

---

# **aeneria Documentation**

**Simon Mellerin**

**sept. 18, 2020**



---

Contenu :

---

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Installer æneria facilement avec YunoHost !</b> | <b>3</b>  |
| 1.1      | Installation . . . . .                             | 3         |
| 1.2      | Documentation utilisateur . . . . .                | 6         |
| 1.3      | Documentation administrateur . . . . .             | 17        |
| 1.4      | Documentation développeur . . . . .                | 20        |
| <b>2</b> | <b>Todo List</b>                                   | <b>25</b> |
| <b>3</b> | <b>Support</b>                                     | <b>27</b> |
|          | <b>Index</b>                                       | <b>29</b> |

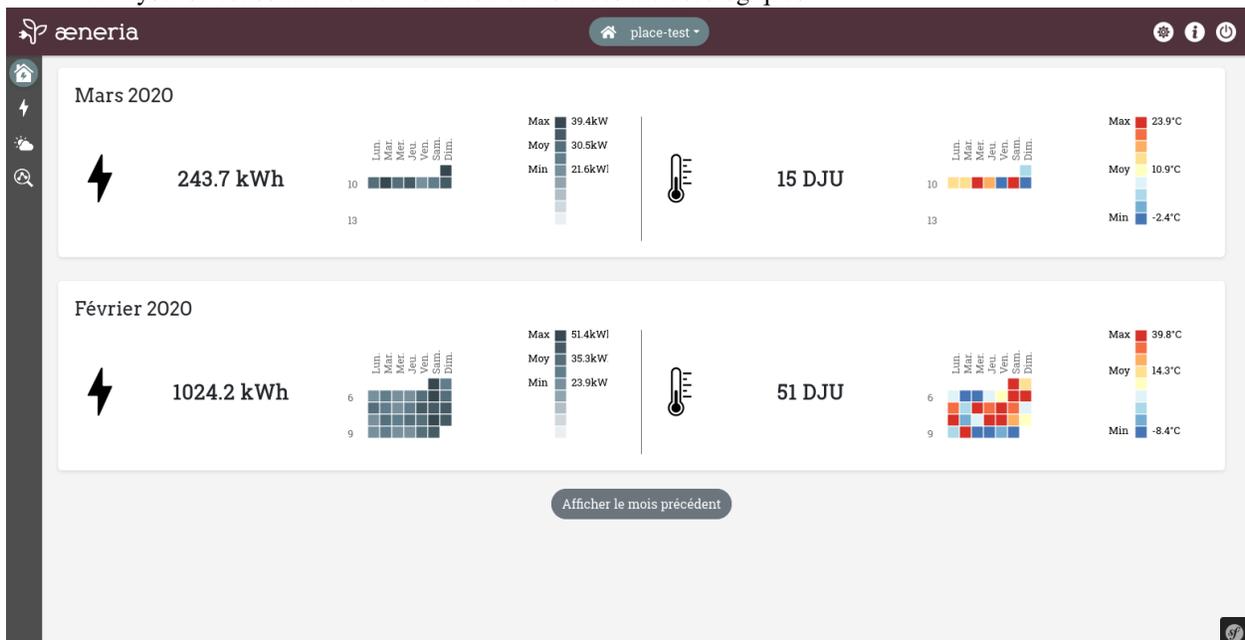




aeneria est un petit tableau de bord permettant d'analyser sa consommation d'électricité à partir des données d'un Linky.

L'idée derrière aeneria est de permettre à l'utilisateur de pouvoir, au travers de quelques tableaux de bord :

- Mieux comprendre sa consommation d'énergie
- Analyser sa consommation en fonction de données météorologiques





---

Installer æneria facilement avec YunoHost!

---



## 1.1 Installation

Tout d'abord pour utiliser æneria,

- Vous devez avoir accès à un Linky et à un [compte Enedis](#)
- Via ce compte, vous devez activer l'option *Courbe de charge* pour pouvoir avoir accès à votre consommation horaire

### 1.1.1 Installation via YunoHost

YunoHost est un projet ayant pour but de promouvoir l'autohébergement. Son but est de faciliter l'administration d'un serveur : [en savoir plus](#)



Des nombreuses applications sont déjà packagées pour être utilisées avec et c'est le cas de aeneria.



### 1.1.2 Installation à la main

aeneria est une application basée sur le framework Symfony. Elle s'installe sur un serveur web disposant d'un PHP récent et d'un serveur de base de données MySQL.

#### Prérequis

- PHP 7.3 et supérieur
- PostgreSQL (9.6 et supérieur)

---

**Note :** MySQL et SQLite devraient fonctionner mais vous aurez à adapter les fichiers `.env` & `config/packages/doctrine.yaml`

Il n'est pas prévu que aeneria Les supporte *officiellement*.

---

**Avertissement :** Les migrations de aeneria sont uniquement générées pour PostgreSQL, si vous utilisez un autre type de serveur, gardez à l'esprit qu'il faudra vérifier chaque migration avant de la lancer !

#### Installation

Retrouvez les différentes versions d'aeneria sur son dépôt Gitlab sur [la page des Releases](#).

Les différentes versions accompagnées de leurs dépendances Composer et des assets compilés sont disponibles sur le dépôt d'aeneria

### 1. Récupérer les sources

Téléchargez et décompressez le dernière version au format tar.gz :

```
wget http://statics.aeneria.com/aeneria-app-latest.tar.gz
tar -xvzf aeneria-app-latest.tar.gz aeneria-app
cd aeneria-app
```

### 2. Créer et renseigner la base de données

Créez une base de données.

