
aeneria Documentation

Simon Mellerin

janv. 13, 2021

Contenu :

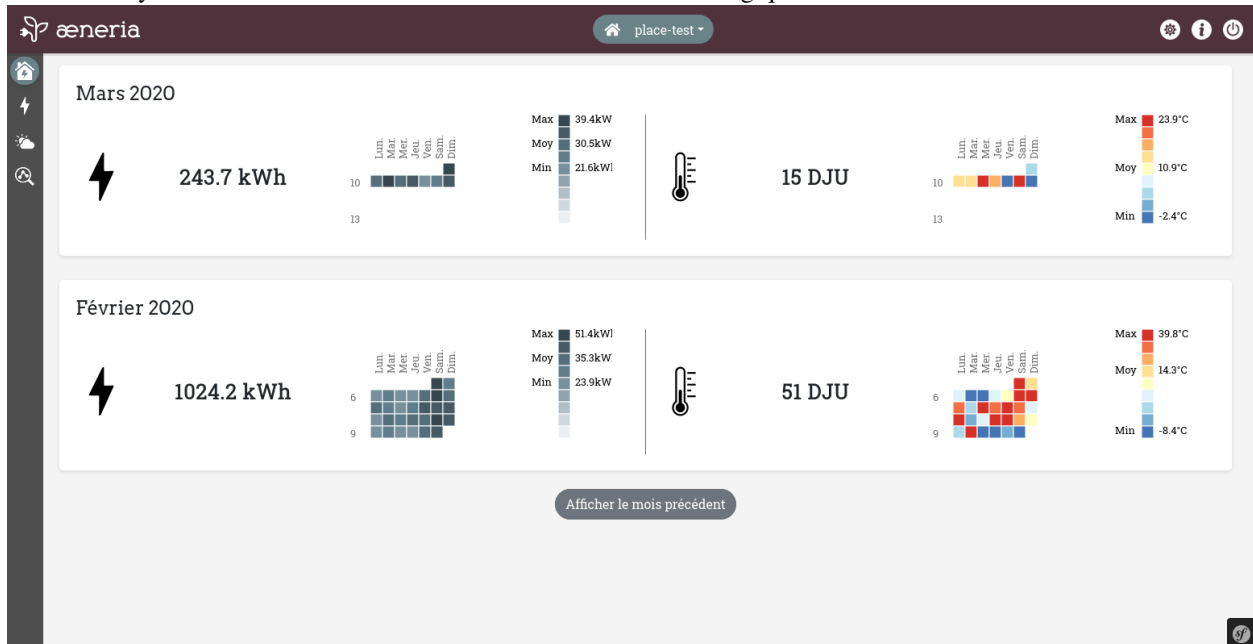
1	Installer æneria facilement avec YunoHost !	3
1.1	Documentation utilisateur	3
1.2	Documentation administrateur	15
1.3	Documentation developpeur	24
2	Todo List	29
3	Support	31
	Index	33



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

L'idée derrière æneria 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



CHAPITRE 1

Installer æneria facilement avec YunoHost !



1.1 Documentation utilisateur

Bienvenue sur la documentation d'æneria !

Retrouvez ici toutes les informations dont vous avez besoin pour utiliser æneria :

1.1.1 Les différents tableaux de bords

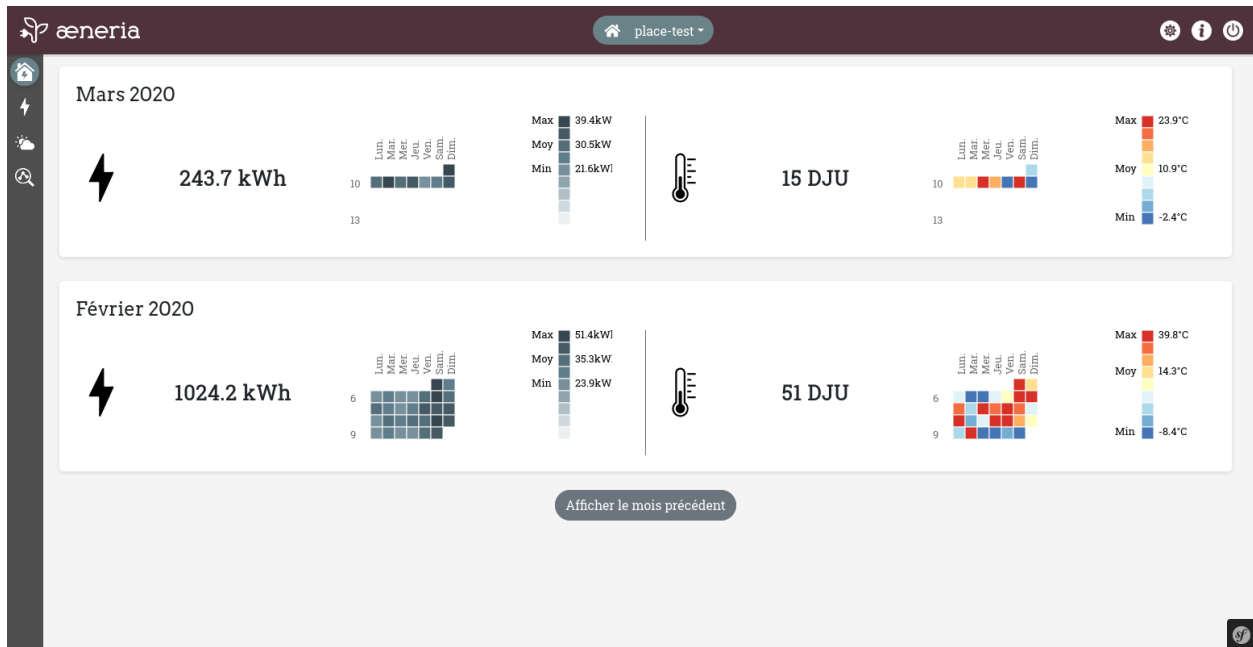
Cette page décrit les différents tableaux de bord disponibles sur æneria.



