

Leafium

1 Index des espaces de nommage	1
1 Index des espaces de nommage	1
1.1 Liste des espaces de nommage	1
2 Index des classes	2
2.1 Liste des classes	2
3 Index des fichiers	2
3.1 Liste des fichiers	2
4 Documentation des espaces de nommage	2
4.1 Référence de l'espace de nommage data	2
4.1.1 Documentation des variables	2
4.2 Référence de l'espace de nommage database	4
4.3 Référence de l'espace de nommage main	4
4.3.1 Documentation des fonctions	4
4.3.2 Documentation des variables	4
5 Documentation des classes	5
5.1 Référence de la classe database.Database	5
5.1.1 Documentation des constructeurs et destructeur	6
5.1.2 Documentation des fonctions membres	6
6 Documentation des fichiers	7
6.1 Référence du fichier /Users/maximecolliat/Documents/Projet/Leafium/src/data.py	7
6.2 Référence du fichier /Users/maximecolliat/Documents/Projet/Leafium/src/database.py	7
6.3 Référence du fichier /Users/maximecolliat/Documents/Projet/Leafium/src/main.py	7

1 Index des espaces de nommage

1.1 Liste des espaces de nommage

Liste de tous les espaces de nommage avec une brève description:

data	2
database	4
main	4

2 Index des classes

2.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

3 Index des fichiers

3.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

<code>/Users/maximecolliat/Documents/Projet/Leafium/src/data.py</code>	7
<code>/Users/maximecolliat/Documents/Projet/Leafium/src/database.py</code>	7
<code>/Users/maximecolliat/Documents/Projet/Leafium/src/main.py</code>	7

4 Documentation des espaces de nommage

4.1 Référence de l'espace de nommage data

Variables

- list `genres_de_films`
- list `realisateurs_de_films`
- list `users`
- list `titres_de_films`
- list `commentaires_et_notes_de_films`
- list `cinemas`

4.1.1 Documentation des variables

4.1.1.1 cinemas `list data.cinemas`

4.1.1.2 commentaires_et_notes_de_films `list data.commentaires_et_notes_de_films`

4.1.1.3 genres_de_films list data.genres_de_films

Valeur initiale :

```
00001 = ["Action", "Comédie", "Drame", "Science-fiction", "Horreur", "Documentaire", "Animation",
00002         "Romance", "Thriller", "Fantasy", "Aventure", "Mystère", "Crime", "Familial",
        "Guerre", "Histoire"]
```

4.1.1.4 realisateurs_de_films list data.realisateurs_de_films

Valeur initiale :

```
00001 = ["Christopher Nolan", "Quentin Tarantino", "Steven Spielberg", "Martin Scorsese",
00002         "Alfred Hitchcock", "Francis Ford Coppola", "Stanley Kubrick", "David
        Fincher",
00003         "Sofia Coppola", "Greta Gerwig", "James Cameron", "George Lucas", "Peter
        Jackson",
00004         "Tim Burton", "Ridley Scott", "Clint Eastwood", "Woody Allen", "Wes
        Anderson", "Spike Lee",
00005         "Sergio Leone",
00006         "Hayao Miyazaki", "Jean-Luc Godard", "Akira Kurosawa", "David Lynch", "Ingmar
        Bergman",
00007         "Federico Fellini",
00008         "Charlie Chaplin", "Roman Polanski", "Quentin Tarantino", "Stanley Kubrick",
00009         "Alfred Hitchcock",
00010         "Martin Scorsese", "Steven Spielberg", "Christopher Nolan", "David Fincher",
00011         "Francis Ford Coppola",
00012         "Peter Jackson", "Woody Allen", "Tim Burton", "Clint Eastwood", "James
        Cameron",
00013         "Coen Brothers"
00014     ]
```