Copiez le fichier `.env.dist` puis adaptez-le :

```
cp .env.dist .env
```

```
# fichier .env

...

###> doctrine/doctrine-bundle ###
# Format described at http://docs.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/configuration.html#connecting-using-a-url
# For PostgreSQL database use: "pgsql://[database_user]:[database_password]@127.0.0.1:5432/[database_name]"
# For MySQL database use: "mysql://[database_user]:[database_password]@127.0.0.1:3306/[database_name]"
# For an SQLite database, use: "sqlite:///kernel.project_dir%/var/data.db"
# Configure your db driver and server_version in config/packages/doctrine.yaml
DATABASE_URL=[VOTRE CONFIG ICI]
###< doctrine/doctrine-bundle ###

...
```

Adaptez également le fichier `config/packages/doctrine.yaml` si votre serveur de base de données n'est pas PostgreSQL :

```
# fichier config/packages/doctrine.yaml

...

# Renseigner ici les info de votre dbal
doctrine:
  dbal:
    # Configure these for your database server

    # Mysql
    # driver: 'pdo_mysql'
    # server_version: '5.2'
    # charset: utf8mb4
    # default_table_options:
    #   charset: utf8mb4
    #   collate: utf8mb4_unicode_ci

    # PostgreSQL
    driver: 'pdo_pgsql'
    server_version: '9.6'
    charset: utf8

    #SQLite
    # driver: pdo_sqlite
    # charset: utf8

...
```

### 3. Générer la base de données

Lancez le commande d'installation d'aeneria :

```
php7.3 bin/console aeneria:install
```

### 4. Créer un administrateur

Ajoutez un premier utilisateur et donnez-lui les droits administrateur :

```
php7.3 bin/console aeneria:user:add [username] [password]
php7.3 bin/console aeneria:user:grant [username]
```

### 5. Générer l'ensemble des flux Météo (facultatif)

Si vous le souhaitez, vous pouvez créer l'ensemble des flux météo pour l'utilisateur admin. L'intérêt est de commencer à stocker toutes les données météo dès l'installation de l'instance. Un utilisateur qui crée son compte dans le futur aura directement accès à l'ensemble de données météos depuis l'installation d'aeneria. Par contre, en faisant ça, l'ensemble des données des 62 stations Météo sera historisé, ce qui augmente la taille de la base de données.

Pour ça, lancer la commande suivante :

```
php7.3 bin/console aeneria:feed:meteo:generate-all [username]
```

---

**Note :** Les données Météo étant dans données publiques, il n'y a pour elles pas de problème de confidentialité. Pour simplifier les traitements, les données des flux météo ne sont jamais supprimés. Si vous souhaitez quand même les supprimer, vous pouvez le faire en utilisant la commande `aeneria :feed :clean-orphans`

---

### 6. Mettre en place le CRON

Mettez en place le CRON en exécutant la commande suivante :

```
echo "*/10 * * * * [user] php7.3 /[app_folder]/bin/console aeneria:fetch-data_
↪false" > /etc/cron.d/aeneria
# où [user] est l'utilisateur linux qui lancera le cron
```

### 7. Configurer le serveur web

Enfin, configurez [NGINX](#) ou [Apache](#) comme pour une application Symfony 5 classique

## 1.2 Documentation utilisateur

### 1.2.1 Les différents tableaux de bords

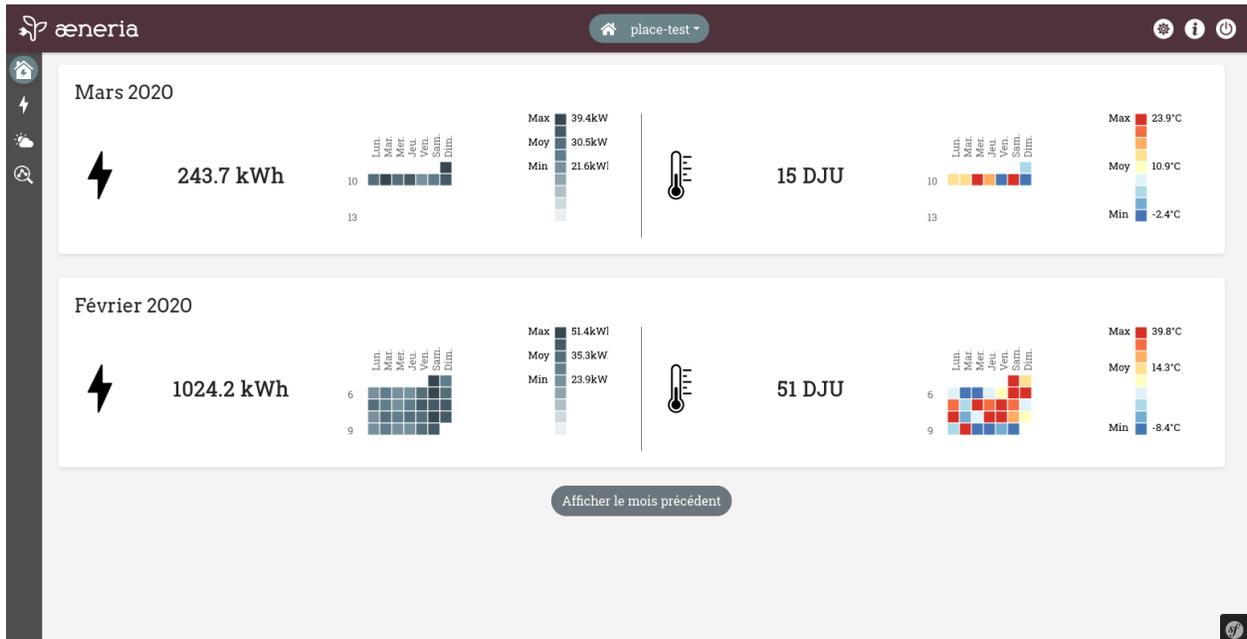
Cette page décrit les différents tableaux de bord disponible sur aeneria.



