

Projet : Gestion de cabinet médical

Table des matières

I.	Présentation générale.....	2
II.	Présentation des ressources	2
a.	Médecins	2
b.	Usagers	2
c.	Rendez-vous	3
d.	Statistiques	3
e.	Authentification.....	3
III.	URL Path Parameters et réécriture de lien	3
IV.	Format des réponses.....	4
V.	Fonctionnalités proposées par nos API	5
A.	Médecins	5
B.	Usagers	10
C.	Rendez-vous	16
D.	Statistiques	21
E.	Authentification.....	22

I. Présentation générale

Nos API offrent aux secrétaires d'un cabinet médical la possibilité d'interagir avec les médecins, les patients et les rendez-vous du cabinet après s'être authentifiées pour garantir la sécurité des données de leur cabinet. Conformément aux normes du protocole HTTP, toutes nos API prennent en charge les méthodes standards telles que GET, POST, PATCH, PUT et DELETE, permettant ainsi de respecter le principe CRUD (Create, Read, Update, Delete) des bases de données.

Elles offrent ainsi aux secrétaires la capacité d'effectuer les actions suivantes sur les médecins, patients et les rendez-vous : affichage, création, modification et suppression. Ces dernières peuvent également obtenir des statistiques sur les usagers et les médecins.

II. Présentation des ressources

Les caractéristiques d'un médecin, d'un usager et d'un rendez-vous sont liées aux champs de la table de données qui les stockent.

a. Médecins

Champs	Type	Rôle
id_medecin	Entier	Identifiant unique pour chacun des médecins. Il est auto-incrémenté.
civilite	Chaine	La civilité (M. ou Mme.) du dit médecin (trigger MySQL).
nom	Chaine	Nom du médecin.
prenom	Chaine	Prénom du médecin.

b. Usagers

Champs	Type	Rôle
id_usager	Entier	Identifiant unique pour chaque patient. Il est auto-incrémenté.
civilite	Chaine	La civilité (M. ou Mme.) du dit patient (trigger MySQL).
nom	Chaine	Nom du patient.
prenom	Chaine	Prénom du patient.
sexe	Caractère	Sexe (H ou F) du patient (trigger MySQL).
adresse	Chaine	Adresse de l'usager.
ville	Chaine	Ville de l'usager.
code_postal	Caractères	Limité à 5 caractères. Code postal de l'usager.
date_naissance	Date	Date de naissance de l'usager. Évidemment un patient ne peut être né après la date du jour (trigger MySQL).
lieu_naissance	Chaine	Lieu de naissance du patient.
num_secu_sociale	Chaine	Limité à 15 caractères. Numéro de sécurité sociale (trigger MySQL).
id_medecin	Entier	Clé étrangère sur la table médecin. Elle indique le médecin référent du patient en question.

c. Rendez-vous

Champs	Type	Rôle
id_rendezvous	Entier	Identifiant unique pour chacun des rendez-vous. Il est auto-incrémenté.
date_consultation	Date	Date de la consultation. Un rendez-vous ne peut pas être fixé avant la date du jour ni le week-end.
heure_consultation	Time	Créneau de la consultation. Avant 08:00, entre 12:00 et 14:00 et après 17:00 les rendez-vous ne sont pas disponibles. De plus, deux créneaux pour le même médecin ne peuvent pas être superposés.
duree_consultation	Entier	Durée de la consultation en minutes. Un rendez-vous ne peut pas durer moins de 15 min, ni durer plus d'une heure. (trigger MySQL)
id_usager	Entier	Identifiant du patient qui a un rendez-vous.
id_medecin	Entier	Identifiant du médecin qui consulte l'utilisateur.

d. Statistiques

Des statistiques sur les médecins et sur les patients sont également disponibles :

- La part de femme et d'homme ayant plus ou moins de 25 ans parmi les patients.
- La durée totale des consultations de chaque médecin.

e. Authentification

À travers notre API d'authentification, nous assurons la sécurité des données en attribuant un jeton aux utilisateurs autorisés, issus de la base de données utilisateur. Ce jeton doit être transmis aux API des médecins, des utilisateurs, des rendez-vous et des statistiques, puis validé pour garantir que seules les personnes autorisées peuvent accéder aux données du cabinet médical.

