



Cette API permet de générer un token à partir d'un login (celui de la secrétaire) et de son mot de passe.

L'URL de base utilisée pour notre API d'authentification est la suivante :

POST /authapi

Requête :

Body :

Réponse :

1

```
I6InNlY3JldGFpcmUxIiwizXhwIjo1MTM0OTA0NjE2MDAwfQ.SDzDxv-Rge  
2gu304VQguIgg0PIWj0mGsJtk7icJNcE0"  
}
```

GET /authapi

Vérifie l'authentification de la personne connectée. Renvoie l'identifiant ainsi que le mot de passe de cette personne.

Réponse

Renvoie un document JSON avec les champs suivants :

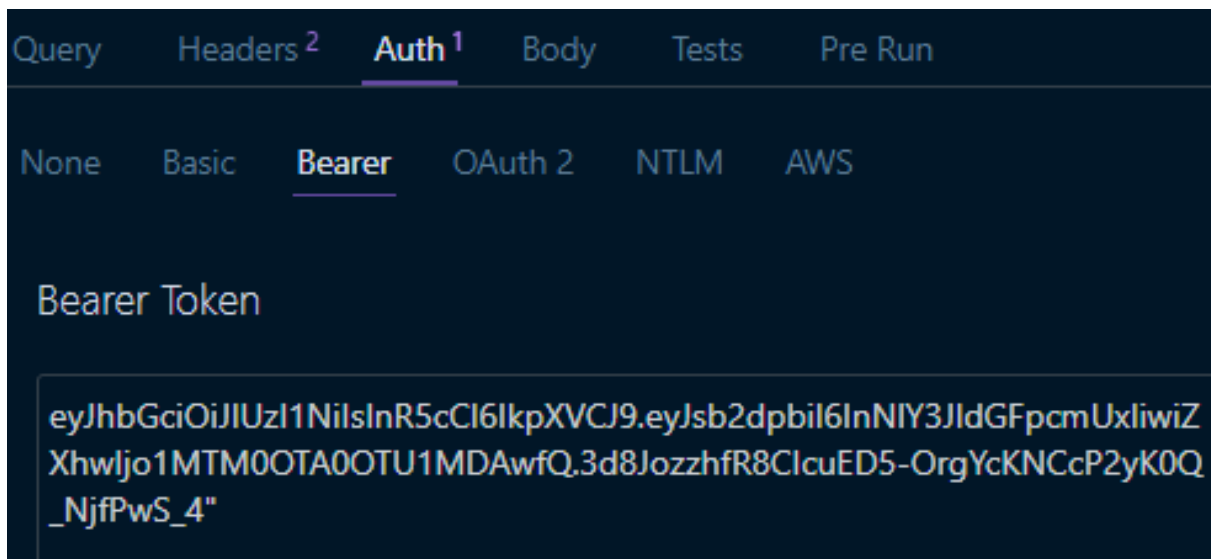
- `login` : identifiant de connexion de la personne souhaitant s'authentifier.
- `mdp` : le mot de passe de la personne s'authentifant.

Exemple

Requête :

```
GET /authapi
```

Token généré après la requête `POST` à mettre dans Auth, dans Bearer :



Réponse :

```
{  
  "login": "secretaire1",
```

```
"mdp": "password1234!"  
}
```

Erreurs

Cette API peut renvoyer les erreurs suivantes :

- **405 Bad Request** : La méthode n'est pas autorisée.
- **401 Unauthorized** : Une condition permettant d'avoir accès aux données n'est pas remplie. Ici, avec le cas de **GET** où le token ou le mot de passe ou le login est invalide.
- **404 Not Found** : La ressource de la requête est introuvable.
- **500 Internal Server Error** : Une erreur inattendue s'est produite dans le serveur.