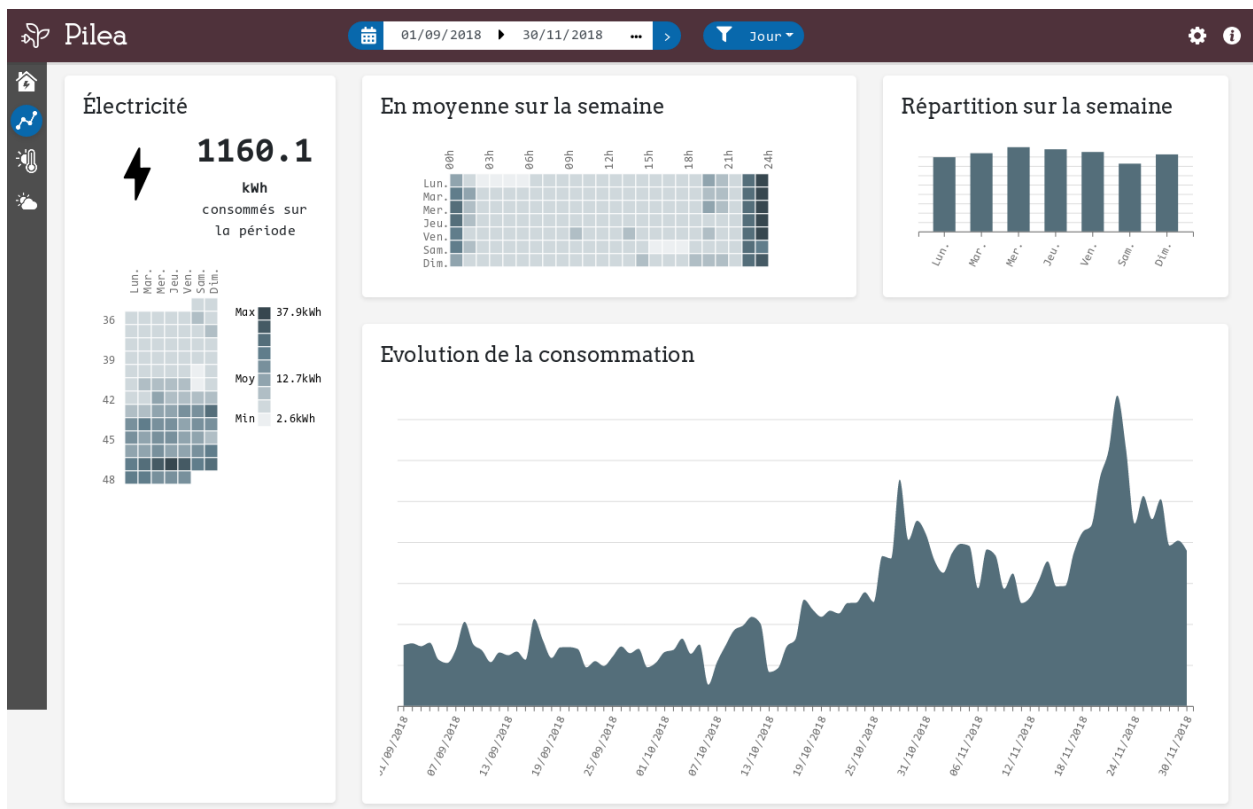
L'accueil

Lors de la connexion à æneria, 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