Champs	Type	Rôle
user	Chaîne	Identifiant d'un user autorisé à accéder aux données.
mdp	Chaîne	Mot de passe d'un user autorisé à accéder aux données.

Le login et le mot de passe pour se connecter à notre API Authentification est :

```
"login": "secretaire1",  
"mdp": "password1234!"
```

III. URL Path Parameters et réécriture de lien

Nos API utilisent les Path Parameters (paramètres dans le chemin d'accès) pour envoyer les identifiants des ressources.

L'accès à une ressource se fait donc sous cette forme :

<https://beaujouripiceanu.alwaysdata.net/{medecins ou usagers ou consultations}/123>

En sachant que pour le cas des statistiques, on y accède ainsi :

<https://beaujouripiceanu.alwaysdata.net/statistiques/{usagers ou medecins}>

Ce format respecte les recommandations REST et nécessite de mettre en place un mécanisme de réécriture d'URL du côté serveur plus un traitement du chemin en PHP « .htaccess ».

IV. Format des réponses

Toutes les réponses sont au format JSON et le format de la réponse est toujours le même.

status_code : code de retour (Compris dans le tableau suivant)

status_message : message personnalisé en fonction du status_code

data : *null* si pas de données retournées | tableau contenant les caractéristiques de la ressource

Status_code	Signification	Description
200	OK	La requête a réussi.
201	Created	La requête a réussi et une nouvelle ressource a été créée
400	Bad Request	Syntaxe de la requête non conforme
401	Unauthorized	Une authentification est nécessaire
403	Forbidden	Accès non autorisé à la ressource demandée
404	Not Found	Ressource demandée non trouvée
405	Method Not Allowed	La méthode demandée n'est pas autorisée pour les données demandées
409	Conflict	La requête entre en conflit avec l'état actuel du serveur
500	Internal Server Error	Erreur interne au serveur

Les codes d'erreurs 4xx sont liés à une erreur du côté client tandis que les codes d'erreur 5xx sont liés à des erreurs côté serveur.

Exemple de réponse à succès dans le cas d'un GET sur l'API Usagers.

```
{
  "status_code": 200,
  "status_message": "Succès",
  "data": [
    {
      "id_usager": 8,
      "civilite": "M",
      "nom": "Dubois",
      "prenom": "Pierre",
      "sexe": "",
      "adresse": "123 Rue de la Republique",
      "ville": "Paris",
      "code_postal": "75001",
      "date_naissance": "1980-05-15",
      "lieu_naissance": "Lyon",
      "num_secu_sociale": "123456789012345",
      "id_medecin": 21
    }
  ]
  ... (autres usagers)
}
```

V. Fonctionnalités proposées par nos API

A. Médecins

a. Obtenir tous les médecins

GET http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/medecins

Permet d'obtenir tous les médecins en ordre alphabétique du nom.

1. Requêtes

Aucun paramètre

2. Réponses

OK (200)

3. Exemple

```
{
  "status_code": 200,
  "status_message": "Succès",
  "data": [
    {
      "id_medecin": 23,
      "civilite": "Mme",
      "nom": "Deschamps",
      "prenom": "Isabelle"
    },
    {
      "id_medecin": 21,
      "civilite": "Mme",
      "nom": "Lafontaine",
      "prenom": "Élise"
    },
    ... (autres médecins)
  ]
}
```

b. Obtenir un seul médecin

GET http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/medecins/1

Permet d'obtenir un médecin à partir de son identifiant.

1. Requêtes

Paramètre d'URL sur l'identifiant du médecin souhaité.

