linux: python -m pip install -U -r requirements/local.txt

# download NLTK packages - please refer to `nltk.txt`
python -m nltk.downloader punkt stopwords

# optionally, install pre-commit git-hooks
pre-commit install
```

10.2.2 Configuration

Renommer le fichier `example.env` en `.env` et le compléter. Pour info, il est possible de générer une clé Django Secret en passant par OpenSSL sur WSL : `wsl -- openssl rand -base64 64` (copier/coller dans le fichier `.env`).

10.2.3 Base de données

Initialiser la base de données :

```
# apply migrations to database
python manage.py migrate

# create the super user
python manage.py createsuperuser
```

Pour charger des enregistrements de base (technologies, métiers, types de contrats, etc.), utiliser `loaddata` :

- voir les commandes dans le fichier de déploiement : `.deploy/release-tasks.sh`
- voir “Restaurer les données”

10.2.4 Lancer

```
# launch development web server
python .\manage.py runserver
# on Linux: python manage.py runserver

# alternatively, use the enhanced command from django-extensions
python .\manage.py runserver_plus
```

Ouvrir le navigateur à l’adresse indiquée dans le terminal. Par défaut : `http://localhost:8000/`.

Avec HTTPS

Pour développer au mieux, il est préférable de servir l’application en HTTPS. C’est possible via `runserver_plus` de Django Extensions (voir la documentation).

```
# create folder where to store certificate and key
mkdir certs
# generate SSL certificate and key
wsl -- openssl req -nodes -new -x509 -days 365 -keyout certs/serverKey.key -out certs/
↪ serverCert.cert
# on Linux: remove 'wsl -- '

# alternatively, use the enhanced command from django-extensions
python .\manage.py runserver_plus --cert-file .\certs\serverCert.cert --key-file .\certs\
↪ serverKey.key
```

Ouvrir le navigateur à l’adresse indiquée dans le terminal. Par défaut : `https://localhost:8000/`. Accepter le risque lié aux certificats auto-signés.

10.3 Docker

10.3.1 Prérequis Docker

- Docker 2.2+ ou dans le détail :
 - Docker Engine : 19.03+
 - Docker Compose 1.25+

10.3.2 Configuration Docker

Renommer fichier `example.env` en `docker.env` et compléter :

```
# DEVELOPMENT
DJANGO_DEBUG=1
USE_DOCKER=1

# GLOBAL
DJANGO_ADMIN_URL="admin"
DJANGO_PROJECT_FOLDER="elgeopaso"
DJANGO_SECURE_SSL_REDIRECT=0
DJANGO_SETTINGS_MODULE="elgeopaso.settings.production"
WEB_CONCURRENCY=4

# SECURITY
DJANGO_SECRET_KEY="change_me_with_generated_key"
DJANGO_ALLOWED_HOSTS="localhost, 0.0.0.0, 127.0.0.1"

# EMAIL
REPORT_RECIPIENTS="elpaso@georezo.net,"
SMTP_USER="elpaso@georezo.net"
SMTP_PSWD=

# PostgreSQL
# -----
POSTGRES_HOST=database
POSTGRES_PORT=5432
POSTGRES_DB=elgeopaso-dev
POSTGRES_USER=elgeopaso
POSTGRES_PASSWORD=elgeopaso
DATABASE_URL="postgres://${POSTGRES_USER}:${POSTGRES_PASSWORD}@${POSTGRES_HOST}:${
↪POSTGRES_PORT}/${POSTGRES_DB}"
```

10.3.3 Usage

Lancer l'application

```
docker-compose -f docker-compose.dev.yml up -d
```

Ouvrir le navigateur sur <http://localhost:8000>.

Données et analyses de base

Après que l'application soit lancée :

```
docker-compose -f docker-compose.dev.yml run --rm webapp sh .deploy/release-tasks.sh
```


11.1 Prérequis

Installer les dépendances additionnelles :

```
python -m pip install -U -r requirements/documentation.txt
```

11.2 Générer la documentation

```
sphinx-build -b html docs docs/_build
```

Ouvrir le fichier `docs/_build/index.html` dans un navigateur.

11.3 Rédiger avec un rendu live

```
sphinx-autobuild -b html docs/ docs/_build
```

Ouvrir <http://localhost:8000>.

e

- elgeopaso, 21
- elgeopaso.accounts, 21
- elgeopaso.accounts.admin, 22
- elgeopaso.accounts.apps, 22
- elgeopaso.accounts.migrations, 21
- elgeopaso.accounts.migrations.0001_initial, 21
- elgeopaso.accounts.models, 22
- elgeopaso.accounts.views, 23
- elgeopaso.api, 23
- elgeopaso.api.apps, 23
- elgeopaso.api.serializers, 23
- elgeopaso.api.urls, 24
- elgeopaso.api.views, 24
- elgeopaso.cms, 26
- elgeopaso.cms.admin, 27
- elgeopaso.cms.apps, 28
- elgeopaso.cms.migrations, 26
- elgeopaso.cms.migrations.0001_cms, 26
- elgeopaso.cms.migrations.0002_auto_20180308_1417, 27
- elgeopaso.cms.migrations.0003_auto_20180308_1427, 27
- elgeopaso.cms.migrations.0004_auto_20180628_2354, 27
- elgeopaso.cms.models, 28
- elgeopaso.cms.sitemaps, 30
- elgeopaso.cms.urls, 30
- elgeopaso.cms.views, 31
- elgeopaso.jobs, 31
- elgeopaso.jobs.admin, 40
- elgeopaso.jobs.analyzer, 31
- elgeopaso.jobs.analyzer.georezo, 31
- elgeopaso.jobs.analyzer.georezo.georezo_analyzer, 32
- elgeopaso.jobs.analyzer.georezo.parsers, 31
- elgeopaso.jobs.analyzer.georezo.parsers.content, 31
- elgeopaso.jobs.analyzer.georezo.parsers.title, 32
- elgeopaso.jobs.apps, 44
- elgeopaso.jobs.crawlers, 33
- elgeopaso.jobs.crawlers.georezo_rss_parser, 33
- elgeopaso.jobs.decorators, 44
- elgeopaso.jobs.filters, 44
- elgeopaso.jobs.management, 34
- elgeopaso.jobs.management.commands, 34
- elgeopaso.jobs.management.commands.csv2db, 35
- elgeopaso.jobs.management.commands.map_builder, 35
- elgeopaso.jobs.management.commands.report, 36
- elgeopaso.jobs.management.commands.reset_analisis, 36
- elgeopaso.jobs.management.commands.rss2db, 36
- elgeopaso.jobs.migrations, 37
- elgeopaso.jobs.migrations.0001_initial, 37
- elgeopaso.jobs.migrations.0002_remove_offer_week, 39
- elgeopaso.jobs.migrations.0003_variations_unique, 39
- elgeopaso.jobs.models, 45
- elgeopaso.jobs.template_tags, 39
- elgeopaso.jobs.template_tags.tpl_extras, 39
- elgeopaso.jobs.template_tags.version, 39
- elgeopaso.jobs.urls, 56
- elgeopaso.jobs.views, 56
- elgeopaso.settings, 56
- elgeopaso.settings.base, 56
- elgeopaso.settings.local, 57
- elgeopaso.settings.production, 57
- elgeopaso.settings.test, 57
- elgeopaso.urls, 58
- elgeopaso.utils, 57
- elgeopaso.utils.custom_stopwords, 57
- elgeopaso.utils.env_loader, 57
- elgeopaso.utils.text_toolbelt, 57
- elgeopaso.wsgi, 58

A

- abbrv** (attribut *elgeopaso.jobs.models.Contract*), 45
- about()** (dans le module *elgeopaso.cms.views*), 31
- AccountsConfig** (classe dans *elgeopaso.accounts.apps*), 22
- actions** (attribut *elgeopaso.jobs.admin.GeorezoRSSAdmin*), 41
- add_arguments()** (méthode *elgeopaso.jobs.management.commands.csv2db.Command*), 35
- add_arguments()** (méthode *elgeopaso.jobs.management.commands.reset_analisis*), 36
- add_arguments()** (méthode *elgeopaso.jobs.management.commands.rss2db.Command*), 36
- analisis()** (méthode *elgeopaso.jobs.analyzer.georezo.georezo_analyzer.GeorezoAnalyzer*), 32
- ApiConfig** (classe dans *elgeopaso.api.apps*), 23
- args** (attribut *elgeopaso.jobs.management.commands.csv2db.Command*), 35
- args** (attribut *elgeopaso.jobs.management.commands.csv2db.Command*), 35
- args** (attribut *elgeopaso.jobs.management.commands.reset_analisis.Command*), 36
- args** (attribut *elgeopaso.jobs.management.commands.reset_analisis.Command*), 36
- args** (attribut *elgeopaso.jobs.management.commands.rss2db.Command*), 37
- Article** (classe dans *elgeopaso.cms.models*), 28
- Article.DoesNotExist**, 28
- Article.MultipleObjectsReturned**, 28
- article_set** (attribut *elgeopaso.cms.models.Category*), 30
- ArticleAdmin** (classe dans *elgeopaso.cms.admin*), 27
- ArticleSitemap** (classe dans *elgeopaso.cms.sitemaps*), 30
- author** (attribut *elgeopaso.cms.models.Article*), 28

author_id (*attribut elgeopaso.cms.models.Article*), 28

B

`base_filters` (*attribut elgeo-*
paso.jobs.filters.OfferFilter), 44

`basename` (*attribut elgeo-*
paso.api.views.ContractViewSet), 24

`basename` (*attribut elgeopaso.api.views.JobViewSet*), 25

`basename` (*attribut elgeopaso.api.views.OfferViewSet*),
and), 25

`basename` (*attribut elgeo-*
Command)paso.api.views.PlaceVariationsViewSet),
25

`basename` (*attribut elgeopaso.api.views.PlaceViewSet*),
and), 25

`basename` (*attribut elgeopaso.api.views.TechnoViewSet*),
26

`build_gaonison_fr_departements()` (*méthode elgeo-*
paso.jobs.management.commands.map_builder),
35

`build_gaonison_fr_departements()` (*méthode elgeo-*
Command),

C

base_filters (attribute of `elgeopaso.jobs.filters.OfferFilter`), [44](#)

basename (attribut elgeo-
paso.api.views.ContractViewSet), 24