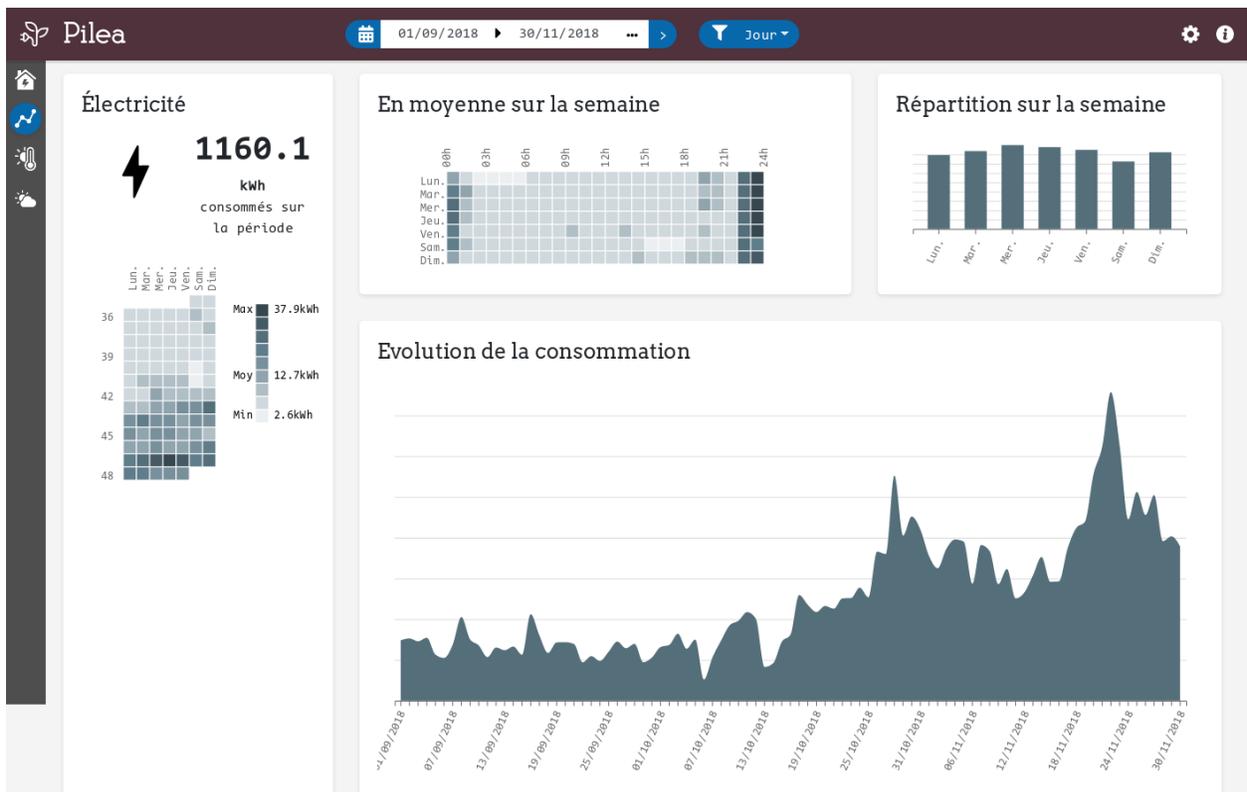
L'accueil

Lors de la connexion à aeneria, vous atterissez sur la page d'accueil.

Cette page contient une première carte résumant les données du mois en cours et une seconde résumant les données du mois précédent.



**⚡ Consommation d'électricité**



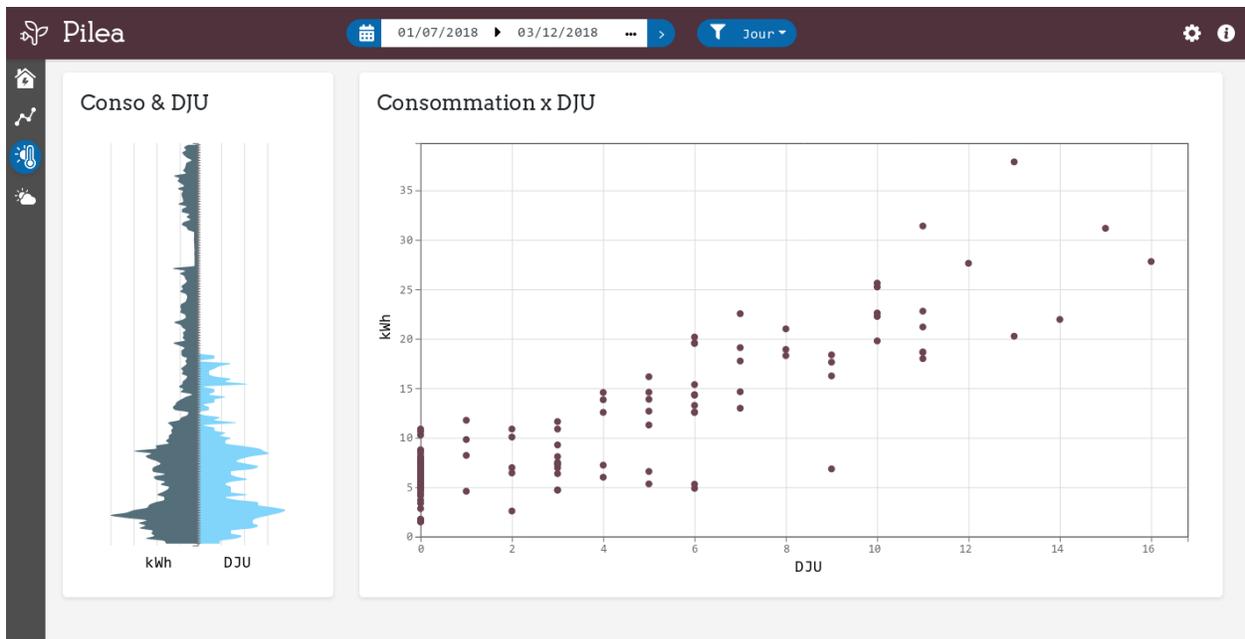


## Analyse de la météo

```
utilisateur/./img/dash_meteo.png
```



## Analyse croisée



### 1.2.2 Comprendre les différents graphiques

Cette page recense les différents types de graphique présents dans æneria et apporte quelques éléments pour commencer à les analyser.