API Cabmed Documentation

Description

Cette API permet de gérer un cabinet médical composé de :

- Usagers
- Médecins
- Statistiques
- Consultations

Base URL

L'URL de base pour toutes les requêtes de l'API est :

<https://api-patientele-cabmed.alwaysdata.net/cabmed>

Endpoints

- **Pour les usagers**

GET /usagers

Retourne la liste de tous les usagers.

Réponse

Retourne un objet JSON avec les propriétés suivantes :

- `status_code` : Le statut du résultat de la requête
- `status_message` : Le message du statut du résultat de la requête
- `data` : Un tableau de données contenant les propriétés suivantes :
 - `id_usager` : L'identifiant unique de l'utilisateur
 - `civilite` : La civilité de l'utilisateur
 - `nom` : Le nom de l'utilisateur
 - `prenom` : Le prénom de l'utilisateur

- `sexe` : Le sexe de l'utilisateur
- `adresse` : L'adresse de l'utilisateur
- `code_postal` : Le code postal de l'utilisateur
- `ville` : La ville où l'utilisateur habite
- `date_nais` : La date de naissance de l'utilisateur
- `lieu_nais` : Le lieu de naissance de l'utilisateur
- `num_secu` : Le numéro de sécurité de l'utilisateur
- `id_medecin` : L'identifiant du médecin référant de l'utilisateur

Exemple

Requête :

```
GET /usagers
```

Réponse:

```
{
  "status_code": 200,
  "status_message": "OK (récupération patient(s))",
  "data": [
    {
      "id_usager": 10,
      "civilite": "Mme",
      "nom": "Delamer",
      "prenom": "Arielle",
      "sexe": "F",
      "adresse": "101, Vague de l'Atlantis",
      "code_postal": "31000",
      "ville": "Toulouse",
      "date_nais": "1995-08-26",
      "lieu_nais": "Atlantica",
      "num_secu": "112233445566678",
      "id_medecin": 1
    },
    {
```

```
    "id_usager": 12,  
    "civilite": "M.",  
    "nom": "Dumond",  
    "prenom": "Armand",  
    "sexe": "H",  
    "adresse": "85, Square de la Couronne",  
    "code_postal": "91120",  
    "ville": "Palaiseau",  
    "date_nais": "1952-05-14",  
    "lieu_nais": "Nantes",  
    "num_secu": "112233445566778",  
    "id_medecin": 1  
  }  
]  
}
```

GET /usagers/<id>

Retourne l'utilisateur d'identifiant id.

Paramètres

- `id` : L'ID de l'utilisateur à récupérer.

Réponse

Retourne un objet JSON avec les propriétés suivantes :

- `status_code` : Le statut du résultat de la requête
- `status_message` : Le message du statut du résultat de la requête
- `data` : Un tableau de données contenant les propriétés suivantes :
 - `id_usager` : L'identifiant unique de l'utilisateur
 - `civilite` : La civilité de l'utilisateur
 - `nom` : Le nom de l'utilisateur
 - `prenom` : Le prénom de l'utilisateur
 - `sexe` : Le sexe de l'utilisateur
 - `adresse` : L'adresse de l'utilisateur

- `code_postal` : Le code postal de l'utilisateur
- `ville` : La ville où l'utilisateur habite
- `date_nais` : La date de naissance de l'utilisateur
- `lieu_nais` : Le lieu de naissance de l'utilisateur
- `num_secu` : Le numéro de sécurité de l'utilisateur
- `id_medecin` : L'identifiant du médecin référant de l'utilisateur

Exemple

Requête :

```
GET /usagers/10
```

Réponse:

```
{
  "status_code": 200,
  "status_message": "OK (récupération patient(s))",
  "data": [
    {
      "id_usager": 10,
      "civilite": "Mme",
      "nom": "Delamer",
      "prenom": "Arielle",
      "sexe": "F",
      "adresse": "101, Vague de l'Atlantis",
      "code_postal": "31000",
      "ville": "Toulouse",
      "date_nais": "1995-08-26",
      "lieu_nais": "Atlantica",
      "num_secu": "112233445566678",
      "id_medecin": 1
    }
  ]
}
```

POST /usagers

Crée un usager.