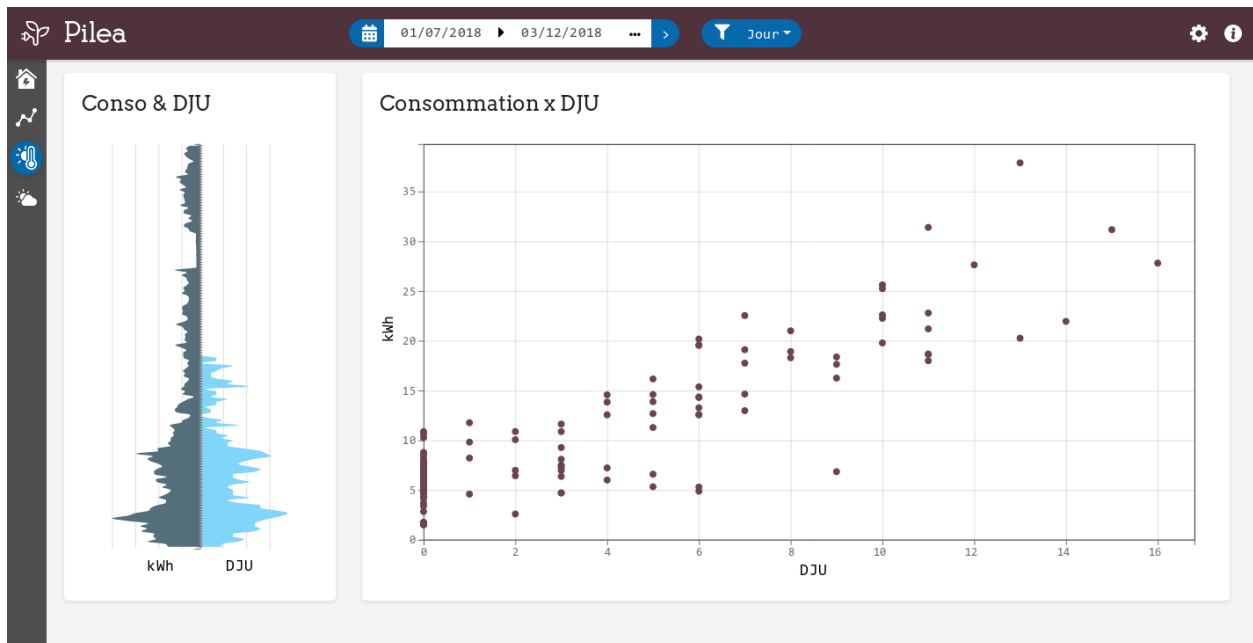


Analyse de la météo

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

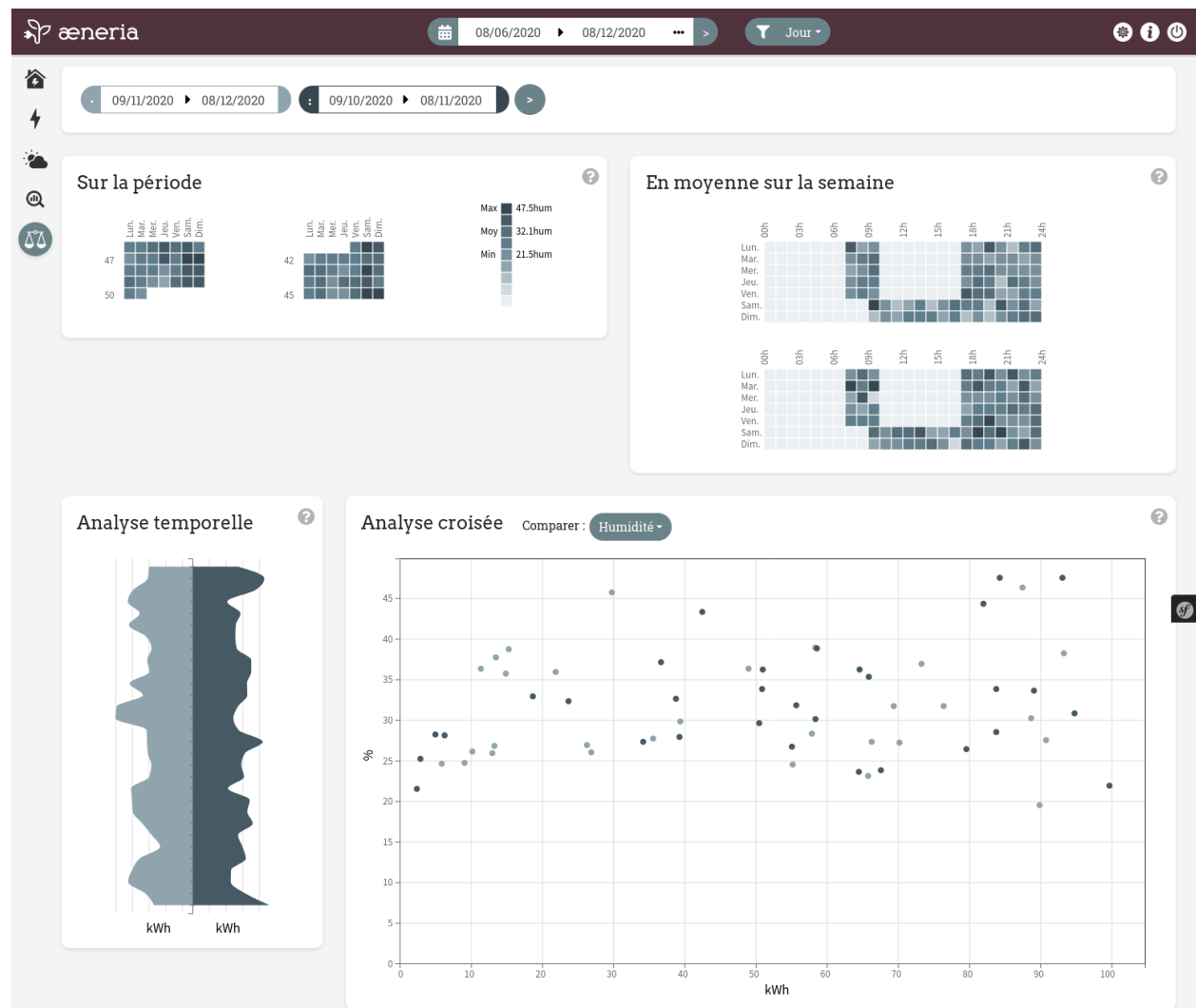


Analyse croisée





Comparaison de périodes



1.1.2 Comprendre les différents graphiques

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

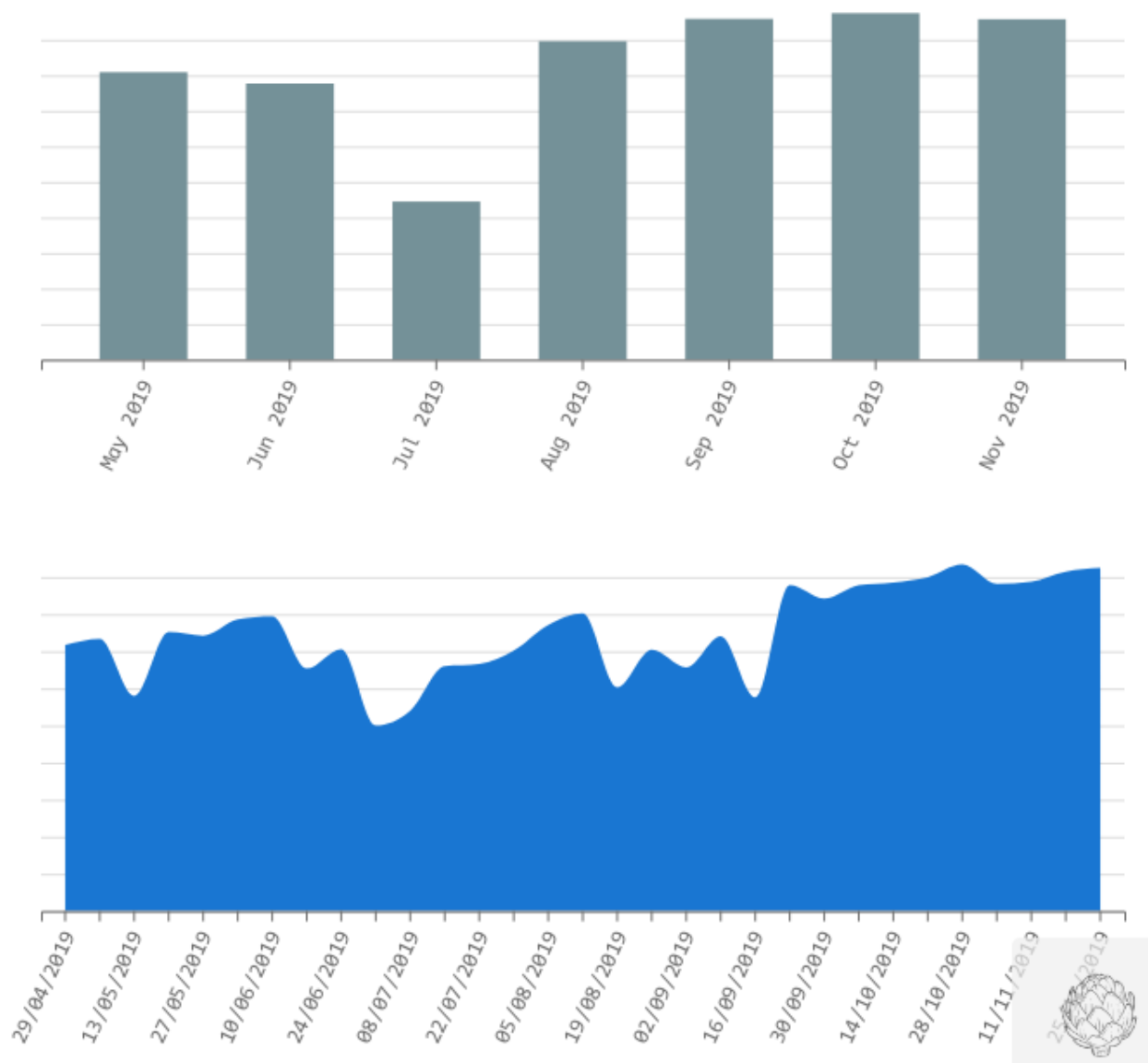
Graphique d'évolution temporelle

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

Il affiche, pour la période sélectionnée, des points de donnée en fonction de la granulosité choisie.

Il présente chaque point 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 donné 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 ?

