










-  Documentation API Ecologis - Version Production
  -  Vue d'ensemble
  -  Authentification
    - Système de rôles
    - Mécanisme JWT
    - Limites de taux
  -  Routes disponibles
    - 1.  Authentification (/auth)
      - POST /auth/register
      - POST /auth/login
      - POST /auth/refresh
      - POST /auth/logout
      - POST /auth/reset-password
      - POST /auth/change-password
    - 2.  Gestion des maisons (/maisons)
      - POST /maisons
      - GET /maisons
      - GET /maisons/:id
      - PUT /maisons/:id
      - DELETE /maisons/:id
      - PATCH /maisons/:id/tarif
      - POST /maisons/residents/ajouter
      - POST /maisons/residents/retirer
    - 3.  Gestion des résidents (/residents)
      - POST /residents
      - GET /residents
      - GET /residents/:id
      - PUT /residents/:id
      - DELETE /residents/:id
    - 4.  Gestion des consommations (/consommations)
      - POST /consommations
      - GET /consommations/resident/:residentId
      - GET /consommations/maison/:maisonId
      - PUT /consommations/:id
      - DELETE /consommations/:id
    - 5.  Gestion des factures (/factures)
      - POST /factures/generer/:residentId

- GET /factures/resident/:residentId
- GET /factures/maison/:maisonId
- GET /factures/:id
- PUT /factures/:id/payer
- 6. 📦 Gestion des abonnements (/abonnements)
  - GET /abonnements
  - POST /abonnements/souscrire
  - POST /abonnements/renouveler
  - GET /abonnements/actuel
  - POST /abonnements/annuler
  - GET /abonnements/historique
- 💰 Logique métier
  - Calcul des factures
  - Gestion des abonnements
  - Notifications automatiques
- 🛠 Variables importantes
  - Tarifs par défaut
  - Limites d'abonnement
- 📱 Instructions pour tester l'API avec Postman
  - 1. Configuration de l'environnement Postman
  - 2. Collection Postman recommandée
    - Authentification
    - Gestion des maisons
    - Gestion des résidents
    - Gestion des consommations
    - Gestion des factures
    - Gestion des abonnements
  - 3. Workflow de test recommandé
  - 4. Gestion automatique des tokens
- 🚀 Guide d'utilisation pour Flutter
  - 1. Configuration de l'API
  - 2. Service d'authentification
  - 3. Service API générique
  - 4. Gestion des erreurs
- 📋 Résumé pour le développeur Flutter
  - Points clés à retenir :
  - Workflow recommandé :
  - Sécurité :

-  [Support et contact](#)



# Documentation API Ecologis - Version Production

---



## Vue d'ensemble

---

L'API Ecologis est une solution de gestion de consommation électrique qui permet aux propriétaires de gérer leurs maisons, résidents et factures d'électricité. Elle gère automatiquement le calcul des factures basé sur les relevés de consommation et les tarifs personnalisés.



**Base URL:** <https://ecologis-api.onrender.com>



**Statut:** Production - Déployé sur Render

---



## Authentification

---

### Système de rôles

- **proprietaire** : Peut gérer maisons, résidents, abonnements et voir toutes les données
- **resident** : Peut voir ses propres consommations et factures

### Mécanisme JWT

- **Access Token** : Valide 15 minutes, utilisé pour toutes les requêtes authentifiées
- **Refresh Token** : Valide 7 jours, utilisé pour renouveler l'access token
- **Format** : `Authorization: Bearer <access_token>`

### Limites de taux

- **Authentification** : 5 tentatives par IP toutes les 15 minutes

- **Résidents** : 10 requêtes par IP par minute
- 



# Routes disponibles

---

## 1. Authentification (/auth)

**POST** /auth/register

**Créer un compte propriétaire**

**URL complète:** <https://ecologis-api.onrender.com/auth/register>

**Body JSON:**

```
{
  "nom": "Doe",
  "prenom": "John",
  "email": "john.doe@example.com",
  "telephone": "+22890123456",
  "motDePasse": "MotDePasse123!"
}
```