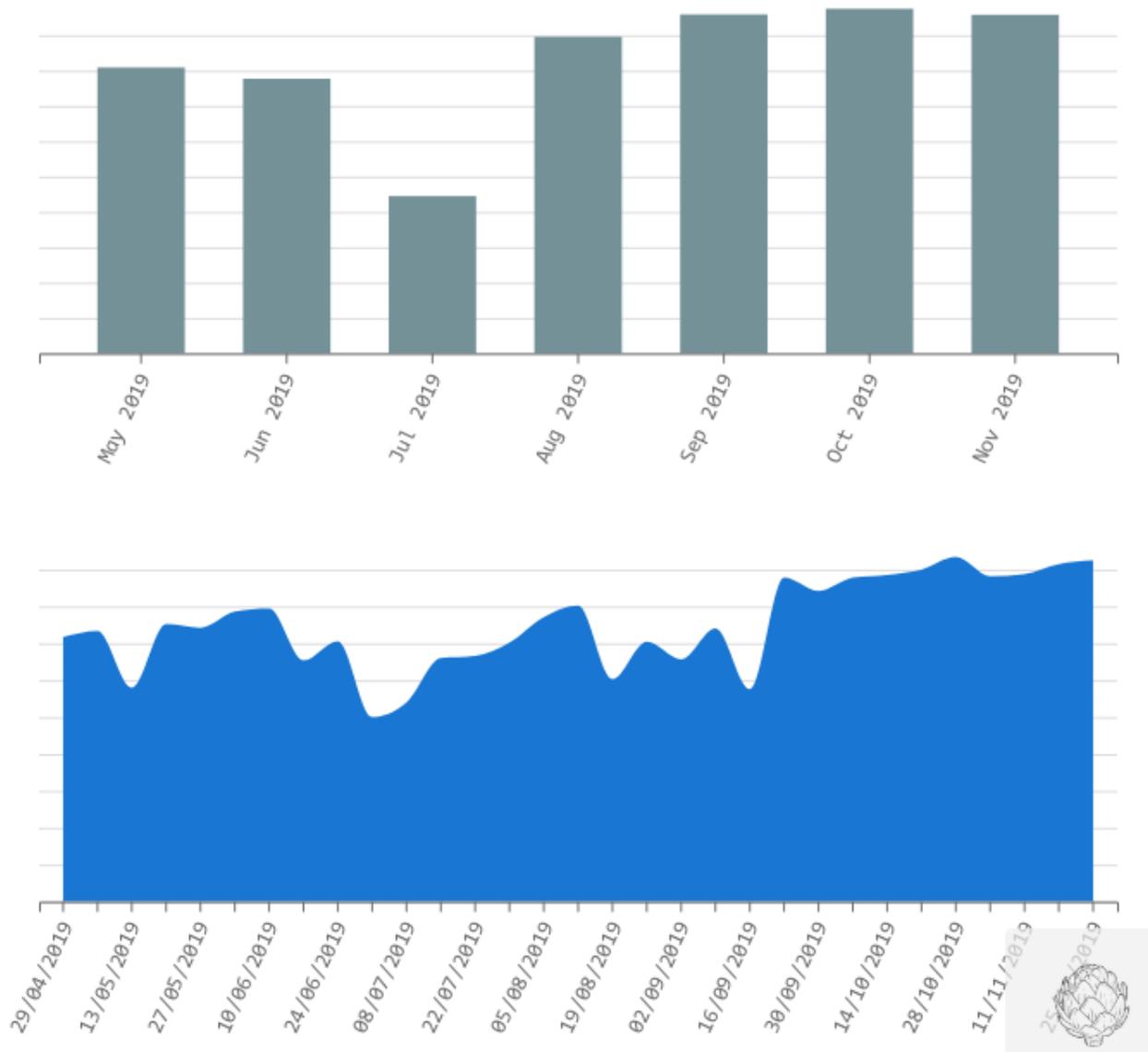
#### Graphique d'évolution temporelle

Le graphique d'évolution temporelle est un graphique assez classique.

Il affiche, pour la période sélectionnée, un point de donnée en fonction de la granulosité choisie.

Il présente chaque point de donnée de donnée sous forme de rectangle, il se transforme en courbe si le nombre de données à afficher est trop élevé.

## Aperçus



## Comment lire le graphique

Ce graphique permet par exemple de relever des consommations anormales :

J'observe ma consommation d'électricité, pour un jour donnée je vois un pic sur le graphique, est-ce normal ? Que s'est-il passé ce jour ?

- Etait-ce un retour de vacances ? Dans ce cas le pic est peut-être normal (reprise de chauffage par exemple)
- Ou bien ai-je oublié d'éteindre un appareil gourmand avant d'aller me coucher ce jour-là ?

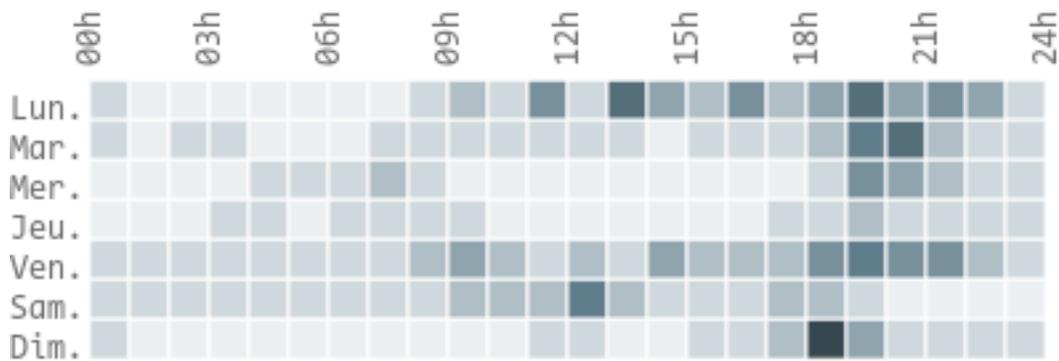
### Carte de chaleur Jours/Heures

Chaque case de la carte de chaleur Jours/Heures affiche la moyenne, sur la période sélectionnée, de la grandeur observée pour une tranche horaire.

Par exemple, la case en haut à gauche représente la consommation moyenne le lundi entre minuit et 1h sur la période sélectionnée.

Plus une case est foncée et plus la consommation est élevée.