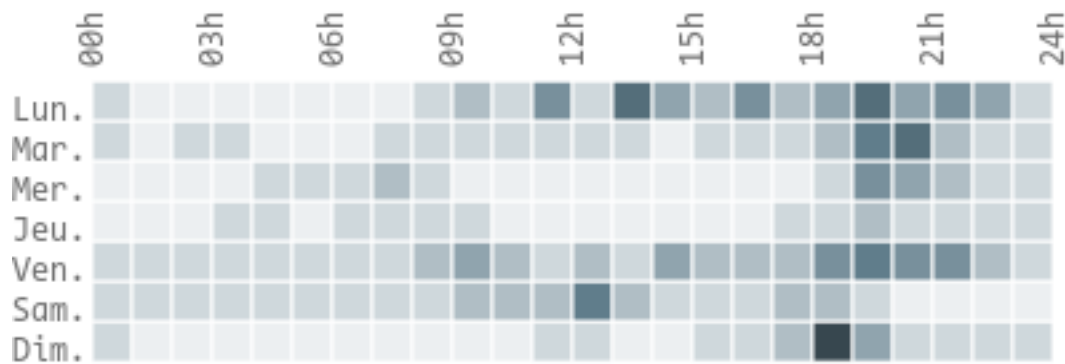
Carte de chaleur Jours/Heures

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

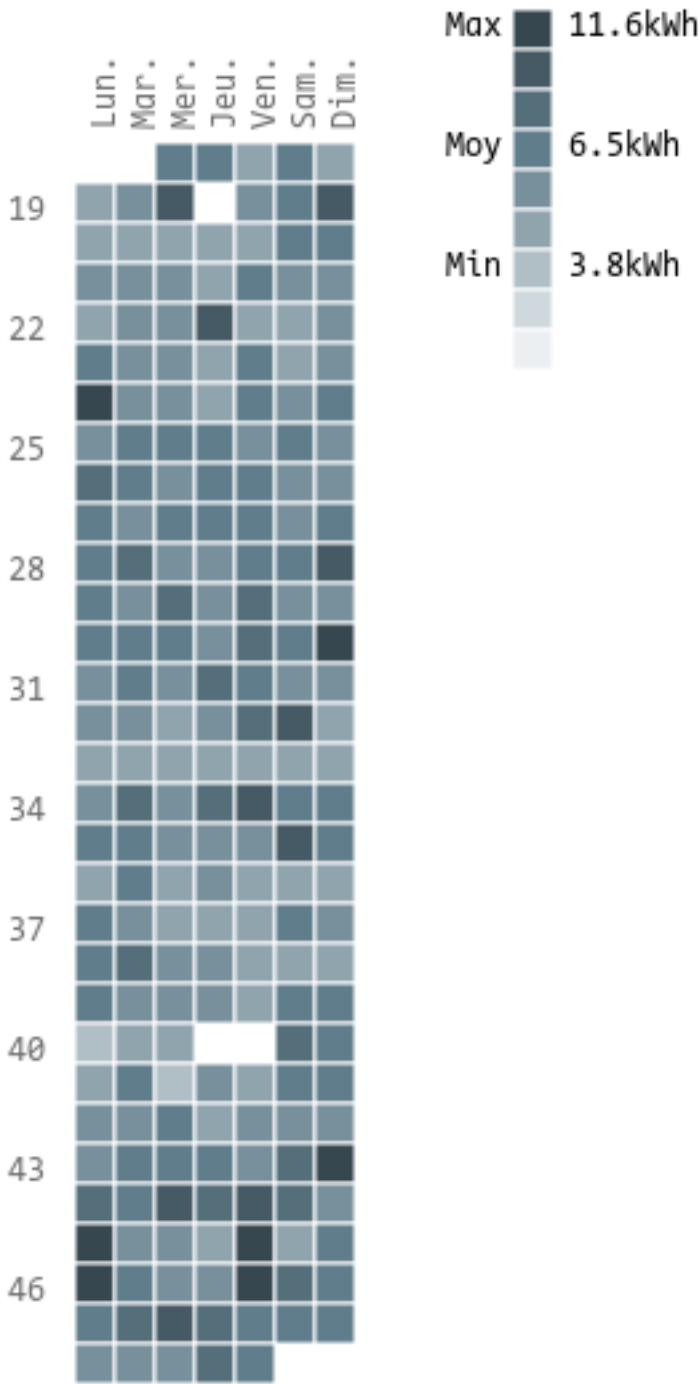
Ce graphique permet de mettre en valeur des schémas de consommation. Les tâches observées doivent refléter vos habitudes de consommation.

- 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 foncées, 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 ?