**Exemple cURL:**

```
curl -X POST https://ecologis-api.onrender.com/auth/register \
-H "Content-Type: application/json" \
-d '{
  "nom": "Doe",
  "prenom": "John",
  "email": "john.doe@example.com",
  "telephone": "+22890123456",
  "motDePasse": "MotDePasse123!"
}'
```

**Réponse succès (201):**

```
{
  "message": "Compte propriétaire créé avec succès",
  "user": {
    "_id": "64f1a2b3c4d5e6f7g8h9i0j1",
    "nom": "Doe",
```

```
"prenom": "John",
"email": "john.doe@example.com",
"telephone": "+22890123456",
"role": "proprietaire",
"firstLogin": false
},
"accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
"refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

## Réponse erreur (400):

```
{
  "message": "Cet email est déjà utilisé"
}
```

## POST /auth/login

### Connexion utilisateur

URL complète: <https://ecologis-api.onrender.com/auth/login>

### Body JSON:

```
{
  "email": "john.doe@example.com",
  "motDePasse": "MotDePasse123!"
}
```

### Exemple cURL:

```
curl -X POST https://ecologis-api.onrender.com/auth/login \
-H "Content-Type: application/json" \
-d '{
  "email": "john.doe@example.com",
  "motDePasse": "MotDePasse123!"
}'
```

## Réponse succès (200):

```
{
  "message": "Connexion réussie",
  "user": {
```

```
{
  "_id": "64f1a2b3c4d5e6f7g8h9i0j1",
  "nom": "Doe",
  "prenom": "John",
  "email": "john.doe@example.com",
  "role": "proprietaire",
  "abonnementId": "64f1a2b3c4d5e6f7g8h9i0j2"
},
"accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
"refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
"abonnement": {
  "type": "premium",
  "prix": 1000,
  "nbResidentsMax": 15
}
}
```

### Réponse erreur (401):

```
{
  "message": "Email ou mot de passe incorrect"
}
```

### POST `/auth/refresh`

#### Renouveler l'access token

**URL complète:** `https://ecologis-api.onrender.com/auth/refresh`

#### Body JSON:

```
{
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

#### Exemple cURL:

```
curl -X POST https://ecologis-api.onrender.com/auth/refresh \
-H "Content-Type: application/json" \
-d '{
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}'
```

### POST `/auth/logout`

**Déconnexion** (*Authentication requise*)

**URL complète:** <https://ecologis-api.onrender.com/auth/logout>

**Headers:** [Authorization: Bearer <access\\_token>](#)

**Exemple cURL:**

```
curl -X POST https://ecologis-api.onrender.com/auth/logout \  
-H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

**POST** [/auth/reset-password](#)

**Changer le mot de passe (premier login)** (*Authentication requise*)

**URL complète:** <https://ecologis-api.onrender.com/auth/reset-password>

**Body JSON:**

```
{  
  "nouveauMotDePasse": "NouveauMotDePasse123!"  
}
```

**POST** [/auth/change-password](#)

**Changer le mot de passe normal** (*Authentication requise*)

**URL complète:** <https://ecologis-api.onrender.com/auth/change-password>

**Body JSON:**

```
{  
  "motDePasseActuel": "AncienMotDePasse123!",  
  "nouveauMotDePasse": "NouveauMotDePasse123!"  
}
```

---

## 2. Gestion des maisons ([/maisons](#))

## POST /maisons

**Créer une maison** (*Authentication + Propriétaire requis*)

**URL complète:** <https://ecologis-api.onrender.com/maisons>

**Body JSON:**

```
{
  "nomMaison": "Villa Sunshine",
  "adresse": {
    "rue": "123 Avenue de la Paix",
    "ville": "Lomé",
    "codePostal": "01BP1234",
    "pays": "Togo"
  },
  "description": "Belle villa avec jardin",
  "tarifKwh": 0.1740
}
```

**Exemple cURL:**