### Aperçus



### Comment lire le graphique

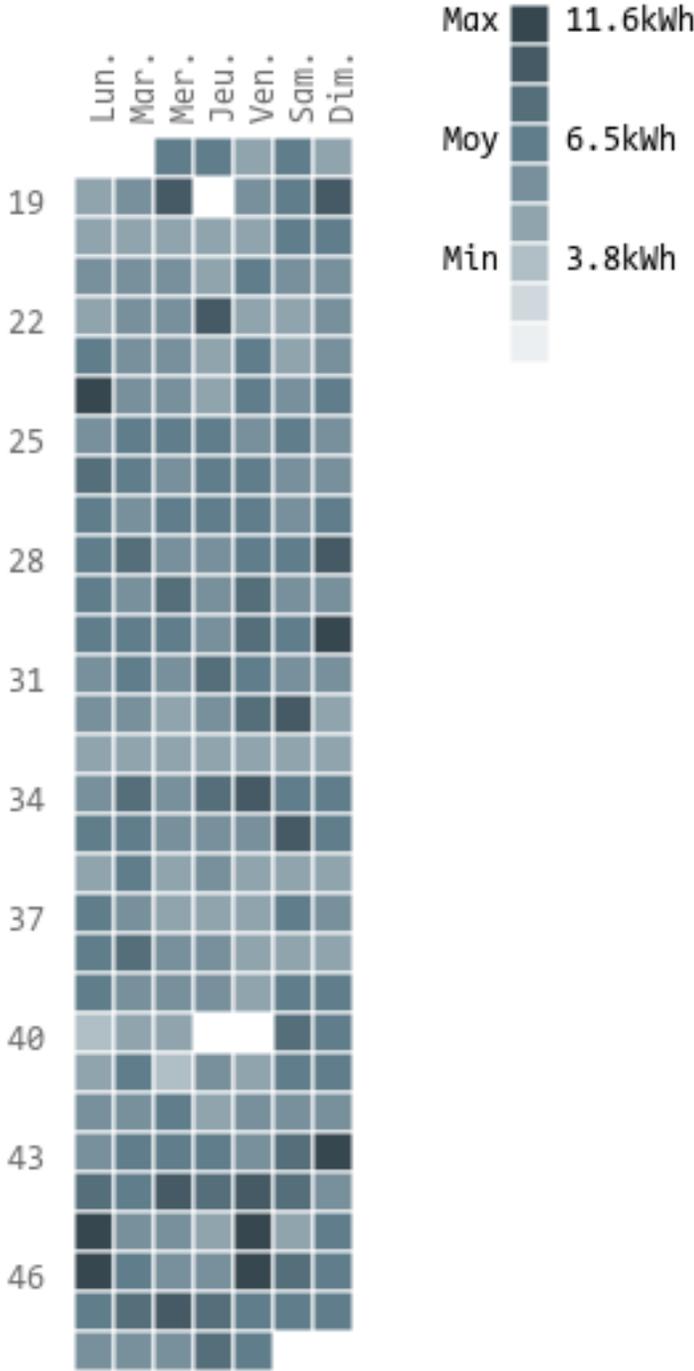
Ce graphique permet de mettre en valeur des schémas de consommation. Les tâches observées doivent refléter vos habitudes de consommation.

- Une tâche foncée apparaît les jours de semaine entre 7h30 et 9h, est-ce normal ?
- Je ne vois aucun schéma se dégager, est-ce normal que ma consommation soit la même quelque soit l'heure et le jour de la semaine ?
- Il y a une tâche continue toute la journée du jeudi, mais pas les autres jours, n'y aurait-il pas un problème dans la programmation de mon chauffage ?

### Carte de chaleur Jours/Semaines

Sur ce graphique, chaque case représente un jour, chaque ligne une semaine et chaque colonne un jour de la semaine. Plus la case est foncée est plus la grandeur observée est grande.

Aperçus



### Comment lire le graphique

Comme la carte de chaleur Jours/Heures, ce graphique permet de mettre en évidence des schémas de consommation. Par exemple : En hiver, j'observe que les 2 dernières colonnes sont plus foncée, c'est peut-être normal si l'on chauffe toute la journée le weekend contrairement aux jours de semaine.

Ensuite, comme le graphique d'évolution, il permet de repérer des pics de consommation : pourquoi cette case est-elle foncée au milieu de l'été ? Que s'est-il passé ce jour là ?

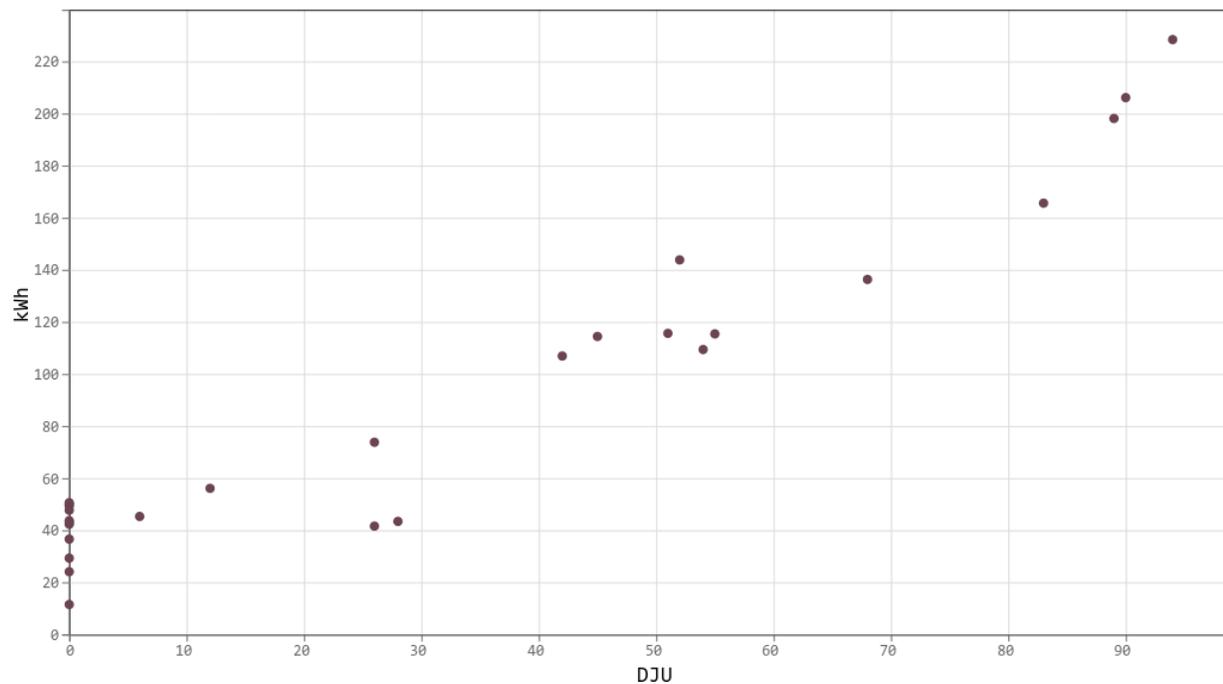
Cette représentation permet aussi d'observer des événements tels que les vacances : je suis parti une semaine en février, pourtant, je ne le vois pas sur le graphique ? Ai-je éteint le chauffage avant de partir ?

Afficher sur une ou plusieurs années, il permet aussi de voir apparaître des saisonnalités (notamment sur le tableau de bords météo)

### Graphique d'analyse croisée

L'analyse croisée permet de chercher des corrélations entre 2 grandeurs. Pour la période sélectionnée, il affiche un point selon la granulosité choisit.

### Aperçus



### Comment lire le graphique

Si on prend le cas où l'on observe la consommation d'électricité et les *DJU* et si l'on considère que le chauffage est électrique. Si l'on sélectionne une période d'hiver, les points affichés devraient se regrouper le long d'une ligne.