```

baseline (attribut elgeopaso.api.views.JobViewSet), 25
baseline (attribut elgeopaso.api.views.OfferViewSet),

```

25
basename (attribut elgeo-

`Command(paso.api.views.PlaceVariationsViewSet),`

```

    basename (attribut elgeopaso.api.views.PlaceViewSet),
    and)

```

basename (*attribut elgeopaso.api.views.TechnoViewSet*),

```
builder.gaoisen_fr_departements() (méthode elgeopaso.jobs.management.commands.map builder
```

35
dbCommand)

`calco.compare.week by year()` (dans le module `el-`

geopaso.jobs.views), 56

paso.accounts.admin.SubscriptionInline),

category (attribut `elgeopaso.cms.models.Article`), 28

category (classe dans `elgeopaso.cms.models`), 29

Category.DoesNotExist, 29

Category.MultipleObjectsReturned, 29

category_id (attribut *elgeopaso.cms.models.Article*),
29

CategoryAdmin (classe dans `elgeopaso.cms.admin`), 28

changeFreq (attribut `elgeopaso`), 28

paso.cms.sitemaps.ArticleSitemap), 30

GeoRSS (*classe dans `org.geopaso.models.GeorezoRSS`*), 46

CmsConfig (*classe dans `elgeopaso.cms.apps`*), 28

code (`attribut elgeopaso.jobs.models.Place`), 51

Command	(classe dans <i>elgeopaso.jobs.management.commands.csv2db</i>), 35	create_parser()	(méthode <i>elgeopaso.jobs.management.commands.csv2db.Command</i>), 35
Command	(classe dans <i>elgeopaso.jobs.management.commands.map_builder</i>), 35	create_parser()	(méthode <i>elgeopaso.jobs.management.commands.reset_analisis.Command</i>), 36
Command	(classe dans <i>elgeopaso.jobs.management.commands.report</i>), 36	create_parser()	(méthode <i>elgeopaso.jobs.management.commands.rss2db.Command</i>), 37
Command	(classe dans <i>elgeopaso.jobs.management.commands.reset_analisis</i>), 36	created	(attribut <i>elgeopaso.cms.models.Article</i>), 29
Command	(classe dans <i>elgeopaso.jobs.management.commands.rss2db</i>), 36	created	(attribut <i>elgeopaso.jobs.models.Contract</i>), 45
comment	(attribut <i>elgeopaso.jobs.models.Contract</i>), 45	created	(attribut <i>elgeopaso.jobs.models.GeorezoRSS</i>), 46
comment	(attribut <i>elgeopaso.jobs.models.JobPosition</i>), 47	created	(attribut <i>elgeopaso.jobs.models.JobPosition</i>), 47
comment	(attribut <i>elgeopaso.jobs.models.Source</i>), 53	created	(attribut <i>elgeopaso.jobs.models.Offer</i>), 49
conditional_cache()	(dans le module <i>elgeopaso.jobs.decorators</i>), 44	created	(attribut <i>elgeopaso.jobs.models.Place</i>), 51
content	(attribut <i>elgeopaso.cms.models.Article</i>), 29	created	(attribut <i>elgeopaso.jobs.models.Source</i>), 53
content	(attribut <i>elgeopaso.jobs.models.GeorezoRSS</i>), 46	created	(attribut <i>elgeopaso.jobs.models.Technology</i>), 54
content	(attribut <i>elgeopaso.jobs.models.Offer</i>), 49	D	
ContentParser	(classe dans <i>elgeopaso.jobs.analyzer.georezo.parsers.content</i>), 31	date_hierarchy	(attribut <i>elgeopaso.jobs.admin.GeorezoRSSAdmin</i>), 41
contract	(attribut <i>elgeopaso.jobs.models.Offer</i>), 49	date_hierarchy	(attribut <i>elgeopaso.jobs.admin.OfferAdmin</i>), 42
Contract	(classe dans <i>elgeopaso.jobs.models</i>), 45	declared_filters	(attribut <i>elgeopaso.jobs.filters.OfferFilter</i>), 44
Contract.DoesNotExist	, 45	dependencies	(attribut <i>elgeopaso.accounts.migrations.0001_initial.Migration</i>), 21
Contract.MultipleObjectsReturned	, 45	dependencies	(attribut <i>elgeopaso.cms.migrations.0001_cms.Migration</i>), 26
contract_id	(attribut <i>elgeopaso.jobs.models.Offer</i>), 49	dependencies	(attribut <i>elgeopaso.cms.migrations.0002_auto_20180308_1417.Migration</i>), 27
ContractsAdmin	(classe dans <i>elgeopaso.jobs.admin</i>), 40	dependencies	(attribut <i>elgeopaso.cms.migrations.0003_auto_20180308_1427.Migration</i>), 27
ContractSerializer	(classe dans <i>elgeopaso.api.serializers</i>), 23	dependencies	(attribut <i>elgeopaso.cms.migrations.0004_auto_20180628_2357.Migration</i>), 27
ContractSerializer.Meta	(classe dans <i>elgeopaso.api.serializers</i>), 23	dependencies	(attribut <i>elgeopaso.jobs.migrations.0001_initial.Migration</i>), 37
ContractVariations	(classe dans <i>elgeopaso.jobs.models</i>), 46	dependencies	(attribut <i>elgeopaso.jobs.migrations.0002_remove_offer_week.Migration</i>), 39
ContractVariations.DoesNotExist	, 46	dependencies	(attribut <i>elgeopaso.jobs.migrations.0003_variations_unique.Migration</i>), 39
ContractVariations.MultipleObjectsReturned	, 46	depth	(attribut <i>elgeopaso.api.serializers.OfferSerializer.Meta</i>), 23
contractvariations_set	(attribut <i>elgeopaso.jobs.models.Contract</i>), 45		
ContractVariationsAdmin	(classe dans <i>elgeopaso.jobs.admin</i>), 40		
ContractVariationsInline	(classe dans <i>elgeopaso.jobs.admin</i>), 40		
ContractViewSet	(classe dans <i>elgeopaso.api.views</i>), 24		
COUNTRY	(attribut <i>elgeopaso.jobs.models.Place</i>), 51		
CRAWLER_LATEST_METADATA	(attribut <i>elgeopaso.jobs.crawlers.georezo_rss_parser.GeorezoRssParser</i>), 33		

description (attribut <i>elgeopaso.api.views.ContractViewSet</i>), 24	module, 24
description (attribut <i>elgeopaso.api.views.JobViewSet</i>), 25	<i>elgeopaso.cms</i> module, 26
description (attribut <i>elgeopaso.api.views.OfferViewSet</i>), 25	<i>elgeopaso.cms.admin</i> module, 27
description (attribut <i>elgeopaso.api.views.PlaceVariationsViewSet</i>), 25	<i>elgeopaso.cms.apps</i> module, 28
description (attribut <i>elgeopaso.api.views.PlaceViewSet</i>), 25	<i>elgeopaso.cms.migrations</i> module, 26
description (attribut <i>elgeopaso.api.views.TechnoViewSet</i>), 26	<i>elgeopaso.cms.migrations.0001_cms</i> module, 26
description (attribut <i>elgeopaso.cms.models.Category</i>), 30	<i>elgeopaso.cms.migrations.0002_auto_20180308_1417</i> module, 27
detail (attribut <i>elgeopaso.api.views.ContractViewSet</i>), 24	<i>elgeopaso.cms.migrations.0003_auto_20180308_1427</i> module, 27
detail (attribut <i>elgeopaso.api.views.JobViewSet</i>), 25	<i>elgeopaso.cms.migrations.0004_auto_20180628_2357</i> module, 27
detail (attribut <i>elgeopaso.api.views.OfferViewSet</i>), 25	<i>elgeopaso.cms.models</i> module, 28