```
curl -X POST https://ecologis-api.onrender.com/maisons \
-H "Authorization: Bearer YOUR_ACCESS_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "nomMaison": "Villa Sunshine",
  "adresse": {
    "rue": "123 Avenue de la Paix",
    "ville": "Lomé",
    "codePostal": "01BP1234"
  },
  "tarifKwh": 0.1740
}'
```

**Réponse succès (201):**

```
{
  "message": "Maison créée avec succès",
  "maison": {
    "_id": "64f1a2b3c4d5e6f7g8h9i0j3",
    "nomMaison": "Villa Sunshine",
    "proprietaireId": "64f1a2b3c4d5e6f7g8h9i0j1",
    "adresse": {
      "rue": "123 Avenue de la Paix",
      "ville": "Lomé",
      "codePostal": "01BP1234",
      "pays": "Togo"
    }
  }
}
```



```
    },
    "tarifKwh": 0.1740,
    "statut": "active"
  }
}
```

## GET /maisons

Lister les maisons (*Authentication requise*)

URL complète: <https://ecologis-api.onrender.com/maisons>

Exemple cURL:

```
curl -X GET https://ecologis-api.onrender.com/maisons \
-H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

Réponse succès (200):

```
{
  "maisons": [
    {
      "_id": "64f1a2b3c4d5e6f7g8h9i0j3",
      "nomMaison": "Villa Sunshine",
      "adresse": {
        "rue": "123 Avenue de la Paix",
        "ville": "Lomé"
      },
      "tarifKwh": 0.1740,
      "listeResidents": [
        {
          "_id": "64f1a2b3c4d5e6f7g8h9i0j4",
          "nom": "Smith",
          "prenom": "Alice",
          "email": "alice.smith@example.com",
          "telephone": "+22890123457"
        }
      ],
      "statut": "active"
    }
  ],
  "count": 1
}
```

## GET /maisons/:id

## Obtenir une maison spécifique (*Authentication requise*)

**URL complète:** <https://ecologis-api.onrender.com/maisons/64f1a2b3c4d5e6f7g8h9i0j3>

**PUT** [/maisons/:id](#)

## Mettre à jour une maison (*Authentication + Propriétaire requis*)

**URL complète:** <https://ecologis-api.onrender.com/maisons/64f1a2b3c4d5e6f7g8h9i0j3>

**DELETE** [/maisons/:id](#)

## Supprimer une maison (*Authentication + Propriétaire requis*)

**URL complète:** <https://ecologis-api.onrender.com/maisons/64f1a2b3c4d5e6f7g8h9i0j3>

**PATCH** [/maisons/:id/tarif](#)

## Mettre à jour le tarif kWh (*Authentication + Propriétaire requis*)

**URL complète:** <https://ecologis-api.onrender.com/maisons/64f1a2b3c4d5e6f7g8h9i0j3/tarif>

### Body JSON:

```
{  
  "tarifKwh": 0.1850  
}
```

**POST** [/maisons/residents/ajouter](#)

## Ajouter un résident à une maison (*Authentication + Propriétaire requis*)

**URL complète:** <https://ecologis-api.onrender.com/maisons/residents/ajouter>

### Body JSON:

```
{
  "maisonId": "64f1a2b3c4d5e6f7g8h9i0j3",
  "residentId": "64f1a2b3c4d5e6f7g8h9i0j4"
}
```

## POST /maisons/residents/retirer

Retirer un résident d'une maison (*Authentication + Propriétaire requis*)

URL complète: <https://ecologis-api.onrender.com/maisons/residents/retirer>

---

## 3. 👥 Gestion des résidents (/residents)

### POST /residents

Ajouter un résident (*Authentication + Propriétaire + Vérification quota requis*)

URL complète: <https://ecologis-api.onrender.com/residents>

Body JSON:

```
{
  "nom": "Smith",
  "prenom": "Alice",
  "email": "alice.smith@example.com",
  "telephone": "+22890123457",
  "maisonId": "64f1a2b3c4d5e6f7g8h9i0j3"
}
```