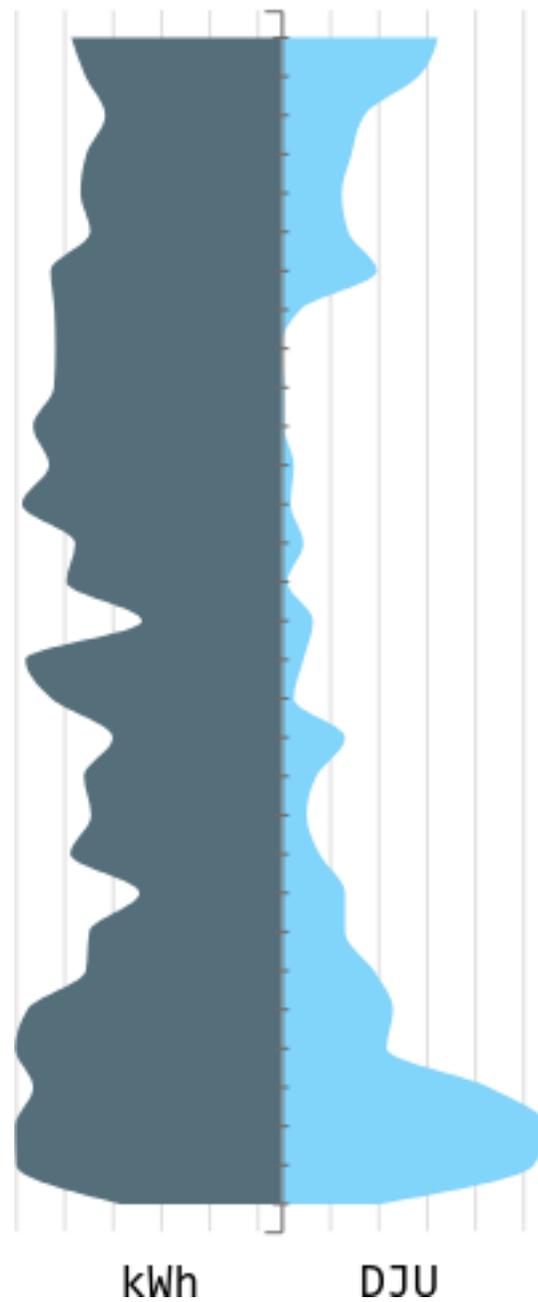
L'analyse de la pente de cette droite permet de caractériser l'isolation et le système de chauffage de votre habitation. Par exemple, si la droite semble horizontale, cela veut peut-être dire que votre chauffage est toujours au maximum et donc qu'il est sous dimensionné.

Si les points semblent complètement désorganisés, c'est qu'il n'y a aucune corrélation entre la température extérieure et votre consommation d'électricité.

### Graphique papillon

La graphique papillon est un graphique d'évolution temporelle, à la vertical, sur lequel on affiche simultanément l'évolution de 2 grandeurs dans le temps.

### Aperçus



## Comment lire le graphique

Comme le graphique d'analyse croisée, le graphique papillon permet de chercher des corrélations entre 2 grandeurs. Dans le cas de æneria, on compare toujours la consommation d'électricité et une grandeur météorologique.

Si 2 grandeurs sont directement corrélées, le graphique devrait ressembler à un papillon, c'est à dire être symétrique, d'où sont nom.

### 1.2.3 Glossaire

**DJU** Degrés Jour Unifié est la différence entre la température extérieure et une température de référence (18°C pour æneria)

Plus le nombre de DJU est grand pour une période donnée et plus il a fait froid.

æneria utilise la méthode chauffagiste pour calculer les DJU.

**Voir aussi :**

[Le DJU sur Wikipédia](#)

**Humidité** C'est la mesure de l'humidité relative dans l'air en %.

— A 0%, l'air est complètement sec

— A 100%, l'air est saturé en eau

Pour l'homme, un air sec est plus confortable qu'un air humide.

**Voir aussi :**

[L'humidité relative sur Wikipédia](#)

**kWh** Kilowatt-heure est l'unité d'énergie généralement utilisée pour mesurer la consommation d'électricité.

**Voir aussi :**

[le kWh sur Wikipédia](#)

**Nébulosité** C'est une mesure qui permet de caractériser la fraction de ciel couverte par des nuages en %.

— A 0%, le ciel est bleu et complètement dégagé

— A 100% le ciel est complètement obstrué par des nuages

**Voir aussi :**

[La nébulosité sur Wikipédia](#)

**Précipitations** C'est la mesure du nombre de millimètres de pluie tombée par m<sup>2</sup>.

**Voir aussi :**

[Les précipitations sur Wikipédia](#)

### 1.2.4 Gestion des adresses

Dans æneria, chaque utilisateur peut gérer des *Adresses*. Pour chaque *adresse*, on configure un compteur Linky et une station d'observation météo.

Une *adresse* peut ensuite être rendue publique pour quelle soit visible par tous les utilisateurs de æneria, ou bien, elle peut être partagée à une liste d'utilisateur.



Pour gérer vos adresses, aller sur la page de configuration en cliquant sur le bouton dans la barre du haut.

Ci-dessous, le formulaire d'ajout d'une adresse :

# Adresses

 L'igloo

-  Linky : 4 rue Voltaire, 11000, Carcassonne, France
-  Météo : Dumont D'urville
-  L'adresse est privée
-  Partagée avec personne

[+ Ajouter une adresse](#)

## Ajouter une adresse

Cliquer sur le bouton `Ajouter` une adresse sous la liste des adresses existantes :

**i Informations générales**

Icone :  ▾

Nom de l'adresse :

Partager avec : personne ▾

Public  
Les données d'une adress public sont visibles par tous les utilisateurs d'aeneria.

---

**📦 Compteur linky**

4 rue Voltaire, 11000, Carcassonne, France [Changer de compteur](#)  
(via votre espace Enedis)

---

**🌤️ Station météo de référence**

Station d'observation

[Annuler](#) [Enregistrer](#)

**Note :** Selon la configuration de l'application, ce bouton peut ne pas être présent.

---

### Editer une adresse

Pour éditer une adresse, sur la liste , cliquer sur le bouton  de l'adresse souhaitée.