Réponse

Retourne un objet JSON avec les propriétés suivantes :

- `status_code` : Le statut du résultat de la requête
- `status_message` : Le message du statut du résultat de la requête
- `data` : L'identifiant unique de l'utilisateur créé

Exemple

Requête :

```
POST /usagers
```

Body :

```
{
  "civilite": "Mme",
  "nom": "Dumond",
  "prenom": "Armandia",
  "sexe": "F",
  "adresse": "85, Square de la Couronne",
  "code_postal": "91120",
  "ville": "Palaiseau",
  "date_nais": "29/05/1952",
  "lieu_nais": "Nantes",
  "num_secu": "3524364758",
  "id_medecin": "1"
}
```

Réponse:

```
{
  "status_code": 201,
  "status_message": "Patient créé",
  "data": "13"
}
```


PATCH /usagers/<id>

Modifie l'utilisateur d'identifiant id.

Paramètres

- **id** : L'ID de l'utilisateur à modifier.

Réponse

Retourne un objet JSON avec les propriétés suivantes :

- **status_code** : Le statut du résultat de la requête
- **status_message** : Le message du statut du résultat de la requête
- **data** : Un tableau de données contenant les propriétés suivantes :
 - **id_usager** : L'identifiant unique de l'utilisateur modifié
 - **civilite** : La civilité de l'utilisateur modifié
 - **nom** : Le nom de l'utilisateur modifié
 - **prenom** : Le prénom de l'utilisateur modifié
 - **sexe** : Le sexe de l'utilisateur modifié
 - **adresse** : L'adresse de l'utilisateur modifié
 - **code_postal** : Le code postal de l'utilisateur modifié
 - **ville** : La ville où l'utilisateur habite modifié
 - **date_nais** : La date de naissance de l'utilisateur modifié
 - **lieu_nais** : Le lieu de naissance de l'utilisateur modifié
 - **num_secu** : Le numéro de sécurité de l'utilisateur modifié
 - **id_medecin** : L'identifiant du médecin référant de l'utilisateur modifié

Exemple

Requête :

```
PATCH /usagers/16
```

Body :

```
{
  "civilite": "Mme",
  "prenom": "Jeanne"
}
```

Réponse :

```
{
  "status_code": 200,
  "status_message": "Patient modifié",
  "data": {
    "id_usager": 16,
    "civilite": "Mme",
    "nom": "Dark",
    "prenom": "Jeanne",
    "sexe": "F",
    "adresse": "101, rue de la roulade",
    "code_postal": "91120",
    "ville": "Routier",
    "date_nais": "1952-05-29",
    "lieu_nais": "Nantes",
    "num_secu": "123245643221",
    "id_medecin": 1
  }
}
```

DELETE /usagers/<id>

Supprime un usager d'ID id.

Réponse

Retourne un objet JSON avec les propriétés suivantes :

- **status_code** : Le statut du résultat de la requête
- **status_message** : Le message du statut du résultat de la requête

Exemple

Requête :

```
DELETE /usagers/1
```

Réponse :

```
{
  "status_code": 200,
  "status_message": "Patient supprimé",
  "data": "1"
}
```

- **Pour les médecins**

GET /medecins

Retourne la liste de tous les médecins.

Réponse

Retourne un objet JSON avec les propriétés suivantes :

- `status_code` : Le statut du résultat de la requête
- `status_message` : Le message du statut du résultat de la requête
- `data` : Un tableau de données contenant les propriétés suivantes :
 - `id_medecin` : L'identifiant unique du médecin
 - `civilite` : La civilité du médecin (M. ou Mme)
 - `nom` : Le nom du médecin
 - `prenom` : Le prénom du médecin

Exemple

Requête :

```
GET /medecins
```

Réponse :

```
{
  "status_code": 200,
```

```

    "status_message": "OK",
    "data": [
      {
        "id_medecin": 1,
        "civilite": "M",
        "nom": "Mermoz",
        "prenom": "Jean"
      },
      {
        "id_medecin": 3,
        "civilite": "M",
        "nom": "Latecoere",
        "prenom": "Georges"
      }
    ]
  }

```

GET /medecins/<id>

Retourne le médecin d'identifiant id.