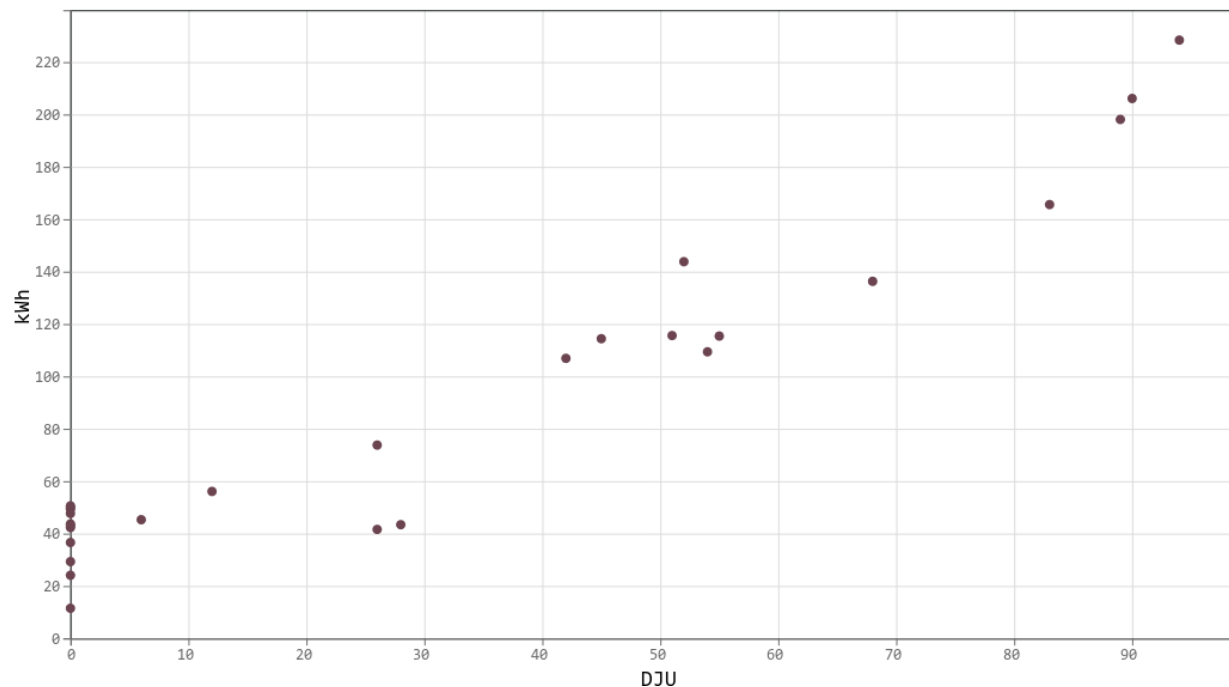
Cette représentation permet aussi d'observer des événements tels que les vacances : je suis parti une semaine en février, pourtant, je ne le vois pas sur le graphique ? Ai-je éteint le chauffage avant de partir ?

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 on sélectionne une période d'hiver, les points affichés devraient se regrouper le long d'une ligne.

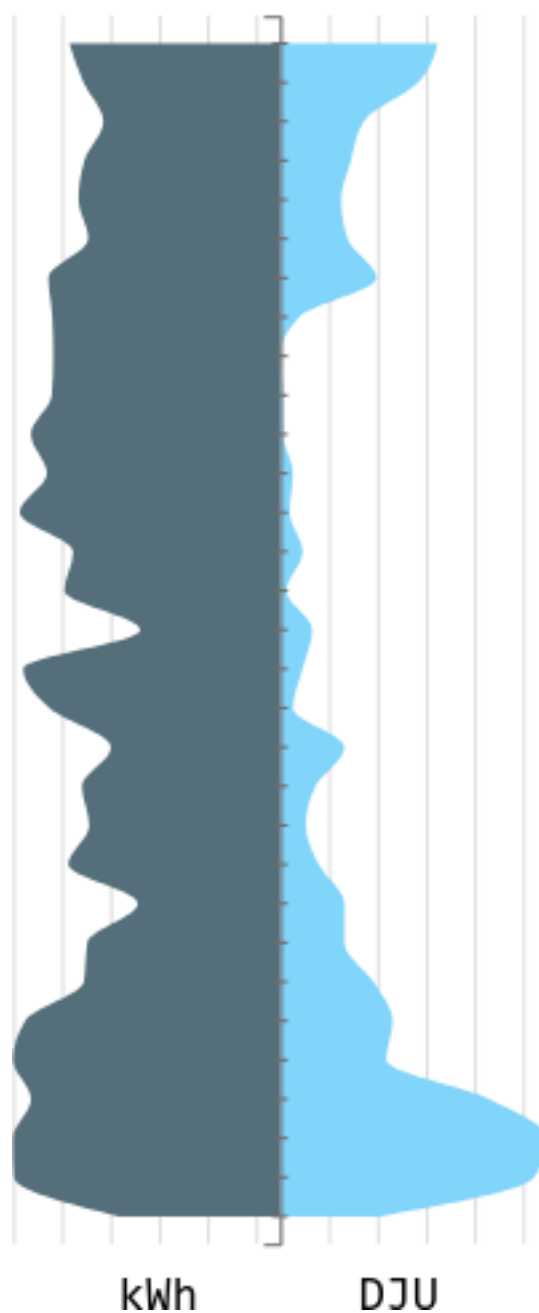
L'analyse de la pente de cette droite permet de caractériser l'isolation et le système de chauffage de votre habitation. Par exemple, si la droite semble horizontale, cela veut peut-être dire que votre chauffage est toujours au maximum et donc qu'il est sous dimensionné.

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.

1.1.3 Glossaire

DJU Les Degrés Jour Unifié sont 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 m².

Voir aussi :

[Les précipitations sur Wikipédia](#)

1.1.4 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, allez 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

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

Informations générales

Icone :  ▾

Nom de l'adresse :

Partager avec : personne ▾

☐ Public

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

Compteur linky

4 rue Voltaire, 11000, Carcassonne, France

(via votre espace Enedis)

Station météo de référence

Station d'observation

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

Editer une adresse

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

Il est alors possible de modifier :

- le nom de l'adresse
- son icône
- 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 à jour 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 météo 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 æneria : 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.