Il est alors possible de modifier :

- le nom de l'adresse
- son icone
- la station météo d'observation de référence
- Le compteur Linky associé à l'adresse

### Rafraichir les données d'une adresse

Il peut parfois être nécessaire de rafraichir les données d'une adresse manuellement : c'est à dire forcer la mise des données depuis les serveurs d'Enedis et de Météo France.

Cliquez sur le bouton  de l'adresse souhaitée.

Le formulaire qui apparait permet de rafraichir les données pour le flux désiré. Il n'est actuellement pas possible de rafraichir les données météos au delà de 2 semaines.

**Note :** Selon la configuration de l'application, cette fonctionnalité peut ne pas être présente.

---

### Exporter les données d'une adresse

æneria permet à chaque utilisateur d'exporter facilement les données d'une adresse au format ODS.

Cliquez sur le bouton  de l'adresse souhaitée.

Il est alors possible d'exporter :

- toutes les données
- seulement les données entre 2 dates

**Note :** Selon la configuration de l'application, cette fonctionnalité peut ne pas être présente.

---

### Importer des données d'une adresse

Il est possible d'importer un fichier de données provenant d'une exportation æneria.

Cliquez sur le bouton  de l'adresse souhaitée.

**Danger :** Attention, si des données existent pour les dates importées, elle seront écrasées !

**Avertissement :** Cette fonctionnalité a été réalisée pour importer des fichiers provenant d'un export de données créé via aeneria : si vous essayez d'importer des fichiers provenant d'une autre source, faites-le à vos risques et périls !

---

**Note :** Selon la configuration de l'application, cette fonctionnalité peut ne pas être présente.

---

## 1.3 Documentation administrateur

aeneria peut-être administré via l'interface web ou bien en ligne de commande via la console Symfony.

### 1.3.1 Via l'interface web

Les utilisateurs avec le rôle *Admin* ont accès aux informations d'administration d'aeneria.

Pour accéder à ces informations, aller sur la page de configuration en cliquant sur le bouton  dans la barre du haut. Si vous êtes administrateur, une carte supplémentaire apparait sous votre liste d'adresses :

# Administration

 Gérer les utilisateurs

 Voir les derniers logs

## Configurations courantes

|  |          |
|--|----------|
| Nombre d'adresses max par utilisateur                    | Illimité |
| Partage d'adresses entre utilisateurs                    | Oui      |
| Les adresses peuvent être publiques                      | Non      |
| Les utilisateurs peuvent recharger leur données via l'UI | Oui      |
| Les utilisateurs peuvent exporter leur données via l'UI  | Oui      |

Ces paramètres peuvent être changés en surchargeant le fichier `.env` [?](#)

Plusieurs éléments sont visibles sur cette page :

- Un bouton pour accéder à la gestion des utilisateurs
- Un bouton pour accéder aux derniers logs de l'application
- La liste des configurations courantes de l'application

## Configurations

Plusieurs fonctionnalités sont paramétrables dans æneria :

- Le nombre maximum d'adresses qu'un utilisateur puisse créer
- La possibilité de partager des adresses entre utilisateurs
- La possibilité de rendre une adresse publique pour l'ensemble des utilisateurs
- La possibilité pour des utilisateur de pouvoir recharger leurs données via l'interface
- La possibilité pour des utilisateur de pouvoir exporter leurs données via l'interface

Les paramètres courants sont visibles via l'interface d'administration mais ne sont pas modifiable via l'interface web.

Pour les changer, il faut modifier le fichier `.env` à la racine d'æneria (le fichier est auto-documenté).

Ci-dessous, un exemple de paramétrage de ce fichier :

```
###> symfony/framework-bundle ###
APP_ENV=prod
APP_SECRET=app_secret_you_should_change_this_value
#TRUSTED_PROXIES=127.0.0.1,127.0.0.2
#TRUSTED_HOSTS='^localhost|example\.com$'
###< symfony/framework-bundle ###
```

(suite sur la page suivante)

(suite de la page précédente)

```

###> doctrine/doctrine-bundle ###
# Format described at http://docs.doctrine-project.org/projects/doctrine-dbal/en/
↪latest/reference/configuration.html#connecting-using-a-url
# For an SQLite database, use: "sqlite:///kernel.project_dir%/var/data.db"
# Configure your db driver and server_version in config/packages/doctrine.yaml
DATABASE_URL=mysql://admin:admin@127.0.0.1:3306/aeneria
# DATABASE_URL=pgsql://admin:password@127.0.0.1:5432/aeneria
###< doctrine/doctrine-bundle ###

# Number of places a user can create (-1 for no limit)
AENERIA_USER_MAX_PLACES=-1
# Can users share place between them
AENERIA_USER_CAN_SHARE_PLACE=1
# Can user fetch data from ui
AENERIA_USER_CAN_FETCH=1
# Can user export data from ui
AENERIA_USER_CAN_EXPORT=1
# Can a place be public
AENERIA_PLACE_CAN_BE_PUBLIC=0

```

## Gestion des utilisateurs

Accédez à la page de gestion des utilisateurs en cliquant sur le bouton **Gérer les utilisateurs** ou en visitant la page `/admin/users`.

Sur cette page se trouve un tableau listant l'ensemble des utilisateurs d'aeneria.

Au bout de chaque ligne, des boutons vous permettent de :

- Modifier l'utilisateur
- Désactiver/activer l'utilisateur
- Supprimer l'utilisateur (et l'ensemble de ses données)

Pour créer un nouvel utilisateur, cliquez sur le bouton **Ajouter un utilisateur**.

Un utilisateur désactivé ne peut plus se connecter, mais ses données ne sont pas supprimées.

## Les logs

L'interface web permet de visualiser les derniers logs d'aeneria. Cliquez sur le bouton **Voir les derniers logs** ou visitez la page `/admin/log`.

### 1.3.2 Via la console Symfony

Plusieurs commandes Symfony existent pour administrer aeneria.