Paramètres

- **id** : L'ID du médecin à récupérer.

Réponse

Retourne un objet JSON avec les propriétés suivantes :

- **status_code** : Le statut du résultat de la requête
- **status_message** : Le message du statut du résultat de la requête
- **data** : Un tableau de données contenant les propriétés suivantes :
 - **id_medecin** : L'identifiant unique du médecin
 - **civilite** : La civilité du médecin (M. ou Mme)
 - **nom** : Le nom du médecin
 - **prenom** : Le prénom du médecin

Exemple

Requête :

```
GET /medecins/1
```

Réponse :

```
{
  "status_code": 200,
  "status_message": "OK",
  "data": [
    {
      "id_medecin": 1,
      "civilite": "M",
      "nom": "Mermoz",
      "prenom": "Jean"
    }
  ]
}
```

POST /medecins

Crée un médecin.

Réponse

Retourne un objet JSON avec les propriétés suivantes :

- `status_code` : Le statut du résultat de la requête
- `status_message` : Le message du statut du résultat de la requête
- `data` : Un tableau de données contenant les propriétés suivantes :
 - `id_medecin` : L'identifiant unique du médecin créé

Exemple

Requête :

```
POST /medecins
```

Body :

```
{
  "civilite": "M.",
  "nom": "Dupond",
  "prenom": "Gérard"
}
```

Réponse :

```
{
  "status_code": 201,
  "status_message": "Médecin rajouté",
  "data": "53"
}
```

PATCH /medecins/<id>

Modifie le médecin d'identifiant id.

Paramètres

- **id** : L'ID du médecin à modifier.

Réponse

Retourne un objet JSON avec les propriétés suivantes :

- **status_code** : Le statut du résultat de la requête
- **status_message** : Le message du statut du résultat de la requête
- **data** : Un tableau de données contenant les propriétés suivantes :
 - **id_medecin** : L'identifiant unique du médecin modifié
 - **civilite** : La civilité du médecin modifié
 - **nom** : Le nom du médecin modifié
 - **prenom** : Le prénom du médecin modifié

Exemple

Requête :

PATCH /medecins/11

Body :

```
{
  "prenom": "Gordon"
}
```

Réponse :

```
{
  "status_code": 200,
  "status_message": "Médecin modifié",
  "data": {
    "id_medecin": 11,
    "civilite": "M.",
    "nom": "Ramsay",
    "prenom": "Gordon"
  }
}
```

DELETE /medecins/<id>

Supprime le médecin d'identifiant id.

Paramètres

- `id` : L'ID du médecin à supprimer.

Réponse

Retourne un objet JSON avec les propriétés suivantes :

- `status_code` : Le statut du résultat de la requête
- `status_message` : Le message du statut du résultat de la requête
- `data` : Un tableau de données contenant les propriétés suivantes :
 - `id_medecin` : L'identifiant unique du médecin supprimé

Exemple

Requête :

```
DELETE /medecins/1
```

Réponse :

```
{
  "status_code": 200,
  "status_message": "Médecin supprimé",
  "data": 1
}
```

- **Pour les consultations**

GET /consultations

Retourne la liste de toutes les consultations.