detail (attribut <i>elgeopaso.api.views.PlaceVariationsViewSet</i>), 25	<i>elgeopaso.cms.sitemaps</i> module, 30
detail (attribut <i>elgeopaso.api.views.PlaceViewSet</i>), 25	<i>elgeopaso.cms.urls</i> module, 30
detail (attribut <i>elgeopaso.api.views.TechnoViewSet</i>), 26	<i>elgeopaso.cms.views</i> module, 31
DEV (attribut <i>elgeopaso.jobs.models.Technology</i>), 54	<i>elgeopaso.jobs</i> module, 31
docs() (dans le module <i>elgeopaso.cms.views</i>), 31	<i>elgeopaso.jobs.admin</i> module, 40
DPT (attribut <i>elgeopaso.jobs.models.Place</i>), 51	<i>elgeopaso.jobs.analyzer</i> module, 31
E	<i>elgeopaso.jobs.analyzer.georezo</i> module, 31
<i>elgeopaso</i> module, 21	<i>elgeopaso.jobs.analyzer.georezo.georezo_analyzer</i> module, 32
<i>elgeopaso.accounts</i> module, 21	<i>elgeopaso.jobs.analyzer.georezo.parsers</i> module, 31
<i>elgeopaso.accounts.admin</i> module, 22	<i>elgeopaso.jobs.analyzer.georezo.parsers.content</i> module, 31
<i>elgeopaso.accounts.apps</i> module, 22	<i>elgeopaso.jobs.analyzer.georezo.parsers.title</i> module, 32
<i>elgeopaso.accounts.migrations</i> module, 21	<i>elgeopaso.jobs.apps</i> module, 44
<i>elgeopaso.accounts.migrations.0001_initial</i> module, 21	<i>elgeopaso.jobs.crawlers</i> module, 33
<i>elgeopaso.accounts.models</i> module, 22	<i>elgeopaso.jobs.crawlers.georezo_rss_parser</i> module, 33
<i>elgeopaso.accounts.views</i> module, 23	<i>elgeopaso.jobs.decorators</i> module, 44
<i>elgeopaso.api</i> module, 23	<i>elgeopaso.jobs.filters</i> module, 44
<i>elgeopaso.api.apps</i> module, 23	<i>elgeopaso.jobs.management</i> module, 34
<i>elgeopaso.api.serializers</i> module, 23	<i>elgeopaso.jobs.management.commands</i>
<i>elgeopaso.api.urls</i> module, 24	
<i>elgeopaso.api.views</i>	

module, 34	external_downloader() (méthode <i>elgeo-</i>
<i>elgeopaso.jobs.management.commands.csv2db</i>	<i>paso.jobs.management.commands.map_builder.Command</i>),
module, 35	35
<i>elgeopaso.jobs.management.commands.map_builderextract_offer_id_from_url()</i>	(mé-
module, 35	thode de la classe <i>elgeo-</i>
<i>elgeopaso.jobs.management.commands.report</i>	<i>paso.jobs.crawlers.georezo_rss_parser.GeorezoRssParser</i>),
module, 36	33
<i>elgeopaso.jobs.management.commands.reset_analisis</i>	
module, 36	F
<i>elgeopaso.jobs.management.commands.rss2db</i>	FEED_DATETIME_RAW_FORMAT (attribut <i>elgeo-</i>
module, 36	<i>paso.jobs.crawlers.georezo_rss_parser.GeorezoRssParser</i>),
<i>elgeopaso.jobs.migrations</i>	33
module, 37	FEED_DATETIME_RAW_FORMAT_ARROW (attribut <i>elgeo-</i>
<i>elgeopaso.jobs.migrations.0001_initial</i>	<i>paso.jobs.crawlers.georezo_rss_parser.GeorezoRssParser</i>),
module, 37	33
<i>elgeopaso.jobs.migrations.0002_remove_offer_weeks</i>	(attribut <i>elgeo-</i>
module, 39	<i>paso.api.serializers.ContractSerializer.Meta</i>),
<i>elgeopaso.jobs.migrations.0003_variations_unique</i>	23
module, 39	fields (attribut <i>elgeo-</i>
<i>elgeopaso.jobs.models</i>	<i>paso.api.serializers.JobSerializer.Meta</i>),
module, 45	23
<i>elgeopaso.jobs.templatetags</i>	fields (attribut <i>elgeo-</i>
module, 39	<i>paso.api.serializers.OfferSerializer.Meta</i>),
<i>elgeopaso.jobs.templatetags.tpl_extras</i>	23
module, 39	fields (attribut <i>elgeo-</i>
<i>elgeopaso.jobs.templatetags.version</i>	<i>paso.api.serializers.PlaceSerializer.Meta</i>),
module, 39	24
<i>elgeopaso.jobs.urls</i>	fields (attribut <i>elgeo-</i>
module, 56	<i>paso.api.serializers.PlaceVariationsSerializer.Meta</i>),
<i>elgeopaso.jobs.views</i>	24
module, 56	fields (attribut <i>elgeo-</i>
<i>elgeopaso.settings</i>	<i>paso.api.serializers.TechnoSerializer.Meta</i>),
module, 56	24
<i>elgeopaso.settings.base</i>	fields (attribut <i>elgeopaso.jobs.filters.OfferFilter.Meta</i>),
module, 56	44
<i>elgeopaso.settings.local</i>	fieldsets (attribut <i>elgeo-</i>
module, 57	<i>paso.cms.admin.ArticleAdmin</i>), 27
<i>elgeopaso.settings.production</i>	fieldsets (attribut <i>elgeo-</i>
module, 57	<i>paso.jobs.admin.GeorezoRSSAdmin</i>), 41
<i>elgeopaso.settings.test</i>	fieldsets (attribut <i>elgeopaso.jobs.admin.OfferAdmin</i>),
module, 57	42
<i>elgeopaso.urls</i>	find_and_load_environment_vars() (dans le mo-
module, 58	dule <i>elgeopaso.utils.env_loader</i>), 57
<i>elgeopaso.utils</i>	formfield_overrides (attribut <i>elgeo-</i>
module, 57	<i>paso.jobs.admin.GeorezoRSSAdmin</i>), 41
<i>elgeopaso.utils.custom_stopwords</i>	
module, 57	G
<i>elgeopaso.utils.env_loader</i>	GeorezoOfferAnalyzer (classe dans <i>elgeo-</i>
module, 57	<i>paso.jobs.analyzer.georezo.georezo_analyzer</i>),
<i>elgeopaso.utils.text_toolbelt</i>	32
module, 57	GeorezoRSS (classe dans <i>elgeopaso.jobs.models</i>), 46
<i>elgeopaso.wsgi</i>	GeorezoRSS.DoesNotExist, 46
module, 58	GeorezoRSS.MultipleObjectsReturned, 46
<i>ext_url</i> (attribut <i>elgeopaso.cms.models.Article</i>), 29	

GeorezoRSSAdmin (classe dans <i>elgeopaso.jobs.admin</i>), 41		
GeorezoRssParser (classe dans <i>elgeo-paso.jobs.crawlers.georezo_rss_parser</i>), 33		
get_absolute_url() (méthode <i>elgeo-paso.cms.models.Article</i>), 29		
get_absolute_url() (méthode <i>elgeo-paso.cms.models.Category</i>), 30		
get_contracts_by_techos() (dans le module <i>elgeo-paso.jobs.views</i>), 56		
get_countries_top5() (dans le module <i>elgeo-paso.jobs.views</i>), 56		
get_fr_dpts_top10() (dans le module <i>elgeo-paso.jobs.views</i>), 56		
get_license_display() (méthode <i>elgeo-paso.jobs.models.Technology</i>), 54		
get_next_by_created() (méthode <i>elgeo-paso.cms.models.Article</i>), 29		
get_next_by_created() (méthode <i>elgeo-paso.jobs.models.Contract</i>), 45		
get_next_by_created() (méthode <i>elgeo-paso.jobs.models.GeorezoRSS</i>), 47		
get_next_by_created() (méthode <i>elgeo-paso.jobs.models.JobPosition</i>), 47		
get_next_by_created() (méthode <i>elgeo-paso.jobs.models.Offer</i>), 49		
get_next_by_created() (méthode <i>elgeo-paso.jobs.models.Place</i>), 51		
get_next_by_created() (méthode <i>elgeo-paso.jobs.models.Source</i>), 53		
get_next_by_created() (méthode <i>elgeo-paso.jobs.models.Technology</i>), 54		