1.2 Documentation administrateur

1.2.1 Installer æneria

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

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 æneria.



Installation à la main

æneria 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 PostgreSQL.

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
rm aeneria-app-latest.tar.gz
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"
```

(suite sur la page suivante)

(suite de la page précédente)

```
# 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. Configurer Enedis Data-connect

æneria utilise l'API Enedis Data Connect pour obtenir les données de consommation d'électricité. Mais pour utiliser cette API il est nécessaire d'avoir un compte. Seulement, pour ouvrir un compte sur la Data Hub d'Enedis, il faut être une entreprise, une association ou une collectivité locale.

Pour permettre à tout le monde d'utiliser æneria, un proxy a été développé pour qu'une instance d'aeneria puisse bénéficier du compte d'aeneria.com.

Au lieu d'utiliser le comportement classique pour se connecter à Enedis :

votre instance æneria <=[via vos propres identifiants de connexion enedis]=> Enedis Data Connect

Vous pouvez configurer votre instance comme ça :

votre instance æneria <=> proxy.aeneria.com <=[via les identifiants de connexion d'aeneria.com]=> Enedis Data Connect

Il y a donc 2 sortes de mode :

Soit vous créez un compte Enedis et vous renseignez vos informations de connexion de cette manière dans le fichier `.env` :

```
# fichier .env

...

ENEDIS_CLIENT_ID=[votreClientIdEnedis]
ENEDIS_CLIENT_SECRET=[votreClientSecretEnedis]
ENEDIS_REDIRECT_URI=[votreRedirectUriEnedis]
ENEDIS_ENDPOINT_AUTH=https://mon-compte-particulier.enedis.fr
ENEDIS_ENDPOINT_TOKEN=https://gw.prn.api.enedis.fr
ENEDIS_ENDPOINT_DATA=https://gw.prn.api.enedis.fr

...
```

Note : Pour obtenir vos propres identifiants de connexion, rendez-vous sur [le Data Hub d'Enedis](#)

Soit vous utilisez proxy.aeneria.com en utilisant cette configuration

```
# fichier .env

...

ENEDIS_CLIENT_ID=whatYouWantItWouldNotBeUsed
ENEDIS_CLIENT_SECRET=whatYouWantItWouldNotBeUsed
ENEDIS_REDIRECT_URI=whatYouWantItWouldNotBeUsed
ENEDIS_ENDPOINT_AUTH=https://proxy.aeneria.com/enedis-data-connect
ENEDIS_ENDPOINT_TOKEN=https://proxy.aeneria.com/enedis-data-connect
ENEDIS_ENDPOINT_DATA=https://gw.prn.api.enedis.fr

...
```

Avertissement : proxy.aeneria.com est un serveur communautaire fourni à titre gracieux.

Merci de l'utiliser raisonnablement et dans un cadre privé non-commercial.

Nous nous réservons le droit de bannir de ce serveur les instances qui en feront un usage trop intensif, et ce **sans explications et sans avertissement**.

5. Créer un administrateur

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

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

6. Générer l'ensemble des flux Météo (facultatif - usage avancée)

Danger : Cette fonctionnalité correspond à un usage avancée. Testez d'abord aeneria sans l'utiliser.

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 météo 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 des 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 supprimées. Si vous souhaitez quand même les supprimer, vous pouvez le faire en utilisant la commande `aeneria:feed:clean-orphans`.

Avertissement : L'adresse générée par cette commande n'est pas destinée à ensuite être utilisée via l'interface d'aeneria. Elle a pour unique but de définir une première fois l'ensemble des stations météo.

7. 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" > /
etc/cron.d/aeneria
# où [user] est l'utilisateur linux qui lancera le cron
```

8. Configurer le serveur web

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

1.2.2 Mettre à jour aeneria

Avertissement :

Avant toute chose :

- faites un backup de la base de données d'aeneria
- Prenez en note la version courante d'aeneria

Rendez-vous dans le répertoire parent du répertoire d'installation d'aeneria, puis procédez comme suit :

```
# Renommer la version courante :
mv aeneria-app aeneria-app-backup

# Téléchargez et décompressez la dernière version au format tar.gz :
wget http://statics.aeneria.com/aeneria-app-latest.tar.gz
tar -xvzf aeneria-app-latest.tar.gz
rm aeneria-app-latest.tar.gz

# Entrez dans le répertoire d'aeneria :
cd aeneria-app

# Affichez le changelog pour vérifier s'il n'y a pas d'avertissement
# pour la mise à jour :
less CHANGELOG.md

# Copiez les différents fichiers de configuration :
mv ../aeneria-app-backup/.env .
mv ../aeneria-app-backup/config/packages/doctrine.yaml config/packages/doctrine.yaml
rsync -av ../aeneria-app-backup/var/ var/

# Lancer les éventuelles migrations :
php7.3 bin/console doctrine:migrations:migrate

# Videz les caches :
php7.3 bin/console c:c
```

Et voilà, votre instance d'aeneria est à jour !

1.2.3 Administrer aeneria

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

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 apparaît 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
- L'activation du mode démo (Désactiver les fonctions de configuration via l'UI)
- Le message affiché sur l'écran de login

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
```

(suite sur la page suivante)

```
#TRUSTED_HOSTS='^localhost|example\.com$'
###< symfony/framework-bundle ###

###> 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
# Activate demo mode
AENERIA_DEMO_MODE=0
# Welcome message
AENERIA_WELCOME_MESSAGE='Bienvenu sur æneria'
```

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`.