Les commandes sont toutes auto-documentées, et s'utilisent comme des commandes Symfony classiques, par exemple :

```

# Se rendre dans le dossier racine d'aeneria
cd /emplacement/de/aeneria

# Pour connaître l'utilisation d'une commande :
php7.3 bin/console aeneria:user:activate --help

```

(suite sur la page suivante)

```
# Pour l'utiliser :  
php7.3 bin/console aeneria:user:activate username
```

### Commandes génériques

- `aeneria:install` : Installateur d'aeneria
- `aeneria:fetch-data` : Récupérer les différentes données (c'est cette commande qui est appelé quotidiennement par le cron d'aeneria)
- `aeneria:version` : Connaître la version courante d'aeneria

### Commandes de gestion des utilisateurs

- `aeneria:user:add` : Ajouter un utilisateur
- `aeneria:user:edit` : Modifier un utilisateur
- `aeneria:user:activate` : Activer un utilisateur
- `aeneria:user:deactivate` : Désactiver un utilisateur
- `aeneria:user:exist` : Savoir si un utilisateur existe déjà
- `aeneria:user:grant` : Donner à un utilisateur le rôle d'admin
- `aeneria:user:ungrant` : Retirer à un utilisateur le rôle d'admin

### Commandes de développement

Cette commande ne peut être utilisées que sur un environnement de développement.

- `aeneria:dev:generate-fake-data` : Générer de fausses données

## 1.4 Documentation développeur

### 1.4.1 Mettre en place un environnement de développement

Il n'y pas encore de docker ou de truc comme ça pour installer facilement le projet. Pour monter un environnement de dev :

- Récupérez le dépôt git du projet :

```
git clone git@gitlab.com:aeneria/aeneria-app.git
```

- Créez une base de données sur votre serveur de base de données (MySQL ou PostgreSQL)
- Installez `composer` ([Voir comment sur le site de composer](#))
- Récupérez les dépendances du projet :

```
php7.3 composer.phar install
```

- Copiez le fichier `.env.dist` et adaptez-le :
  - Modifiez la variable `APP_ENV`
  - Adaptez la chaîne de connexion de la base de données

```
cp .env.dist .env
```

```
# Fichier .env

...

###> symfony/framework-bundle ###
# Changer la variable ``APP_ENV``
APP_ENV=dev

...

###> doctrine/doctrine-bundle ###
# Format described at http://docs.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/configuration.html#connecting-using-a-url
↪ latest/reference/configuration.html#connecting-using-a-url
# For MySQL database use: "pgsql://[database_user]:[database_password]@127.0.0.1:5432/[database_name]"
↪ 1:5432/[database_name]
# For PostgreSQL database use: "mysql://[database_user]:[database_password]@127.0.0.1:3306/[database_name]"
↪ 1:3306/[database_name]
# For an SQLite database, use: "sqlite:///kernel.project_dir%/var/data.db"
# Configure your db driver and server_version in config/packages/doctrine.yaml
DATABASE_URL=[VOTRE CONFIG ICI]
###< doctrine/doctrine-bundle ###
```

— Installer aeneria :

```
php7.3 bin/console aeneria:install
```

— Ajoutez un premier utilisateur et donnez-lui les droits administrateur :

```
php7.3 bin/console aeneria:user:add [username] [password]
php7.3 bin/console aeneria:user:grant [username]
```

— Générer des données de tests :

```
# Génère des données pour les 3 derniers mois pour un utilisateur user-test/password
# attention, la génération peut-être un peu longue, vous pouvez réduire le nombre de
# de données créées avec l'option --from
php7.3 bin/console aeneria:dev:generate-fake-data
```

— Générer les assets une première fois :

```
# Installer les dépendances javascript
yarn install

# Générer les assets en mode dev
yarn dev
```

— Enfin, configurez NGINX ou Apache comme pour une application Symfony 5 classique.

## 1.4.2 Générer les assets

Les assets sont gérés avec Webpack Encore.

```
# Installer les dépendances
yarn install

# Build en mode prod
```

(suite sur la page suivante)

```
yarn build
```

```
# Build en mode dev
yarn dev
```

```
# Build en mode watch
yarn dev --watch
```

### 1.4.3 Générer cette Documentation

La documentation est automatiquement générée à chaque nouveau tag à l'aide de [Read the Docs](#). Tout est donc basé sur [Sphinx](#) et écrit en RST.

Les fichiers se trouvent dans le dossier `docs`.

Si vous la modifiez, il est nécessaire de la générer en local pour être sûr qu'il n'y a pas d'erreur de syntaxe.

Pour se faire, suivez ces étapes :

```
# Vérifier que vous avez bien descendu le repository
# du thème sphinx pour aeneria
git submodule init
git submodule update --recursive

# La première fois que vous générez cette doc, install
# l'environnement python pour sphinx
pip install --user virtualenv
mkdir ~/venvs
virtualenv ~/venvs/sphinx
. ~/venvs/sphinx/bin/activate
pip install sphinx

# Régénérer la documentation
. ~/venvs/sphinx/bin/activate
cd docs/
make html
```

La page d'accueil de la documentation ainsi générée se trouve ici : `docs/_build/html.index.html`

### 1.4.4 Tests & CS fixer

#### PHPUNIT

Pour lancer les tests PHPUNIT, il faut préalablement avoir créé un minimum de données de tests :

```
# Il faut avoir un utilisateur 'admin/password' avec des données à jour:
php7.3 bin/console aeneria:dev:generate-fake-data --from="7 days ago" --user-
↳name=admin --user-password=password
# Cette commande est à lancer une fois par jour

# On s'assure qu'il a les droits admin:
php7.3 bin/console aeneria:user:grant admin

# Il faut avoir un utilisateur 'user-test/password' avec des données pour les 7
↳derniers jours:
```

(suite sur la page suivante)

(suite de la page précédente)

```
php7.3 bin/console aeneria:dev:generate-fake-data --from="7 days ago" --user-  
↳name=user-test --user-password=password  
# La commande précédente est à lancer une fois par jour  
  
# Enfin, on s'assure que user-test n'est pas admin  
php7.3 bin/console aeneria:user:ungrant user-test  
  
# On peut maintenant lancer les tests l'esprit tranquille:  
php7.3 bin/phpunit
```

## CS Fixer

Avant de commiter, passez-donc un petit coup de CS-Fixer pour s'assurer que le style de code reste homogène :

```
vendor/bin/php-cs-fixer fix --allow-risky=yes
```



## CHAPITRE 2

---

### Todo List

---

- Ajouter un onglet pour comparer des périodes et/ou des adresses
- Implémenter l'API de Gazpar de GRDF



## CHAPITRE 3

---

Support

---

Posez vos questions sur [Gitlab](#) ou via [Twitter](#).



**D**

DJU, **14**

**H**

Humidité, **14**

**K**

kWH, **14**

**N**

Nebulosité, **14**

**P**

Précipitations, **14**