get_next_by_updated() (méthode <i>elgeo-paso.cms.models.Article</i>), 29		
get_next_by_updated() (méthode <i>elgeo-paso.jobs.models.Contract</i>), 45		
get_next_by_updated() (méthode <i>elgeo-paso.jobs.models.GeorezoRSS</i>), 47		
get_next_by_updated() (méthode <i>elgeo-paso.jobs.models.JobPosition</i>), 47		
get_next_by_updated() (méthode <i>elgeo-paso.jobs.models.Offer</i>), 49		
get_next_by_updated() (méthode <i>elgeo-paso.jobs.models.Place</i>), 51		
get_next_by_updated() (méthode <i>elgeo-paso.jobs.models.Source</i>), 53		
get_next_by_updated() (méthode <i>elgeo-paso.jobs.models.Technology</i>), 54		
get_offers_by_period() (dans le module <i>elgeo-paso.jobs.views</i>), 56		
get_previous_by_created() (méthode <i>elgeo-paso.cms.models.Article</i>), 29		
get_previous_by_created() (méthode <i>elgeo-paso.jobs.models.Contract</i>), 45		
get_previous_by_created() (méthode <i>elgeo-paso.jobs.models.GeorezoRSS</i>), 47		
get_previous_by_created() (méthode <i>elgeo-paso.jobs.models.JobPosition</i>), 48		
get_previous_by_created() (méthode <i>elgeo-paso.jobs.models.Offer</i>), 49		
get_previous_by_created() (méthode <i>elgeo-paso.jobs.models.Place</i>), 51		
get_previous_by_created() (méthode <i>elgeo-paso.jobs.models.Source</i>), 53		
get_previous_by_created() (méthode <i>elgeo-paso.jobs.models.Technology</i>), 54		
get_previous_by_updated() (méthode <i>elgeo-paso.cms.models.Article</i>), 29		
get_previous_by_updated() (méthode <i>elgeo-paso.jobs.models.Contract</i>), 45		
get_previous_by_updated() (méthode <i>elgeo-paso.jobs.models.GeorezoRSS</i>), 47		
get_previous_by_updated() (méthode <i>elgeo-paso.jobs.models.JobPosition</i>), 48		
get_previous_by_updated() (méthode <i>elgeo-paso.jobs.models.Offer</i>), 49		
get_previous_by_updated() (méthode <i>elgeo-paso.jobs.models.Place</i>), 52		
get_previous_by_updated() (méthode <i>elgeo-paso.jobs.models.Source</i>), 53		
get_previous_by_updated() (méthode <i>elgeo-paso.jobs.models.Technology</i>), 54		
get_scale_display() (méthode <i>elgeo-paso.jobs.models.Place</i>), 52		
get_type_soft_display() (méthode <i>elgeo-paso.jobs.models.Technology</i>), 54		
get_types_contract_by_period() (dans le module <i>elgeopaso.jobs.views</i>), 56		
H		
handle() (méthode <i>elgeo-paso.jobs.management.commands.csv2db.Command</i>), 35		
handle() (méthode <i>elgeo-paso.jobs.management.commands.map_builder.Command</i>), 35		
handle() (méthode <i>elgeo-paso.jobs.management.commands.report.Command</i>), 36		
handle() (méthode <i>elgeo-paso.jobs.management.commands.reset_analisis.Command</i>), 36		
handle() (méthode <i>elgeo-paso.jobs.management.commands.rss2db.Command</i>), 37		
help (attribut <i>elgeopaso.jobs.management.commands.csv2db.Command</i>), 35		

[help \(attribut elgeopaso.jobs.management.commands.map_jobs_position_admin \(classe dans elgeopaso.jobs.admin\), 35](#)
[help \(attribut elgeopaso.jobs.management.commands.report_contract_variations \(classe dans elgeopaso.jobs.models\), 36](#)
[help \(attribut elgeopaso.jobs.management.commands.reset_job_position_admin \(classe dans elgeopaso.jobs.admin\), 36](#)
[help \(attribut elgeopaso.jobs.management.commands.rss2db.Command \(classe dans elgeopaso.jobs.admin\), 37](#)
I
[id \(attribut elgeopaso.accounts.models.Subscription\), 22](#)
[id \(attribut elgeopaso.cms.models.Article\), 29](#)
[id \(attribut elgeopaso.cms.models.Category\), 30](#)
[id \(attribut elgeopaso.jobs.models.ContractVariations\), 46](#)
[id \(attribut elgeopaso.jobs.models.JobPositionVariations\), 48](#)
[id \(attribut elgeopaso.jobs.models.Offer\), 49](#)
[id \(attribut elgeopaso.jobs.models.PlaceVariations\), 52](#)
[id \(attribut elgeopaso.jobs.models.Source\), 53](#)
[id \(attribut elgeopaso.jobs.models.Technology\), 54](#)
[id \(attribut elgeopaso.jobs.models.TechnologyVariations\), 55](#)
[id_rss \(attribut elgeopaso.jobs.models.GeorezoRSS\), 47](#)
[id_rss \(attribut elgeopaso.jobs.models.Offer\), 49](#)
[import_georezo_backup\(\) \(méthode elgeopaso.jobs.management.commands.csv2db.Command\), 35](#)
[initial \(attribut elgeopaso.accounts.migrations.0001_initial.Migration\), 21](#)
[initial \(attribut elgeopaso.cms.migrations.0001 cms.Migration\), 26](#)
[initial \(attribut elgeopaso.jobs.migrations.0001_initial.Migration\), 37](#)
[inlines \(attribut elgeopaso.accounts.admin.UserAdmin\), 22](#)
[inlines \(attribut elgeopaso.jobs.admin.ContractsAdmin\), 40](#)
[inlines \(attribut elgeopaso.jobs.admin.JobPositionAdmin\), 41](#)
[inlines \(attribut elgeopaso.jobs.admin.PlaceAdmin\), 42](#)
[inlines \(attribut elgeopaso.jobs.admin.TechnologyAdmin\), 43](#)
[items\(\) \(méthode elgeopaso.cms.sitemaps.ArticleSitemap\), 30](#)
J
[JobPosition \(classe dans elgeopaso.jobs.models\), 47](#)
[JobPosition.DoesNotExist, 47](#)
[JobPosition.MultipleObjectsReturned, 47](#)
[JobPositionAdmin \(classe dans elgeopaso.jobs.admin\), 41](#)
[JobPositionVariations \(classe dans elgeopaso.jobs.models\), 48](#)
[JobPositionVariations.DoesNotExist, 48](#)
[JobPositionVariations.MultipleObjectsReturned, 48](#)
[jobpositionvariations_set \(attribut elgeopaso.jobs.models.JobPosition\), 48](#)
[JobPositionVariationsAdmin \(classe dans elgeopaso.jobs.admin\), 41](#)
[JobPositionVariationsInline \(classe dans elgeopaso.jobs.admin\), 42](#)
[jobs_positions \(attribut elgeopaso.jobs.models.Offer\), 49](#)
[JobsConfig \(classe dans elgeopaso.jobs.apps\), 44](#)
[JobSerializer \(classe dans elgeopaso.api.serializers\), 23](#)
[JobSerializer.Meta \(classe dans elgeopaso.api.serializers\), 23](#)
[JobViewSet \(classe dans elgeopaso.api.views\), 25](#)
L
[label \(attribut elgeopaso.jobs.models.ContractVariations\), 46](#)
[label \(attribut elgeopaso.jobs.models.JobPositionVariations\), 48](#)
[label \(attribut elgeopaso.jobs.models.PlaceVariations\), 52](#)
[label \(attribut elgeopaso.jobs.models.TechnologyVariations\), 55](#)
[lastmod\(\) \(méthode elgeopaso.cms.sitemaps.ArticleSitemap\), 30](#)
