

# Magasin Connecté 4.0

## Guide d'utilisation

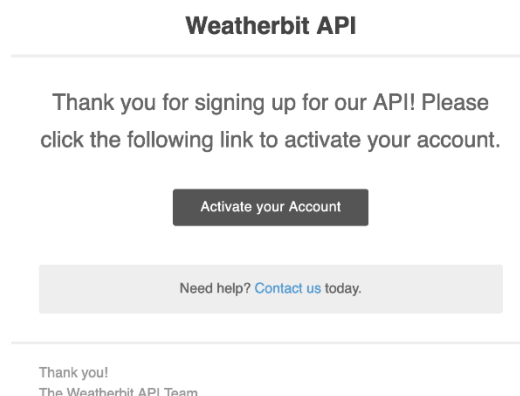
Gervilliers Mattias, Kusiak Guillaume,  
Pistoresi Marius et Pulitano Alessio

## Table des matières

Guide de création de clé api weatherbit .....	3
Migration API SolarEdge V2 .....	4
Modification des routes api utilisées.....	4
Guide d'utilisation du site .....	5
Horaires .....	6
Utilisateurs.....	6
Le choix de nos technologies .....	7

# Guide de création de clé api weatherbit

1. Crée un compte sur <https://www.weatherbit.io/account/create>
2. Activer le lien de confirmation reçu par mail



3. Ce login sur <https://www.weatherbit.io/account/login>
4. Renseigner un nom, FR - France pour le pays, la raison d'utilisation de l'api (school project), et choisir le free trial

Welcome to Weatherbit.io!

Before you can access your free API key, we need just a few more things:

Company Name/Organization

Name\*

Billing Currency\*

Country\*

VAT ID

API Usage Purpose\*

API Tier\*  [Billing Terms and Conditions](#)

**Plan Details**

**Plan: Free Trial (21 Days)**

- 1,500 req/day
- 1,500 historical req/day
- Current weather + alerts
- 20 years historical
- 16 day forecasts
- 240 hour forecasts
- 60 minute forecasts
- + Energy / Air Quality / Agweather / Climate Normals API
- Non-Commercial use only
- 95.0% Uptime
- Price: 60/mo

[Proceed to Dashboard](#)

5. Vous allez ensuite recevoir la clé api par mail ou sur le dashboard <https://www.weatherbit.io/account/dashboard> (à noter qu'il faut attendre environ 30min-1 heure pour que la clé soit fonctionnelle)

## Welcome to the Weatherbit API

You are currently subscribed to the "Trial Plan".

If you would like to upgrade or change your plan, please navigate to the "Manage Subscription" tab.

Your API Key is currently provisioning, and can be accessed in the "API Key" tab. **Provisioning typically takes 10-15 mins.**

[Okay!](#)

# Migration API SolarEdge V2

Guide officiel: <https://developers.solaredge.com/docs/monitoring/lpc9ijhyrc72l-migrating-from-v1-to-v2>

Après son expiration courant 2024 la v1 de l'api solarEdge que nous utilisons actuellement va devenir obsolète.

Tout d'abord il faut générer une nouvelle clé api nommée User key, qui sera passé dans le header en tant que "X-API-Key".

ex: curl -H "X-API-Key: 95904CD72B844D5DA2EA1C4D13EB66CA"

<https://api.solaredge.com/monitoring/sites/1234>

Pour générer cette clé api il faut aller dans la console administration de l'api ou contacter le support de solar edge

## Modification des routes api utilisées

### fetchDataPeriod

**V1:** /site/{siteId}/dataPeriod

**V2:** /sites/{siteId}/energy?resolution=lifetime  
(check l'objet période)

<https://developers.solaredge.com/docs/monitoring/13d262845566d-site-energy>

### fetchEnergy

**V1:** /site/{siteId}/energy?timeUnit={timeUnit}&startDate={startDate}&endDate={endDate}

**V2:** /sites/{siteId}/energy?resolution={resolution}&from={startDate}&to={endDate}

<https://developers.solaredge.com/docs/monitoring/13d262845566d-site-energy>

### fetchEnergyDetails

**V1:** /site/{siteId}/energyDetails?meters=PRODUCTION,CONSUMPTION&timeUnit={timeUnit}&startTime={startTime}&endTime={endTime}

**V2:** /sites/{siteId}/meters/{serialNumber}/export-energy

<https://developers.solaredge.com/docs/monitoring/1261eef741ceb-meter-export-energy>

### fetchOverview

**V1:** /site/{siteId}/overview

**V2:** /sites/{siteId}/overview?from={startDate}to={endDate}

<https://developers.solaredge.com/docs/monitoring/bc101de319142-site-overview>

### fetchPower

**V1:** /site/{siteId}/power?startTime={startTime}&endTime={endTime}

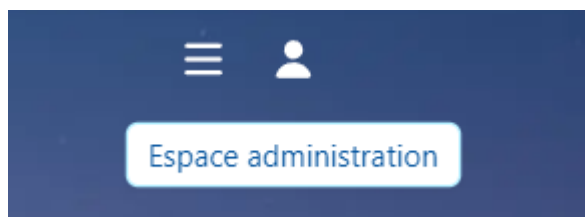
**V2:** /sites/{siteId}/power?resolution={resolution}&from={startDate}&to={endDate}