Réponse

Retourne un objet JSON avec les propriétés suivantes :

- `status_code` : Le statut du résultat de la requête
- `status_message` : Le message du statut du résultat de la requête
- `data` : Un tableau de données contenant les propriétés suivantes :
 - `id_consult` : L'identifiant unique de la consultation
 - `id_medecin` : L'identifiant unique du médecin
 - `id_usager` : L'identifiant unique de l'utilisateur
 - `date_consult` : La date de la consultation
 - `heure_consult` : L'heure de la consultation
 - `duree_consult` : La durée de la consultation
 - `nom_usager` : Le nom de l'utilisateur
 - `nom_medecin` : Le nom du médecin

Exemple

Requête :

GET /consultations

Réponse :

```
{
  "status_code": 200,
  "status_message": "OK (récupération consultation
(s))",
  "data": [
    {
      "id_consult": 1,
      "id_medecin": 3,
      "id_usager": 10,
      "date_consult": "2024-10-24",
      "heure_consult": "11:30:00",
      "duree_consult": 30,
      "nom_usager": "Delamer",
      "nom_medecin": "Mermoz"
    },
    {
      "id_consult": 2,
      "id_medecin": 8,
      "id_usager": 12,
      "date_consult": "2022-10-24",
      "heure_consult": "12:30:00",
      "duree_consult": 45,
      "nom_usager": "Dumond",
      "nom_medecin": "Latecoere"
    }
  ]
}
```

GET /consultations/<id>

Retourne la consultation d'identifiant id.

Paramètres

- **id** : L'ID de la consultation à récupérer.

Réponse

Retourne un objet JSON avec les propriétés suivantes :

- `status_code` : Le statut du résultat de la requête
- `status_message` : Le message du statut du résultat de la requête
- `data` : Un tableau de données contenant les propriétés suivantes :
 - `id_consult` : L'identifiant unique de la consultation
 - `id_medecin` : L'identifiant unique du médecin
 - `id_usager` : L'identifiant unique de l'utilisateur
 - `date_consult` : La date de la consultation
 - `heure_consult` : L'heure de la consultation
 - `duree_consult` : La durée de la consultation
 - `nom_usager` : Le nom de l'utilisateur
 - `nom_medecin` : Le nom du médecin

Exemple

Requête :

```
GET /consultations/1
```

Réponse :

```
{
  "status_code": 200,
  "status_message": "OK (récupération consultation(s))",
  "data": [
    {
      "id_consult": 1,
      "id_medecin": 3,
      "id_usager": 10,
      "date_consult": "2024-10-24",
      "heure_consult": "11:30:00",
      "duree_consult": 30,
      "nom_usager": "Delamer",

```

```
        "nom_medecin": "Mermoz"
    }
]
}
```

POST /consultations

Crée une consultation.

Réponse

Retourne un objet JSON avec les propriétés suivantes :

- `status_code` : Le statut du résultat de la requête
- `status_message` : Le message du statut du résultat de la requête
- `data` : Un tableau de données contenant les propriétés suivantes :
 - `id_consult` : L'identifiant unique de la consultation créée

Exemple

Requête :

```
POST /consultations
```

Body :

```
{
  "id_medecin" : 1,
  "id_usager" : 1,
  "date_consult" : "2024-07-01",
  "heure_consult": "09:30:00",
  "duree_consult" : 45
}
```

Réponse :

```
{
  "status_code": 201,
  "status_message": "Consultation créée",
}
```

```
"data": "3"
}
```

PATCH /consultations/<id>

Modifie la consultation d'identifiant id.

Paramètres

- **id** : L'ID de la consultation à modifier.

Réponse

Retourne un objet JSON avec les propriétés suivantes :

- **status_code** : Le statut du résultat de la requête
- **status_message** : Le message du statut du résultat de la requête
- **data** : Un tableau de données contenant les propriétés suivantes :
 - **id_consult** : L'identifiant unique de la consultation modifiée
 - **date_consult** : La date de la consultation modifiée
 - **heure_consult** : L'heure de la consultation modifiée
 - **duree_consult** : La durée de la consultation modifiée
 - **id_medecin** : L'identifiant du médecin
 - **id_usager** : L'identifiant de l'utilisateur

Exemple

Requête :

```
PATCH /consultations/3
```

Body :

```
{
  "id_usager": "16",
  "id_medecin": "11",
  "date_consult": "12/10/26",
  "heure_consult": "12:30",
```

```
"duree_consult": "45"
}
```

Réponse :

```
{
  "status_code": 200,
  "status_message": "Consultation modifiée",
  "data": {
    "id_consult": 26,
    "date_consult": "2012-10-26",
    "heure_consult": "12:30:00",
    "duree_consult": 45,
    "id_medecin": 11,
    "id_usager": 16
  }
}
```

DELETE /consultations/<id>

Supprime la consultation d'identifiant id.