[license \(attribut elgeopaso.jobs.models.Technology\), 54](#)
[list_display \(attribut elgeopaso.cms.admin.ArticleAdmin\), 27](#)
[list_display \(attribut elgeopaso.cms.admin.CategoryAdmin\), 28](#)
[list_display \(attribut elgeopaso.jobs.admin.ContractsAdmin\), 41](#)
[list_display \(attribut elgeopaso.jobs.admin.ContractVariationsAdmin\), 40](#)
[list_display \(attribut elgeopaso.jobs.admin.ContractVariationsInline\), 40](#)
[list_display \(attribut elgeopaso.jobs.admin.GeorezoRSSAdmin\), 41](#)
[list_display \(attribut elgeopaso.jobs.admin.JobPositionAdmin\), 41](#)
[list_display \(attribut elgeopaso.jobs.admin.JobPositionVariationsAdmin\), 41](#)

[41](#)
[list_display](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.JobPositionVariationsInline](#)),
[42](#)
[list_display](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.OfferAdmin](#)), [42](#)
[list_display](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.PlaceAdmin](#)), [42](#)
[list_display](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.PlaceVariationsAdmin](#)),
[42](#)
[list_display](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.PlaceVariationsInline](#)),
[43](#)
[list_display](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.SourcesAdmin](#)), [43](#)
[list_display](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.TechnologyAdmin](#)), [43](#)
[list_display](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.TechnologyVariationsAdmin](#)),
[43](#)
[list_display](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.TechnoVariationsInline](#)),
[43](#)
[list_display_links](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.GeorezoRSSAdmin](#)), [41](#)
[list_filter](#) ([attribut](#) [elgeo-](#)
[paso.cms.admin.ArticleAdmin](#)), [27](#)
[list_filter](#) ([attribut](#) [elgeo-](#)
[paso.cms.admin.CategoryAdmin](#)), [28](#)
[list_filter](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.ContractsAdmin](#)), [41](#)
[list_filter](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.ContractVariationsAdmin](#)),
[40](#)
[list_filter](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.GeorezoRSSAdmin](#)), [41](#)
[list_filter](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.JobPositionAdmin](#)), [41](#)
[list_filter](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.JobPositionVariationsAdmin](#)),
[41](#)
[list_filter](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.OfferAdmin](#)), [42](#)
[list_filter](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.PlaceAdmin](#)), [42](#)
[list_filter](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.PlaceVariationsAdmin](#)),
[42](#)
[list_filter](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.SourcesAdmin](#)), [43](#)
[list_filter](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.TechnologyAdmin](#)), [43](#)
[list_filter](#) ([attribut](#) [elgeo-](#)

[paso.jobs.admin.TechnologyVariationsAdmin](#)),
[43](#)
[list_select_related](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.GeorezoRSSAdmin](#)), [41](#)
[list_select_related](#) ([attribut](#) [elgeo-](#)
[paso.jobs.admin.OfferAdmin](#)), [42](#)
[load_previous_crawler_metadata\(\)](#)
(a méthode de la classe [elgeo-](#)
[paso.jobs.crawlers.georezo_rss_parser.GeorezoRssParser](#)),
[33](#)

M

[media](#) ([elgeopaso.accounts.admin.SubscriptionInline](#)
[property](#)), [22](#)
[media](#) ([elgeopaso.accounts.admin.UserAdmin](#) [property](#)),
[22](#)
[media](#) ([elgeopaso.cms.admin.ArticleAdmin](#) [property](#)), [27](#)
[media](#) ([elgeopaso.cms.admin.CategoryAdmin](#) [property](#)),
[28](#)
[media](#) ([elgeopaso.jobs.admin.ContractsAdmin](#) [property](#)),
[41](#)
[media](#) ([elgeopaso.jobs.admin.ContractVariationsAdmin](#)
[property](#)), [40](#)
[media](#) ([elgeopaso.jobs.admin.ContractVariationsInline](#)
[property](#)), [40](#)
[media](#) ([elgeopaso.jobs.admin.GeorezoRSSAdmin](#) [pro-](#)
[perty](#)), [41](#)
[media](#) ([elgeopaso.jobs.admin.JobPositionAdmin](#) [pro-](#)
[perty](#)), [41](#)
[media](#) ([elgeopaso.jobs.admin.JobPositionVariationsAdmin](#)
[property](#)), [41](#)
[media](#) ([elgeopaso.jobs.admin.JobPositionVariationsInline](#)
[property](#)), [42](#)
[media](#) ([elgeopaso.jobs.admin.OfferAdmin](#) [property](#)), [42](#)
[media](#) ([elgeopaso.jobs.admin.PlaceAdmin](#) [property](#)), [42](#)
[media](#) ([elgeopaso.jobs.admin.PlaceVariationsAdmin](#) [pro-](#)
[perty](#)), [42](#)
[media](#) ([elgeopaso.jobs.admin.PlaceVariationsInline](#) [pro-](#)
[perty](#)), [43](#)
[media](#) ([elgeopaso.jobs.admin.SourcesAdmin](#) [property](#)),
[43](#)
[media](#) ([elgeopaso.jobs.admin.TechnologyAdmin](#) [pro-](#)
[perty](#)), [43](#)
[media](#) ([elgeopaso.jobs.admin.TechnologyVariationsAdmin](#)
[property](#)), [43](#)
[media](#) ([elgeopaso.jobs.admin.TechnoVariationsInline](#)
[property](#)), [43](#)
[Migration](#) ([classe](#) [dans](#) [elgeo-](#)
[paso.accounts.migrations.0001_initial](#)), [21](#)
[Migration](#) ([classe](#) [dans](#) [elgeo-](#)
[paso.cms.migrations.0001_cms](#)), [26](#)
[Migration](#) ([classe](#) [dans](#) [elgeo-](#)
[paso.cms.migrations.0002_auto_20180308_1417](#)),
[27](#)

Migration	(classe dans elgeopaso.cms.migrations.0003_auto_20180308_1427),	27	model	(attribut elgeopaso.jobs.admin.TechnologyVariationsAdmin),	43
Migration	(classe dans elgeopaso.cms.migrations.0004_auto_20180628_2357),	27	model	(attribut elgeopaso.jobs.admin.TechnoVariationsInline),	43
Migration	(classe dans elgeopaso.jobs.migrations.0001_initial),	37	module	(attribut elgeopaso.jobs.filters.OfferFilter.Meta),	44
Migration	(classe dans elgeopaso.jobs.migrations.0002_remove_offer_week),	39		elgeopaso, 21	
Migration	(classe dans elgeopaso.jobs.migrations.0003_variations_unique),	39		elgeopaso.accounts, 21	
model	(attribut elgeopaso.accounts.admin.SubscriptionInline),	22		elgeopaso.accounts.admin, 22	
model	(attribut elgeopaso.api.serializers.ContractSerializer.Meta),	23		elgeopaso.accounts.apps, 22	