<https://developers.solaredge.com/docs/monitoring/1162015c5c19c-site-power>

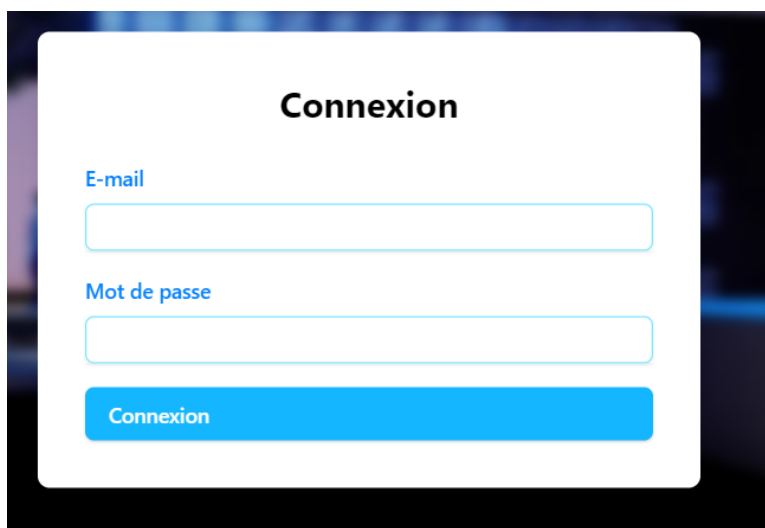
## Guide d'utilisation du site

Vous pouvez gérer les horaires de l'épicerie solidaire ainsi que les utilisateurs depuis l'espace d'administration disponible sur le site. Pour vous y rendre :

1. Cliquer sur votre compte en haut à droite



Vous accéderez à un formulaire de connexion. Utilisez votre adresse email et votre mot de passe pour vous identifier.

A white login form titled 'Connexion' centered at the top. Below the title are two input fields: the first is labeled 'E-mail' and the second is labeled 'Mot de passe'. At the bottom of the form is a blue button with the text 'Connexion' in white.

Une fois la connexion établie, la page d'administration apparaît.

Deux onglets sont alors accessibles : "Horaires" et "Utilisateurs". Vous pouvez dès à présent modifier les horaires ou les informations des utilisateurs de votre magasin connecté.

## Horaires

Administration

Horaires

Utilisateurs

Gestion des horaires de l'épicerie solidaire

Semaine : 03/06/2024 au 09/06/2024

AGORAé Etoile

Enregistrer

**Lundi**

09:00

12:00

Pause déjeuner Ouvert



**Mardi**

--:--

--:--

et

--:--

--:--

Pause déjeuner Ouvert



**Mercredi**

07:00

--:--

Pause déjeuner Ouvert



**Judi**

--:--

--:--

et

--:--

--:--

Pause déjeuner Ouvert



## Utilisateurs

Administration

Horaires

Utilisateurs

Gestion des utilisateurs

Nouvel utilisateur

E-mail

Rôle

- ☐ Administrateur
- ☒ Vendeur

Mot de passe

Confirmer le mot de passe

Ajouter

## Le choix de nos technologies

Création d'une application web avec Progressive Web App

Langage utilisé :

- Nuxt

Bibliothèques utilisées :

- Nuxt UI
- Chart.js
- Moment.js
- Crypto.js

Nuxt est un framework récent basé sur Vue.js, développé en France, et bénéficiant d'une communauté active. Voici les raisons qui nous ont conduit à choisir ce framework :

- Rendu côté serveur (SSR) : Améliore le SEO et les performances.
- Origine française et modernité : Framework récent et innovant.
- Communauté active : Large communauté offrant support et ressources.
- Routes automatiques : Gestion automatique des routes via le dossier pages.

Nous avons également choisi une Progressive Web App (PWA) pour permettre l'installation de l'application web sur votre smartphone en tant qu'application mobile. Cela offre, entre autres avantages, la possibilité de maintenir une seule base de code pour l'application web et la PWA, plutôt que d'avoir deux codes distincts pour le web et le mobile.

De plus nous avons utilisé Nuxt UI pour simplifier la conception de l'interface utilisateur (UI) de l'application. Cette bibliothèque offre une gamme d'outils et de composants prêts à l'emploi, nous permettant ainsi de créer rapidement une UI attrayante et fonctionnelle. En utilisant Nuxt UI, nous avons réduit le temps de développement nécessaire tout en assurant une cohérence visuelle à travers l'ensemble de l'application.

Nous avons utilisé la bibliothèque Chart.js pour créer des graphiques, notamment en intégrant des données provenant de l'API Solar Panel. Cette solution nous a permis de générer rapidement des graphiques interactifs et informatifs pour visualiser clairement les données solaires.