

---

# **pythonthesorimed Documentation**

***Release***

**Jimmy Girardet**

**Apr 14, 2018**



---

## Contents

---

<b>1</b>	<b>pythonthesorimed</b>	<b>1</b>
1.1	Features . . . . .	1
1.2	Installation . . . . .	1
1.3	Usage . . . . .	1
1.4	Parseur . . . . .	2
1.5	Testing . . . . .	3
1.6	Coverage . . . . .	3
<b>2</b>	<b>Reference de l’API</b>	<b>5</b>
2.1	ThesoItem . . . . .	5
<b>3</b>	<b>Indices and tables</b>	<b>7</b>



Interface python pour thesorimed

- License : GNU General Public License v3
- Documentation: <https://pythonthesorimed.readthedocs.org/en/latest/>
- Source: <https://github.com/jgirardet/pythonthesorimed>

Simple Interface Python pour utiliser l'api thesorimed compatible postgresql uniquement.

## 1.1 Features

- Génère l'API python depuis le fichier thesormed api.sql
- Fournit la class ThesoItem pour les appels à la base

## 1.2 Installation

```
pipenv install pythonthesorimed  
ou  
pip install pythonthesorimed
```

## 1.3 Usage

### 1.3.1 Initialisation

```
from pythonthesorimed.thesoitem import ThesoItem  
session = ThesoItem(host, base, user, password)
```

### 1.3.2 Utilisation des API thesorimed

```
result = session.proc('api_name', param1, param2, param3)
```

Le retour est un Record ou un liste de Record. Record est un objet possédant comme attribut les colonnes associées à la requête

par exemple :

```
>>> session.proc('get_the_gen_equiv', 3, 1)
[Record(gsp_nom='RIFADINE 300MG GELULE', sp_code_equiv=4793, lib_virt=
↳ 'Rifampicine 300 mg gelule', sp_param='RIMACTAN 300MG GELULE', sp_code_sq_pk=3),
  Record(gsp_nom='RIMACTAN 300MG GELULE', sp_code_equiv=3, lib_virt='Rifampicine_
↳ 300 mg gelule', sp_param='RIMACTAN 300MG GELULE', sp_code_sq_pk=3)]

>>> result[0].gsp_nom
'RIFADINE 300MG GELULE'
```

### 1.3.3 Fuzzy Search

Recherche par nom de tous les groupes de spécialités (gsp) et spécialités selon le motif proposé. L'idée est d'épurer un maximum les résultats afin d'éviter tous les doublons dus notamment aux différents laboratoires de génériques. le principe est le suivant :

- Tout d'abord on récupère les gsp correspondant au motif
- Puis on récupère les spécialités qui sont triées la manière suivante:
  - On retire les spécialités appartenant à des gsp déjà sélectionnés.
  - Pour les spécialités restantes ayant un gsp commun, une seule est conservée
  - Les spécialités sans gsp sont conservées par défaut pour ne pas risquer d'avoir un manque.
- Le tout est retournée en une seule liste, d'abord les gsp puis les spécialités restantes. Le contenu de la liste est des dict au format : nom, cip, code, gsp\_code.

Les différents éléments de la requête sont séparés par un espace et doivent être l'ordre de l'appellation officielle. Par exemple : "codéine paracétamol" ne retournera pas "paracétamol + codéine".

La recherche doit s'effectuer en minuscule.

```
session.fuzzy('chaîne')
```

## 1.4 Parseur

Cloner le dépôt puis :

```
make dev
pr python build.py path/to/api.sql
```

api.py est maintenant dans pythonthesorimed

## 1.5 Testing

`make test` : toutes les fonctionnalités ne nécessitant pas le base de donnée thesorimed.

`make test-local` : tests avec des appels à thesorimed.

`make all` : tout.

Travis utilise la version “`make test`” d’où le coverage plus faible

## 1.6 Coverage

`make test-coverage`

`make test-coverage-all`





## CHAPTER 2

---

Reference de l'API

---

### 2.1 Thesoltem



## CHAPTER 3

---

### Indices and tables

---

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