model	(attribut elgeopaso.api.serializers.JobSerializer.Meta),	23		elgeopaso.accounts.migrations, 21	
model	(attribut elgeopaso.api.serializers.OfferSerializer.Meta),	23		elgeopaso.accounts.migrations.0001_initial,	21
model	(attribut elgeopaso.api.serializers.PlaceSerializer.Meta),	24		elgeopaso.accounts.models, 22	
model	(attribut elgeopaso.api.serializers.PlaceVariationsSerializer.Meta),	24		elgeopaso.accounts.views, 23	
model	(attribut elgeopaso.api.serializers.TechnoSerializer.Meta),	24		elgeopaso.api, 23	
model	(attribut elgeopaso.jobs.admin.ContractVariationsAdmin),	40		elgeopaso.api.apps, 23	
model	(attribut elgeopaso.jobs.admin.ContractVariationsInline),	40		elgeopaso.api.serializers, 23	
model	(attribut elgeopaso.jobs.admin.JobPositionVariationsAdmin),	42		elgeopaso.api.urls, 24	
model	(attribut elgeopaso.jobs.admin.JobPositionVariationsInline),	42		elgeopaso.api.views, 24	
model	(attribut elgeopaso.jobs.admin.PlaceVariationsAdmin),	43		elgeopaso.cms, 26	
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.cms.admin, 27	
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.cms.apps, 28	
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.cms.migrations, 26	
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.cms.migrations.0001_cms, 26	
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.cms.migrations.0002_auto_20180308_1417,	27
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.cms.migrations.0003_auto_20180308_1427,	27
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.cms.migrations.0004_auto_20180628_2357,	27
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.cms.models, 28	
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.cms.sitemaps, 30	
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.cms.urls, 30	
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.cms.views, 31	
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.jobs, 31	
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.jobs.admin, 40	
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.jobs.analyzer, 31	
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.jobs.analyzer.georezo, 31	
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.jobs.analyzer.georezo.georezo_analyzer,	32
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.jobs.analyzer.georezo.parsers,	31
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.jobs.analyzer.georezo.parsers.content,	31
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.jobs.analyzer.georezo.parsers.title,	32
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.jobs.apps, 44	
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.jobs.crawlers, 33	
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.jobs.crawlers.georezo_rss_parser,	33
model	(attribut elgeopaso.jobs.admin.PlaceVariationsInline),	43		elgeopaso.jobs.decorators, 44	

elgeopaso.jobs.filters, 44
 elgeopaso.jobs.management, 34
 elgeopaso.jobs.management.commands, 34
 elgeopaso.jobs.management.commands.csv2db, 35
 elgeopaso.jobs.management.commands.map_builder, 35
 elgeopaso.jobs.management.commands.report, 36
 elgeopaso.jobs.management.commands.reset_name, 36
 elgeopaso.jobs.management.commands.rss2db, 36
 elgeopaso.jobs.migrations, 37
 elgeopaso.jobs.migrations.0001_initial, 37
 elgeopaso.jobs.migrations.0002_remove_offer_week, 39
 elgeopaso.jobs.migrations.0003_variations_unique, 39
 elgeopaso.jobs.models, 45
 elgeopaso.jobs.templatetags, 39
 elgeopaso.jobs.templatetags.tpl_extras, 39
 elgeopaso.jobs.templatetags.version, 39
 elgeopaso.jobs.urls, 56
 elgeopaso.jobs.views, 56
 elgeopaso.settings, 56
 elgeopaso.settings.base, 56
 elgeopaso.settings.local, 57
 elgeopaso.settings.production, 57
 elgeopaso.settings.test, 57
 elgeopaso.urls, 58
 elgeopaso.utils, 57
 elgeopaso.utils.custom_stopwords, 57
 elgeopaso.utils.env_loader, 57
 elgeopaso.utils.text_toolbelt, 57
 elgeopaso.wsgi, 58
 name (attribut elgeopaso.jobs.models.ContractVariations), 46
 name (attribut elgeopaso.jobs.models.JobPosition), 48
 name (attribut elgeopaso.jobs.models.JobPositionVariations), 48
 name (attribut elgeopaso.jobs.models.Place), 52
 name (attribut elgeopaso.jobs.models.PlaceVariations), 52
 name (attribut elgeopaso.jobs.models.Source), 53
 name (attribut elgeopaso.jobs.models.Technology), 54
 name (attribut elgeopaso.jobs.models.TechnologyVariations), 55
 name_id (attribut elgeopaso.jobs.models.ContractVariations), 46
 name_id (attribut elgeopaso.jobs.models.JobPositionVariations), 46
 name_id (attribut elgeopaso.jobs.models.PlaceVariations), 53
 name_id (attribut elgeopaso.jobs.models.TechnologyVariations), 55
 ND (attribut elgeopaso.jobs.models.ContractVariations), 46
 ND (attribut elgeopaso.jobs.models.JobPositionVariations), 48
 ND (attribut elgeopaso.jobs.models.PlaceVariations), 52
 ND (attribut elgeopaso.jobs.models.Technology), 54
 ND (attribut elgeopaso.jobs.models.TechnologyVariations), 55
 now (attribut elgeopaso.jobs.management.commands.report.Command), 36
 now (attribut elgeopaso.jobs.management.commands.rss2db.Command), 37

N

name (attribut elgeopaso.accounts.apps.AccountsConfig), 22
 name (attribut elgeopaso.api.apps.ApiConfig), 23
 name (attribut elgeopaso.api.views.ContractViewSet), 24
 name (attribut elgeopaso.api.views.JobViewSet), 25
 name (attribut elgeopaso.api.views.OfferViewSet), 25
 name (attribut elgeopaso.api.views.PlaceVariationsViewSet), 25
 name (attribut elgeopaso.api.views.PlaceViewSet), 25
 name (attribut elgeopaso.api.views.TechnoViewSet), 26
 name (attribut elgeopaso.cms.apps.CmsConfig), 28
 name (attribut elgeopaso.cms.models.Category), 30
 name (attribut elgeopaso.jobs.apps.JobsConfig), 44
 name (attribut elgeopaso.jobs.models.Contract), 45

O

objects (attribut elgeopaso.accounts.models.Subscription), 22
 objects (attribut elgeopaso.cms.models.Article), 29
 objects (attribut elgeopaso.cms.models.Category), 30
 objects (attribut elgeopaso.jobs.models.Contract), 45
 objects (attribut elgeopaso.jobs.models.ContractVariations), 46