Exemple cURL:

```
curl -X POST https://ecologis-api.onrender.com/residents \
-H "Authorization: Bearer YOUR_ACCESS_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "nom": "Smith",
  "prenom": "Alice",
  "email": "alice.smith@example.com",
  "telephone": "+22890123457",
  "maisonId": "64f1a2b3c4d5e6f7g8h9i0j3"
}'
```

```
"maisonId": "64f1a2b3c4d5e6f7g8h9i0j3"
}'
```

## Réponse succès (201):

```
{
  "message": "Résident ajouté avec succès",
  "resident": {
    "_id": "64f1a2b3c4d5e6f7g8h9i0j4",
    "nom": "Smith",
    "prenom": "Alice",
    "email": "alice.smith@example.com",
    "telephone": "+22890123457",
    "firstLogin": true
  },
  "credentialsSent": true,
  "temporaryPassword": "TempPass123!"
}
```

## GET /residents

Lister les résidents (*Authentication + Propriétaire requis*)

URL complète: <https://ecologis-api.onrender.com/residents>

## GET /residents/:id

Obtenir un résident spécifique (*Authentication + Propriétaire requis*)

URL complète: <https://ecologis-api.onrender.com/residents/64f1a2b3c4d5e6f7g8h9i0j4>

## PUT /residents/:id

Mettre à jour un résident (*Authentication + Propriétaire requis*)

URL complète: <https://ecologis-api.onrender.com/residents/64f1a2b3c4d5e6f7g8h9i0j4>

## DELETE /residents/:id

Supprimer un résident (*Authentication + Propriétaire requis*)

**URL complète:** <https://ecologis-api.onrender.com/residents/64f1a2b3c4d5e6f7g8h9i0j4>

---

## 4. ⚡ Gestion des consommations (/consommations)

**POST** /consommations

**Enregistrer une consommation** (*Authentication requise*)

**URL complète:** <https://ecologis-api.onrender.com/consommations>

**Body JSON:**

```
{
  "residentId": "64f1a2b3c4d5e6f7g8h9i0j4",
  "maisonId": "64f1a2b3c4d5e6f7g8h9i0j3",
  "kwh": 125.5,
  "mois": 12,
  "annee": 2025,
  "commentaire": "Relevé du compteur principal"
}
```

**Exemple cURL:**

```
curl -X POST https://ecologis-api.onrender.com/consommations \
-H "Authorization: Bearer YOUR_ACCESS_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "residentId": "64f1a2b3c4d5e6f7g8h9i0j4",
  "maisonId": "64f1a2b3c4d5e6f7g8h9i0j3",
  "kwh": 125.5,
  "mois": 12,
  "annee": 2025
}'
```

**Réponse succès (201):**

```
{
  "message": "Consommation enregistrée avec succès",
  "consommation": {
    "_id": "64f1a2b3c4d5e6f7g8h9i0j5",

```

```
{
  "residentId": "64f1a2b3c4d5e6f7g8h9i0j4",
  "maisonId": "64f1a2b3c4d5e6f7g8h9i0j3",
  "kwh": 125.5,
  "mois": 12,
  "annee": 2025,
  "montant": 21.84,
  "statut": "enregistree"
}
```

## GET /consommations/resident/:residentId

**Historique des consommations d'un résident** (*Authentication requise*)

**URL complète:** <https://ecologis-api.onrender.com/consommations/resident/64f1a2b3c4d5e6f7g8h9i0j4>

**Query params:**

- **annee** (optionnel) : Année spécifique
- **mois** (optionnel) : Mois spécifique