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

# 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

1.2.4 Migrer depuis Pilea

Un peu d'histoire

Pilea est l'ancien nom d'aeneria.

Les versions 0.5.x de Pilea reposaient sur l'architecture de l'ancien espace personnel Enedis. Elles utilisaient des technique de web-scraping pour récupérer les données de consommation d'électricité.

Cette solution n'était pas pérenne et au moment où Enedis a changé son site, les scripts de Pilea ne fonctionnaient plus.

Une refonte de toute une partie de Pilea a donc été réalisée pour s'appuyer sur la nouvelle API d'Enedis : [Data Connect](#). L'utilisation de l'API officielle permet une plus grande stabilité.

Cette refonte a été l'occasion de revoir plusieurs parties du code de Pilea, rendant les version 0.5.x incompatibles avec les version 1.x. Le nom de Pilea est peu évocateur et n'est pas très *moteur-de-recherche-friendly*. Cette petite refonte était donc l'occasion de le changer.

L'histoire de Pilea s'arrête donc avec la version 0.5.8 pour laisser sa place à **aeneria** !

Migrer de Pilea vers æneria

Comme évoqué précédemment, il n'est pas possible de faire une mise à jour automatique de Pilea vers æneria. Mais, grâce aux fonctions d'import et d'export, il est possible de transférer ses données depuis Pilea vers æneria.

Danger : Lisez bien la procédure avant de commencer.

Ne pas désinstallez Pilea avant la fin de la procédure.

Note : Il n'y a pas de moyen d'exporter l'ensemble des données de l'ensemble des utilisateurs d'un seul coup. Ceux-ci vont devoir tous se connecter pour sauvegarder *leurs* données, avant de les réimporter dans æneria.

Voici donc la marche à suivre :

- Mettez à jour Pilea vers une version supérieure ou égale à 0.5.6 (le numéro de version est visible dans l'encart « à propos » sur la page de configuration)
- Installez æneria
- **Pour chaque utilisateur**
 - Connectez-vous à Pilea
 - Allez sur la page de configuration
 - Pour chaque adresse, exportez l'ensemble des données en cliquant sur le bouton 
 - Connectez-vous à æneria
 - Allez sur la page de configuration
 - Créez chacune des adresses
 - Pour chaque adresse, importez les données précédemment exportées en cliquant sur le bouton 
- Vérifiez que chaque utilisateurs a bien importé ses données
- Désinstallez Pilea

1.3 Documentation developpeur

1.3.1 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`
 - Adapter la chaine 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
# For MySQL database use: "pgsql://[database_user]:[database_password]@127.0.0.1:5432/[database_name]"
# For PostgreSQL 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 ###
```

— 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 [admin_email] [password]
php7.3 bin/console aeneria:user:grant [admin_email]
```

— 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.

1.3.2 Générer les assets

Les assets sont gérés avec [Webpack Encore](#).

```
# Installer les dépendances
yarn install

# Build en mode prod
yarn build

# Build en mode dev
yarn dev

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

1.3.3 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`

1.3.4 Tester & améliorer la qualité du code

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@example.com --user-password=password
# Cette commande est à lancer une fois par jour

# On s'assure qu'il a les droits admin:
```

(suite sur la page suivante)

(suite de la page précédente)

```
php7.3 bin/console aeneria:user:grant admin@example.com

# Il faut avoir un utilisateur 'user-test/password' avec des données pour les 7
↳ derniers jours:
php7.3 bin/console aeneria:dev:generate-fake-data --from="7 days ago" --user-
↳ name=user-test@example.com --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@example.com

# 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
```

1.3.5 Livrer une nouvelle version

Pour livrer une nouvelle version d'aeneria :

Commencez par :

- Mettre à jour le numéro de version dans `config/services.yaml`
- Renseigner les nouveautés de cette version dans le fichier `CHANGELOG.md`

Commitez ces changements et poussez un nouveau tag :

```
git add CHANGELOG.md config/services.yaml
git commit -m "Prepare 1.2.3"
git push
git tag 1.2.3
git push --tag
```

La CI Gitlab va alors :

- Générer les assets en mode prod
- Télécharger les dépendances Composer en mode prod
- Passer les tests
- Archiver les sources (avec les dépendances Composer et les assets)
- Envoyer cette archive sur `statics.aeneria.com`
- Créer une release Gitlab

Il reste ensuite à modifier la [release Gitlab](#) pour y renseigner le changelog (copier/coller du fichier `CHANGELOG.md`)

La documentation sera régénérée automatiquement par [readthedocs](#).

C'est bon, votre nouvelle version est en ligne !

CHAPITRE 2

Todo List

- Ajouter la consommation de gaz via l'API GRDF ADICT

CHAPITRE 3

Support

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

D

DJU, [12](#)

H

Humidité, [12](#)

K

kWH, [12](#)

N

Nebulosité, [12](#)

P

Précipitations, [12](#)