 objects (attribut elgeopaso.jobs.models.GeorezoRSS), 47
 objects (attribut elgeopaso.jobs.models.JobPosition), 48
 objects (attribut elgeopaso.jobs.models.JobPositionVariations), 49
 objects (attribut elgeopaso.jobs.models.Offer), 50
 objects (attribut elgeopaso.jobs.models.Place), 52
 objects (attribut elgeopaso.jobs.models.PlaceVariations), 53

- ul style="list-style-type: none; padding-left: 0;">
- objects (attribut *elgeopaso.jobs.models.Source*), 53
- objects (attribut *elgeopaso.jobs.models.Technology*), 54
- objects (attribut *elgeopaso.jobs.models.TechnologyVariations*), 55
- Offer (classe dans *elgeopaso.jobs.models*), 49
- Offer.DoesNotExist, 49
- Offer.MultipleObjectsReturned, 49
- offer_set (attribut *elgeopaso.jobs.models.Contract*), 45
- offer_set (attribut *elgeopaso.jobs.models.JobPosition*), 48
- offer_set (attribut *elgeopaso.jobs.models.Place*), 52
- offer_set (attribut *elgeopaso.jobs.models.Source*), 53
- offer_set (attribut *elgeopaso.jobs.models.Technology*), 54
- OfferAdmin (classe dans *elgeopaso.jobs.admin*), 42
- OfferFilter (classe dans *elgeopaso.jobs.filters*), 44
- OfferFilter.Meta (classe dans *elgeopaso.jobs.filters*), 44
- offers_to_update() (méthode *elgeopaso.jobs.admin.GeorezoRSSAdmin*), 41
- OfferSerializer (classe dans *elgeopaso.api.serializers*), 23
- OfferSerializer.Meta (classe dans *elgeopaso.api.serializers*), 23
- OfferViewSet (classe dans *elgeopaso.api.views*), 25
- offre_traitee (*elgeopaso.jobs.models.GeorezoRSS* property), 47
- operations (attribut *elgeopaso.accounts.migrations.0001_initial.Migration*), 21
- operations (attribut *elgeopaso.cms.migrations.0001_cms.Migration*), 26
- operations (attribut *elgeopaso.cms.migrations.0002_auto_20180308_1417.Migration*), 27
- operations (attribut *elgeopaso.cms.migrations.0003_auto_20180308_1427.Migration*), 27
- operations (attribut *elgeopaso.cms.migrations.0004_auto_20180628_2357.Migration*), 27
- operations (attribut *elgeopaso.jobs.migrations.0001_initial.Migration*), 37
- operations (attribut *elgeopaso.jobs.migrations.0002_remove_offer_week.Migration*), 39
- operations (attribut *elgeopaso.jobs.migrations.0003_variations_unique.Migration*), 39
- ordering (attribut *elgeopaso.cms.admin.ArticleAdmin*), 27
- ordering (attribut *elgeopaso.cms.admin.CategoryAdmin*), 28
- ordering (attribut *elgeopaso.jobs.admin.ContractsAdmin*), 41
- ordering (attribut *elgeopaso.jobs.admin.OfferAdmin*), 42
- ordering (attribut *elgeopaso.jobs.admin.PlaceAdmin*), 42
- ordering (attribut *elgeopaso.jobs.admin.SourcesAdmin*), 43
- ordering (attribut *elgeopaso.jobs.admin.TechnologyAdmin*), 43
- OSS (attribut *elgeopaso.jobs.models.Technology*), 54
- ## P
- parse_contract_type() (méthode *elgeopaso.jobs.analyzer.georezo.parsers.title.TitleParser*), 32
 - parse_jobs_positions() (méthode *elgeopaso.jobs.analyzer.georezo.parsers.title.TitleParser*), 32
 - parse_new_offers() (méthode *elgeopaso.jobs.crawlers.georezo_rss_parser.GeorezoRssParser*), 33
 - parse_place() (méthode *elgeopaso.jobs.analyzer.georezo.parsers.title.TitleParser*), 32
 - parse_technology() (méthode *elgeopaso.jobs.analyzer.georezo.parsers.content.ContentParser*), 31
 - place (attribut *elgeopaso.jobs.models.Offer*), 50
 - Place (classe dans *elgeopaso.jobs.models*), 51
 - Place.DoesNotExist, 51
 - Place.MultipleObjectsReturned, 51
 - place_id (attribut *elgeopaso.jobs.models.Offer*), 50
 - PlaceAdmin (classe dans *elgeopaso.jobs.admin*), 42
 - PlaceSerializer (classe dans *elgeopaso.api.serializers*), 24
 - PlaceSerializer.Meta (classe dans *elgeopaso.api.serializers*), 24
 - PlaceVariations (classe dans *elgeopaso.jobs.models*), 52
 - PlaceVariations.DoesNotExist, 52
 - PlaceVariations.MultipleObjectsReturned, 52
 - placevariations_set (attribut *elgeopaso.jobs.models.Place*), 52
 - PlaceVariationsAdmin (classe dans *elgeopaso.jobs.admin*), 42
 - PlaceVariationsInline (classe dans *elgeopaso.jobs.admin*), 43
 - PlaceVariationsSerializer (classe dans *elgeopaso.api.serializers*), 24
 - PlaceVariationsSerializer.Meta (classe dans *elgeopaso.api.serializers*), 24

PlaceVariationsViewSet (classe dans *elgeopaso.api.views*), 25
PlaceViewSet (classe dans *elgeopaso.api.views*), 25
prepopulated_fields (attribut *elgeopaso.cms.admin.ArticleAdmin*), 28
prepopulated_fields (attribut *elgeopaso.cms.admin.CategoryAdmin*), 28
priority (attribut *elgeopaso.cms.sitemaps.ArticleSitemap*), 30
PROPRIETARY (attribut *elgeopaso.jobs.models.Technology*), 54
pub_date (attribut *elgeopaso.jobs.models.GeorezoRSS*), 47
pub_date (attribut *elgeopaso.jobs.models.Offer*), 50
published (attribut *elgeopaso.cms.models.Article*), 29

Q

query_transform() (dans le module *elgeopaso.jobs.templatetags.tpl_extras*), 39
queryset (attribut *elgeopaso.api.views.ContractViewSet*), 24
queryset (attribut *elgeopaso.api.views.JobViewSet*), 25
queryset (attribut *elgeopaso.api.views.OfferViewSet*), 25
queryset (attribut *elgeopaso.api.views.PlaceVariationsViewSet*), 25
queryset (attribut *elgeopaso.api.views.PlaceViewSet*), 26
queryset (attribut *elgeopaso.api.views.TechnoViewSet*), 26

R

raw_offer (attribut *elgeopaso.jobs.models.Offer*), 50
raw_offer_id (attribut *elgeopaso.jobs.models.Offer*), 50
readonly_fields (attribut *elgeopaso.jobs.admin.ContractsAdmin*), 41
readonly_fields (attribut *elgeopaso.jobs.admin.GeorezoRSSAdmin*), 41
readonly_fields (attribut *elgeopaso.jobs.admin.JobPositionAdmin*), 41
readonly_fields (attribut *elgeopaso.jobs.admin.OfferAdmin*), 42
readonly_fields (attribut *elgeopaso.jobs.admin.PlaceAdmin*), 42
readonly_fields (attribut *elgeopaso.jobs.admin.SourcesAdmin*), 43
readonly_fields (attribut *elgeopaso.jobs.admin.TechnologyAdmin*), 43
remove_html_markup() (méthode de la classe *elgeopaso.utils.text_toolbelt.TextToolbelt*), 57
replaces (attribut *elgeopaso.cms.migrations.0001_cms.Migration*),

26
report_hour (attribut *elgeopaso.accounts.models.Subscription*), 22
report_week (attribut *elgeopaso.accounts.models.Subscription*), 22