**Exemple cURL:**

```
curl -X GET "https://ecologis-api.onrender.com/consommations/resident/64f1a2b3c4d5e6f7g8h9i0j4?annee=2025&mois=12" \
-H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

## GET /consommations/maison/:maisonId

**Consommations d'une maison** (*Authentication requise*)

**URL complète:** <https://ecologis-api.onrender.com/consommations/maison/64f1a2b3c4d5e6f7g8h9i0j3>

## PUT /consommations/:id

**Mettre à jour une consommation** (*Authentication requise*)

**URL complète:** <https://ecologis-api.onrender.com/consommations/64f1a2b3c4d5e6f7g8h9i0j5>

**DELETE** /consommations/:id

**Supprimer une consommation** (*Authentication + Propriétaire requis*)

**URL complète:** <https://ecologis-api.onrender.com/consommations/64f1a2b3c4d5e6f7g8h9i0j5>

---

## 5. 📄 **Gestion des factures (/factures)**

**POST** /factures/generer/:residentId

**Générer une facture pour un résident** (*Authentication requise*)

**URL complète:** <https://ecologis-api.onrender.com/factures/generer/64f1a2b3c4d5e6f7g8h9i0j4>

**Body JSON:**

```
{
  "mois": 12,
  "annee": 2025,
  "fraisFixes": 2.50
}
```

**Exemple cURL:**

```
curl -X POST https://ecologis-api.onrender.com/factures/generer/64f1a2b3c4d5e6f7g8h9i0j4 \
-H "Authorization: Bearer YOUR_ACCESS_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "mois": 12,
  "annee": 2025,
  "fraisFixes": 2.50
}'
```

**Réponse succès (201):**

```
{
  "message": "Facture générée avec succès",
  "facture": {
```

```
{
  "_id": "64f1a2b3c4d5e6f7g8h9i0j6",
  "numeroFacture": "FACT-202512-0001",
  "residentId": "64f1a2b3c4d5e6f7g8h9i0j4",
  "montant": 24.34,
  "statut": "non payée",
  "dateEmission": "2025-12-01T10:00:00.000Z",
  "dateEcheance": "2025-12-31T10:00:00.000Z",
  "details": {
    "kwh": 125.5,
    "prixKwh": 0.1740,
    "fraisFixes": 2.50
  }
}
```

## GET /factures/resident/:residentId

**Factures d'un résident** (*Authentication requise*)

**URL complète:** [https://ecologis-](https://ecologis-api.onrender.com/factures/resident/64f1a2b3c4d5e6f7g8h9i0j4)

[api.onrender.com/factures/resident/64f1a2b3c4d5e6f7g8h9i0j4](https://ecologis-api.onrender.com/factures/resident/64f1a2b3c4d5e6f7g8h9i0j4)

**Query params:**

- **statut** (optionnel) : payée, non payée, en retard
- **annee** (optionnel) : Année spécifique

## GET /factures/maison/:maisonId

**Factures d'une maison** (*Authentication requise*)

**URL complète:** [https://ecologis-](https://ecologis-api.onrender.com/factures/maison/64f1a2b3c4d5e6f7g8h9i0j3)

[api.onrender.com/factures/maison/64f1a2b3c4d5e6f7g8h9i0j3](https://ecologis-api.onrender.com/factures/maison/64f1a2b3c4d5e6f7g8h9i0j3)

## GET /factures/:id

**Obtenir une facture spécifique** (*Authentication requise*)

**URL complète:** [https://ecologis-](https://ecologis-api.onrender.com/factures/64f1a2b3c4d5e6f7g8h9i0j6)

[api.onrender.com/factures/64f1a2b3c4d5e6f7g8h9i0j6](https://ecologis-api.onrender.com/factures/64f1a2b3c4d5e6f7g8h9i0j6)

## PUT /factures/:id/payer

**Marquer une facture comme payée** (*Authentication requise*)



**URL complète:** <https://ecologis-api.onrender.com/factures/64f1a2b3c4d5e6f7g8h9i0j6/payer>

---

## 6. Gestion des abonnements (/abonnements)

**GET** /abonnements

Liste des offres disponibles (*Public*)

**URL complète:** <https://ecologis-api.onrender.com/abonnements>

**Exemple cURL:**

```
curl -X GET https://ecologis-api.onrender.com/abonnements
```

**Réponse succès (200):**