4.1.1.5 titres_de_films list data.titres_de_films

4.1.1.6 users list data.users

Valeur initiale :

```
00001 = [
00002     "Jean", "Pierre", "Paul", "Jacques", "Marie", "Julie", "Sophie", "Alice", "Lucie", "Emma",
    "Louise", "Léa", "Chloé",
00003     "Léo", "Louis", "Hugo", "Arthur", "Raphaël", "Ethan", "Maël", "Gabriel", "Jules", "Adam", "Lucas",
    "Noah", "Liam",
00004     "Emma", "Olivia", "Ava", "Isabella", "Sophia", "Charlotte", "Mia", "Amelia", "Harper", "Evelyn",
    "Abigail", "Emily",
00005     "James", "John", "Robert", "Michael", "William", "David", "Richard", "Joseph", "Thomas",
    "Charles", "Christopher",
00006     "Mary", "Patricia", "Jennifer", "Linda", "Elizabeth", "Barbara", "Susan", "Jessica", "Sarah",
    "Karen", "Nancy",
00007     "Mohamed", "Adam", "Liam", "Noah", "Ethan", "Lucas", "Mason", "Logan", "Oliver", "James",
    "Alexander", "Elijah",
00008     "Ava", "Sophia", "Isabella", "Mia", "Charlotte", "Abigail", "Emily", "Harper", "Amelia", "Evelyn",
    "Elizabeth",
00009     "Alice", "Bob", "Charlie", "David", "Eva", "Frank", "Grace", "Hector", "Isabel", "Jack",
    "Katherine", "Liam", "Mia",
00010     "Nathan", "Olivia", "Pablo", "Quinn", "Rachel", "Samuel", "Taylor", "Ulysses", "Victoria",
    "William", "Xavier",
00011     "Yvonne", "Zachary", "Rahman", "Sofia", "Tariq", "Umar", "Vivian", "Walter", "Xena", "Yasmine",
    "Zara"
00012 ]
```

4.2 Référence de l'espace de nommage database

Classes

— class [Database](#)

4.3 Référence de l'espace de nommage main

Fonctions

— def `connection` ()

Variables

— `try` :

— def `db` = `connection`()

4.3.1 Documentation des fonctions

4.3.1.1 `connection()` `def main.connection ()`

Connection à la base de données MongoDB en local ou sur le serveur de l'IEM
:return:

4.3.2 Documentation des variables

4.3.2.1 `db` `def main.db = connection()`

4.3.2.2 `try` `main.try :`

5 Documentation des classes

5.1 Référence de la classe `database.Database`

Fonctions membres publiques

— def `__init__` (self, str `host`, int `port`, str `database`, str `user`=None, str `password`=None)

— def `create_collection_cinemas` (self)

— def `create_collection_films` (self)

— def `drop_collection` (self)

- def `populate` (self, int film)
- def `clear` (self)
- def `get` (self, str collection, dict query=None, dict projection=None)

FONCTIONS DE BASE #.

- def `get_one` (self, str collection, dict query=None, dict projection=None)
- def `get_films` (self, dict query=None, dict projection=None)
- def `get_cinemas` (self, dict query=None, dict projection=None)
- def `get_film` (self, dict query=None, dict projection=None)
- def `get_cinema` (self, dict query=None, dict projection=None)
- def `insert` (self, str collection, dict document)
- def `insert_many` (self, str collection, list documents)
- def `update` (self, str collection, dict query, dict update)
- def `insert_film` (self, dict film)
- def `insert_films` (self, list films)
- def `insert_cinema` (self, dict cinema)
- def `insert_cinemas` (self, list `cinemas`)
- def `get_average_rating_by_movie` (self, int limit)

FONCTIONS DE REQUÊTES #.

- def `get_movie_by_category` (self, str category)
- def `get_movie_by_director` (self, str director)
- def `get_total_tickets_sold` (self)
- def `get_top_movies_by_tickets_sold` (self, int limit)
- def `get_top_cinema_by_tickets_sold` (self, int limit)
- def `get_movie_name_under_price` (self, int price, int limit)
- def `delete_films_with_director` (self, str director)

Attributs publics

- `host`
 - `port`
 - `database`
 - `user`
 - `password`
-

— [db](#)

5.1.1 Documentation des constructeurs et destructeur

5.1.1.1 `__init__()` `def database.Database.__init__ (`
 `self,`
 `str host,`
 `int port,`
 `str database,`
 `str user = None,`
 `str password = None)`

5.1.2 Documentation des fonctions membres

5.1.2.1 `clear()` `def database.Database.clear (`
 `self)`

Permet de vider la base de données.
:param self:
:return: None

5.1.2.2 `create_collection_cinemas()` `def database.Database.create_collection_cinemas (`
 `self)`

Crée la collection cinemas avec un schéma de validation.
:param self:
:return: None

Voici le graphe des appelants de cette fonction :

6 Documentation des fichiers

6.1 Référence du fichier `/Users/maximecolliat/Documents/Projet/Leafium/src/data.py`

Espaces de nommage

— namespace [data](#)

Variables

- list `data.genres_de_films`
- list `data.realisateurs_de_films`
- list `data.users`
- list `data.titres_de_films`
- list `data.commentaires_et_notes_de_films`
- list `data.cinemas`

6.2 Référence du fichier /Users/maximecolliat/Documents/Projet/Leafium/src/database.py

Classes

- class `database.Database`

Espaces de nommage

- namespace `database`

6.3 Référence du fichier /Users/maximecolliat/Documents/Projet/Leafium/src/main.py

Espaces de nommage

- namespace `main`

Fonctions

- def `main.connection ()`

Variables

- `main.try :`
 - def `main.db = connection()`
-