S

save_parsing_metadata() (méthode *elgeopaso.jobs.crawlers.georezo_rss_parser.GeorezoRssParser*), 34
scale (attribut *elgeopaso.jobs.models.Place*), 52
SCALES (attribut *elgeopaso.jobs.models.Place*), 51
search_fields (attribut *elgeopaso.cms.admin.ArticleAdmin*), 28
search_fields (attribut *elgeopaso.cms.admin.CategoryAdmin*), 28
search_fields (attribut *elgeopaso.jobs.admin.ContractsAdmin*), 41
search_fields (attribut *elgeopaso.jobs.admin.ContractVariationsAdmin*), 40
search_fields (attribut *elgeopaso.jobs.admin.GeorezoRSSAdmin*), 41
search_fields (attribut *elgeopaso.jobs.admin.JobPositionAdmin*), 41
search_fields (attribut *elgeopaso.jobs.admin.JobPositionVariationsAdmin*), 42
search_fields (attribut *elgeopaso.jobs.admin.OfferAdmin*), 42
search_fields (attribut *elgeopaso.jobs.admin.PlaceAdmin*), 42
search_fields (attribut *elgeopaso.jobs.admin.PlaceVariationsAdmin*), 43
search_fields (attribut *elgeopaso.jobs.admin.TechnologyAdmin*), 43
search_fields (attribut *elgeopaso.jobs.admin.TechnologyVariationsAdmin*), 43
serializer_class (attribut *elgeopaso.api.views.ContractViewSet*), 24
serializer_class (attribut *elgeopaso.api.views.JobViewSet*), 25
serializer_class (attribut *elgeopaso.api.views.OfferViewSet*), 25
serializer_class (attribut *elgeopaso.api.views.PlaceVariationsViewSet*), 25
serializer_class (attribut *elgeopaso.api.views.PlaceViewSet*), 26
serializer_class (attribut *elgeopaso.api.views.TechnoViewSet*), 26

short_content (*elgeopaso.cms.models.Article* property), 29
short_content (*elgeopaso.jobs.models.GeorezoRSS* property), 47
short_content (*elgeopaso.jobs.models.Offer* property), 50
show_clean_offer() (méthode *elgeopaso.jobs.admin.GeorezoRSSAdmin*), 41
show_raw_offer() (méthode *elgeopaso.jobs.admin.OfferAdmin*), 42
slug_name (attribut *elgeopaso.cms.models.Category*), 30
slug_title (attribut *elgeopaso.cms.models.Article*), 29
SOFTWARE (attribut *elgeopaso.jobs.models.Technology*), 54
source (attribut *elgeopaso.jobs.models.GeorezoRSS*), 47
source (attribut *elgeopaso.jobs.models.Offer*), 50
Source (classe dans *elgeopaso.jobs.models*), 53
Source.DoesNotExist, 53
Source.MultipleObjectsReturned, 53
source_id (attribut *elgeopaso.jobs.models.Offer*), 50
SourcesAdmin (classe dans *elgeopaso.jobs.admin*), 43
stats_contrats() (dans le module *elgeopaso.jobs.views*), 56
stats_home() (dans le module *elgeopaso.jobs.views*), 56
Subscription (classe dans *elgeopaso.accounts.models*), 22
Subscription.DoesNotExist, 22
Subscription.MultipleObjectsReturned, 22
SubscriptionInline (classe dans *elgeopaso.accounts.admin*), 22
suffix (attribut *elgeopaso.api.views.ContractViewSet*), 24
suffix (attribut *elgeopaso.api.views.JobViewSet*), 25
suffix (attribut *elgeopaso.api.views.OfferViewSet*), 25
suffix (attribut *elgeopaso.api.views.PlaceVariationsViewSet*), 25
suffix (attribut *elgeopaso.api.views.PlaceViewSet*), 26
suffix (attribut *elgeopaso.api.views.TechnoViewSet*), 26

T

technologies (attribut *elgeopaso.jobs.models.Offer*), 51
Technology (classe dans *elgeopaso.jobs.models*), 54
Technology.DoesNotExist, 54
Technology.MultipleObjectsReturned, 54
TechnologyAdmin (classe dans *elgeopaso.jobs.admin*), 43
TechnologyVariations (classe dans *elgeopaso.jobs.models*), 55
TechnologyVariations.DoesNotExist, 55
TechnologyVariations.MultipleObjectsReturned, 55

technologyvariations_set (attribut *elgeopaso.jobs.models.Technology*), 55
TechnologyVariationsAdmin (classe dans *elgeopaso.jobs.admin*), 43
TechnoSerializer (classe dans *elgeopaso.api.serializers*), 24
TechnoSerializer.Meta (classe dans *elgeopaso.api.serializers*), 24
TechnoVariationsInline (classe dans *elgeopaso.jobs.admin*), 43
TechnoViewSet (classe dans *elgeopaso.api.views*), 26
TextToolbelt (classe dans *elgeopaso.utils.text_toolbelt*), 57
timeline() (dans le module *elgeopaso.jobs.views*), 56
title (attribut *elgeopaso.cms.models.Article*), 29
title (attribut *elgeopaso.jobs.models.GeorezoRSS*), 47
title (attribut *elgeopaso.jobs.models.Offer*), 51
TitleParser (classe dans *elgeopaso.jobs.analyzer.georezo.parsers.title*), 32
to_update (attribut *elgeopaso.jobs.models.GeorezoRSS*), 47
tokenize() (méthode de la classe *elgeopaso.utils.text_toolbelt.TextToolbelt*), 57
TOM (attribut *elgeopaso.jobs.models.Place*), 51
TYPE_LICENSE (attribut *elgeopaso.jobs.models.Technology*), 54
TYPE_SOFT (attribut *elgeopaso.jobs.models.Technology*), 54
type_soft (attribut *elgeopaso.jobs.models.Technology*), 55

U

UNDEFINED (attribut *elgeopaso.jobs.models.Place*), 51
updated (attribut *elgeopaso.cms.models.Article*), 29
updated (attribut *elgeopaso.jobs.models.Contract*), 45
updated (attribut *elgeopaso.jobs.models.GeorezoRSS*), 47
updated (attribut *elgeopaso.jobs.models.JobPosition*), 48
updated (attribut *elgeopaso.jobs.models.Offer*), 51
updated (attribut *elgeopaso.jobs.models.Place*), 52
updated (attribut *elgeopaso.jobs.models.Source*), 53
updated (attribut *elgeopaso.jobs.models.Technology*), 55
url (attribut *elgeopaso.jobs.models.Source*), 54
url_to_file() (méthode *elgeopaso.jobs.management.commands.map_builder.Command*), 35
user (attribut *elgeopaso.accounts.models.Subscription*), 22
user_id (attribut *elgeopaso.accounts.models.Subscription*), 23
UserAdmin (classe dans *elgeopaso.accounts.admin*), 22

V

`verbose_name` (attribut *elgeo-paso.accounts.apps.AccountsConfig*), [22](#)
`verbose_name` (attribut *elgeo-paso.cms.apps.CmsConfig*), [28](#)
`verbose_name` (attribut *elgeo-paso.jobs.apps.JobsConfig*), [44](#)
`verbose_name_plural` (attribut *elgeo-paso.accounts.admin.SubscriptionInline*), [22](#)
`version_date()` (dans le module *elgeo-paso.jobs.templatetags.version*), [39](#)
`version_number()` (dans le module *elgeo-paso.jobs.templatetags.version*), [40](#)
`view_article()` (dans le module *elgeopaso.cms.views*), [31](#)
`view_category()` (dans le module *elgeo-paso.cms.views*), [31](#)

Y

`yearweek` (*elgeopaso.jobs.models.Offer* property), [51](#)