```
{
  "offres": [
    {
      "type": "basic",
      "nom": "Basic",
      "prix": 500,
      "nbResidentsMax": 5,
      "description": "Idéal pour les petites propriétés",
      "fonctionnalites": [
        "Gestion jusqu'à 5 résidents",
        "Génération de factures",
        "Historique des consommations",
        "Notifications WhatsApp"
      ]
    },
    {
      "type": "premium",
      "nom": "Premium",
      "prix": 1000,
      "nbResidentsMax": 15,
      "description": "Parfait pour les propriétés moyennes",
      "fonctionnalites": [
        "Gestion jusqu'à 15 résidents",
        "Génération de factures",
        "Historique des consommations",
        "Notifications WhatsApp",
        "Statistiques avancées",
      ]
    }
  ]
}
```

```
        "Support prioritaire"
    ]
},
{
    "type": "enterprise",
    "nom": "Enterprise",
    "prix": 2000,
    "nbResidentsMax": 50,
    "description": "Pour les grandes propriétés",
    "fonctionnalites": [
        "Gestion jusqu'à 50 résidents",
        "Génération de factures",
        "Historique des consommations",
        "Notifications WhatsApp",
        "Statistiques avancées",
        "Support prioritaire",
        "API personnalisée",
        "Formation incluse"
    ]
}
]
```

## POST /abonnements/souscrire

**Souscrire à un abonnement** (*Authentication + Propriétaire requis*)

**URL complète:** <https://ecologis-api.onrender.com/abonnements/souscrire>

**Body JSON:**

```
{
  "type": "premium"
}
```

## POST /abonnements/renouveler

**Renouveler un abonnement** (*Authentication + Propriétaire requis*)

**URL complète:** <https://ecologis-api.onrender.com/abonnements/renouveler>

## GET /abonnements/actuel

**Obtenir l'abonnement actuel** (*Authentication + Propriétaire requis*)

**URL complète:** <https://ecologis-api.onrender.com/abonnements/actuel>

POST /abonnements/annuler

Annuler un abonnement (*Authentication + Propriétaire requis*)

URL complète: <https://ecologis-api.onrender.com/abonnements/annuler>

GET /abonnements/historique

Historique des abonnements (*Authentication + Propriétaire requis*)

URL complète: <https://ecologis-api.onrender.com/abonnements/historique>

---



## Logique métier

---

### Calcul des factures

1. **Tarif personnalisé** : Chaque maison a son propre tarif kWh (défini par le propriétaire)
2. **Tarif par défaut** : Si aucun tarif n'est défini, utilisation du tarif standard (0.1740 FCFA/kWh)
3. **Frais fixes** : Possibilité d'ajouter des frais fixes (maintenance, etc.)
4. **Formule** :  $\text{Montant} = (\text{kWh} \times \text{Tarif kWh}) + \text{Frais fixes}$

### Gestion des abonnements

- **Quota résidents** : Limite le nombre de résidents selon le type d'abonnement
- **Expiration automatique** : Notifications 7 jours avant expiration
- **Renouvellement** : Extension automatique de la durée

### Notifications automatiques

- **Consommation élevée** : Si la consommation dépasse la moyenne des 3 derniers mois
- **Nouvelle facture** : Envoi WhatsApp automatique lors de la génération
- **Factures en retard** : Rappels automatiques après 30 jours
- **Expiration abonnement** : Alertes 7 jours avant expiration



# Variables importantes

---

## Tarifs par défaut

- **Prix kWh standard** : 0.1740 FCFA/kWh
- **Frais fixes** : 0 FCFA (configurable par propriétaire)
- **Échéance facture** : 30 jours après émission

## Limites d'abonnement

- **Basic** : 5 résidents max (500 FCFA/mois)
  - **Premium** : 15 résidents max (1000 FCFA/mois)
  - **Enterprise** : 50 résidents max (2000 FCFA/mois)
- 



## Instructions pour tester l'API avec Postman

---

### 1. Configuration de l'environnement Postman

Créez un environnement avec ces variables :