Paramètres

- **id** : L'ID de la consultation à supprimer.

Réponse

Retourne un objet JSON avec les propriétés suivantes :

- **status_code** : Le statut du résultat de la requête
- **status_message** : Le message du statut du résultat de la requête
- **data** : Un tableau de données contenant les propriétés suivantes :
 - **id_medecin** : L'identifiant unique de la consultation supprimée

Exemple

Requête :

```
DELETE /consultation/3
```

Réponse :

```
{
  "status_code": 200,
  "status_message": "Consultation supprimée",
  "data": "3"
}
```

- **Pour les statistiques**

GET /stats/usagers

Retourne la liste des statistiques des usagers.

Réponse

Retourne un objet JSON avec les propriétés suivantes :

- **status_code** : Le statut du résultat de la requête
- **status_message** : Le message du statut du résultat de la requête
- **data** : Un tableau de données contenant les propriétés suivantes :
 - **NbFemmesMoins25Ans** : Le nombre de femmes ayant moins de 25 ans
 - **NbFemmesMilieu** : Le nombre de femmes âgées entre 25 et 50 ans
 - **NbFemmesPlus50Ans** : Le nombre de femmes ayant plus de 50 ans
 - **NbHommesMoins25Ans** : Le nombre d'hommes ayant moins de 25 ans
 - **NbHommesMilieu** : Le nombre d'hommes âgées entre 25 et 50 ans
 - **NbHommesPlus50Ans** : Le nombre d'hommes ayant plus de 50 ans

Exemple

Requête :

```
GET /stats/usagers
```

Réponse :

```
{
  "status_code": 200,
```

```
"status_message": "OK",
"data": {
  "NbFemmesMoins25Ans": [
    {
      "NbFemmesMoins25ans": 0
    }
  ],
  "NbFemmesMilieu": [
    {
      "NbFemmesMilieu": 1
    }
  ],
  "NbFemmesPlus50Ans": [
    {
      "NbFemmesPlus50ans": 1
    }
  ],
  "NbHommesMoins25Ans": [
    {
      "NbHommesMoins25ans": 0
    }
  ],
  "NbHommesMilieu": [
    {
      "NbHommesMilieu": 0
    }
  ],
  "NbHommesPlus50Ans": [
    {
      "NbHommesPlus50ans": 2
    }
  ]
}
```

GET /stats/médecins

Retourne la liste des statistiques des médecins.

Réponse

Retourne un objet JSON avec les propriétés suivantes :

- `status_code` : Le statut du résultat de la requête
- `status_message` : Le message du statut du résultat de la requête
- `data` : Un tableau de données contenant les propriétés suivantes :
 - `NbHeures` : Le nombre total d'heures réalisé par le médecin
 - `NomMedecin` : Le nom du médecin concerné

Exemple

Requête :

```
GET /stats/medecins
```

Réponse :

```
{
  "status_code": 200,
  "status_message": "OK",
  "data": [
    {
      "NbHeures": "0.5000",
      "NomMedecin": "Durand"
    },
    {
      "NbHeures": "0.7500",
      "NomMedecin": "Pierre"
    }
  ]
}
```

Erreurs

Cette API peut renvoyer les erreurs suivantes :

- `405 Bad Request` : La méthode n'est pas autorisée.

- `401 Unauthorized` : Une condition permettant d'avoir accès aux données n'est pas remplie.
- `404 Not Found` : La ressource de la requête est introuvable.
- `500 Internal Server Error` : Une erreur inattendue s'est produite dans le serveur.