```
BASE_URL: https://ecologis-api.onrender.com
ACCESS_TOKEN: (vide au début)
REFRESH_TOKEN: (vide au début)
USER_ID: (vide au début)
```

### 2. Collection Postman recommandée

#### Authentification

- POST {{BASE\_URL}}/auth/register - Créer un compte
- POST {{BASE\_URL}}/auth/login - Se connecter
- POST {{BASE\_URL}}/auth/refresh - Renouveler le token
- POST {{BASE\_URL}}/auth/logout - Se déconnecter

## Gestion des maisons

- POST {{BASE\_URL}}/maisons - Créer une maison
- GET {{BASE\_URL}}/maisons - Lister les maisons
- GET {{BASE\_URL}}/maisons/:id - Obtenir une maison
- PUT {{BASE\_URL}}/maisons/:id - Modifier une maison
- DELETE {{BASE\_URL}}/maisons/:id - Supprimer une maison

## Gestion des résidents

- POST {{BASE\_URL}}/residents - Ajouter un résident
- GET {{BASE\_URL}}/residents - Lister les résidents
- GET {{BASE\_URL}}/residents/:id - Obtenir un résident
- PUT {{BASE\_URL}}/residents/:id - Modifier un résident
- DELETE {{BASE\_URL}}/residents/:id - Supprimer un résident

## Gestion des consommations

- POST {{BASE\_URL}}/consommations - Enregistrer une consommation
- GET {{BASE\_URL}}/consommations/resident/:id - Historique résident
- GET {{BASE\_URL}}/consommations/maison/:id - Consommations maison

## Gestion des factures

- POST {{BASE\_URL}}/factures/generer/:residentId - Générer une facture
- GET {{BASE\_URL}}/factures/resident/:id - Factures d'un résident
- GET {{BASE\_URL}}/factures/maison/:id - Factures d'une maison
- PUT {{BASE\_URL}}/factures/:id/payer - Marquer comme payée

## Gestion des abonnements

- GET {{BASE\_URL}}/abonnements - Liste des offres
- POST {{BASE\_URL}}/abonnements/souscrire - Souscrire
- GET {{BASE\_URL}}/abonnements/actuel - Abonnement actuel

## 3. Workflow de test recommandé

1. Créer un compte propriétaire (POST /auth/register)
2. Se connecter (POST /auth/login) → Récupérer les tokens
3. Créer une maison (POST /maisons)
4. Ajouter un résident (POST /residents)
5. Enregistrer une consommation (POST /consommations)
6. Générer une facture (POST /factures/generer/:residentId)

## 4. Gestion automatique des tokens

Dans Postman, utilisez ce script de test pour automatiser la gestion des tokens :

```
// Dans l'onglet "Tests" de la requête de login
if (pm.response.code === 200) {
  const response = pm.response.json();
  pm.environment.set("ACCESS_TOKEN", response.accessToken);
  pm.environment.set("REFRESH_TOKEN", response.refreshToken);
  pm.environment.set("USER_ID", response.user._id);
}

// Dans l'onglet "Pre-request Script" des autres requêtes
pm.request.headers.add({
  key: "Authorization",
  value: "Bearer " + pm.environment.get("ACCESS_TOKEN")
});
```



## Guide d'utilisation pour Flutter

### 1. Configuration de l'API

```
class ApiConfig {
  static const String baseUrl = 'https://ecologis-api.onrender.com';
  static const Duration timeout = Duration(seconds: 30);

  // Headers par défaut
  static Map<String, String> get defaultHeaders => {
    'Content-Type': 'application/json',
    'Accept': 'application/json',
  }
}
```

```
};  
}
```

## 2. Service d'authentification

```
class AuthService {  
    static String? accessToken;  
    static String? refreshToken;  
  
    static Future<bool> login(String email, String password) async {  
        try {  
            final response = await http.post(  
                Uri.parse('${ApiConfig.baseUrl}/auth/login'),  
                headers: ApiConfig.defaultHeaders,  
                body: json.encode({  
                    'email': email,  
                    'motDePasse': password,  
                })),  
            );  
  
            if (response.statusCode == 200) {  
                final data = json.decode(response.body);  
                accessToken = data['accessToken'];  
                refreshToken = data['refreshToken'];  
                return true;  
            }  
            return false;  
        } catch (e) {  
            print('Erreur de connexion: $e');  
            return false;  
        }  
    }  
  
    static Future<String?> getValidToken() async {  
        if (accessToken != null) {  
            // Vérifier si le token est expiré  
            // Si oui, utiliser refreshToken  
            return accessToken;  
        }  
        return null;  
    }  
}
```

## 3. Service API générique

```
class ApiService {  
    static Future<Map<String, String>> getHeaders() async {
```

```

    final token = await AuthService.getValidToken();
    return {
      ...ApiConfig.defaultHeaders,
      if (token != null) 'Authorization': 'Bearer $token',
    };
  }

  static Future<http.Response> get(String endpoint) async {
    final headers = await getHeaders();
    return await http.get(
      Uri.parse('${ApiConfig.baseUrl}$endpoint'),
      headers: headers,
    ).timeout(ApiConfig.timeout);
  }

  static Future<http.Response> post(String endpoint, Map<String, dynamic> data)
  async {
    final headers = await getHeaders();
    return await http.post(
      Uri.parse('${ApiConfig.baseUrl}$endpoint'),
      headers: headers,
      body: json.encode(data),
    ).timeout(ApiConfig.timeout);
  }
}

```

## 4. Gestion des erreurs

```

class ApiException implements Exception {
  final String message;
  final int statusCode;

  ApiException(this.message, this.statusCode);

  static ApiException fromResponse(http.Response response) {
    try {
      final body = json.decode(response.body);
      return ApiException(body['message'] ?? 'Erreur inconnue',
response.statusCode);
    } catch (e) {
      return ApiException('Erreur de parsing', response.statusCode);
    }
  }
}

// Utilisation
try {
  final response = await ApiService.get('/maisons');
  if (response.statusCode == 200) {
    // Traitement du succès
  } else {
    throw ApiException.fromResponse(response);
  }
}

```



```
}  
} on ApiException catch (e) {  
    print('Erreur API: ${e.message} (${e.statusCode})');  
} catch (e) {  
    print('Erreur réseau: $e');  
}
```



# Résumé pour le développeur Flutter

## Points clés à retenir :

1. **Authentification obligatoire** : Toutes les routes (sauf `/auth/register`, `/auth/login`, `/auth/refresh`, `/abonnements`) nécessitent un token JWT valide
2. **Gestion des rôles** : Vérifier le rôle utilisateur avant d'afficher certaines fonctionnalités
3. **Refresh automatique** : Implémenter la logique de refresh token pour maintenir la session
4. **Gestion des erreurs** : Traiter les codes de statut HTTP et les messages d'erreur
5. **Notifications push** : L'API envoie automatiquement des notifications FCM pour les événements importants
6. **Calculs automatiques** : Les montants des factures sont calculés automatiquement par l'API

## Workflow recommandé :

1. Authentification → Récupération des tokens
2. Vérification du rôle utilisateur
3. Chargement des données selon les permissions
4. Gestion des erreurs et des tokens expirés
5. Mise à jour en temps réel via WebSocket (optionnel)

## Sécurité :

- Stocker les tokens de manière sécurisée (SecureStorage)
- Ne jamais exposer les tokens dans les logs

- Vérifier les permissions côté client ET côté serveur
  - Implémenter la déconnexion automatique en cas d'erreur 401
- 

## Support et contact

---

 **URL de production** : <https://ecologis-api.onrender.com>

 **Support technique** : Contactez l'équipe de développement backend

 **Statut de l'API** : Production - Stable

 **Version** : 1.0.0

 **Dernière mise à jour** : Décembre 2025

 **Environnement** : Render (Production)