2. Réponses

OK (200)

4. Exemple

```
{
  "status_code": 200,
  "status_message": "Succès",
  "data": [
    {
      "id_medecin": 1,
      "civilite": "M",
      "nom": "Cheng",
      "prenom": "Li"
    }
  ]
}
```

c. Créer un médecin

POST http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/medecins

Permet de créer un médecin et de le stocker dans la base de données.

1. Requêtes

Paramètres de corps de la requête (se référer au tableau page 2) :

civilite	(obligatoire)
nom	(obligatoire)
prenom	(obligatoire)

2. Réponses

OK (201)

3. Exemple

```
{
  "status_code": 201,
  "status_message": "Succès",
  "data": [
    {
      "id_medecin": 13,
      "civilite": "Mme",
      "nom": "Beaujour",
      "prenom": "Jade"
    }
  ]
}
```

d. Mettre à jour partiellement un médecin

PATCH http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/medecins/13

Permet de mettre à jour les caractéristiques d'un médecin. Cette mise à jour peut se faire sur une partie seulement des caractéristiques du médecin.

1. Requêtes

Paramètre d'URL obligatoire sur l'identifiant de la ressource à mettre à jour

Paramètres de corps de la requête (se référer au tableau page 2) :

civilite (optionnel)

nom (optionnel)

prenom (optionnel)

2. Réponses

OK (200)

3. Exemple

```
{
  "status_code": 200,
  "status_message": "Données id : 13 modifiées avec succès",
  "data": [
    {
      "id_medecin": 13,
      "civilite": "Mme",
      "nom": "Beaujour",
      "prenom": "Denisa"
    }
  ]
}
```


e. Mettre à jour totalement un médecin

PUT http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/medecins/13

Permet de mettre à jour toutes les caractéristiques d'un médecin. Cette mise à jour doit se faire sur la totalité des caractéristiques.

1. Requêtes

Paramètre d'URL obligatoire sur l'identifiant de la ressource à mettre à jour

Paramètres de corps de la requête (se référer au tableau page 2) :

civilite	<i>(obligatoire)</i>
nom	<i>(obligatoire)</i>
prenom	<i>(obligatoire)</i>

2. Réponses

OK (200)

3. Exemple

```
{
  "status_code": 200,
  "status_message": "Données id : 13 modifiées avec succès",
  "data": [
    {
      "id_medecin": 13,
      "civilite": "Mme",
      "nom": "Ripiceanu",
      "prenom": "Denisa"
    }
  ]
}
```

f. Supprimer un médecin

DEL http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/medecins/13

Permet de supprimer un médecin identifié par son id.

1. Requêtes

Paramètre d'URL obligatoire sur l'identifiant de la ressource à supprimer

2. Réponses

OK (200)

3. Exemple

```
{
  "status_code": 200,
  "status_message": "Le médecin a été supprimé avec succès.",
  "data": null
}
```

B. Usagers

a. Obtenir tous les usagers

GET http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/usagers

Permet d'obtenir tous les usagers en ordre alphabétique du nom.

1. Requêtes

Aucun paramètre

2. Réponses

OK (200)

3. Exemple

```
{
  "status_code": 200,
  "status_message": "Succès",
  "data": [
    {
      "id_usager": 17,
      "civilite": "M",
      "nom": "Beauséjour",
      "prenom": "Richard",
      "sexe": "H",
      "adresse": "1 Impasse André Marlin",
      "ville": "Toulouse",
      "code_postal": "31200",
      "date_naissance": "2004-05-16",
      "lieu_naissance": "Guadeloupe",
      "num_secu_sociale": "1361322535334",
      "id_medecin": 4
    },
    {
      "id_usager": 12,
      "civilite": "Mme",
      "nom": "Bertrand",
      "prenom": "Emma",
      "sexe": "F",
      "adresse": "890 Avenue de l'Été",
      "ville": "Lille",
      "code_postal": "59000",
      "date_naissance": "1988-11-30",
      "lieu_naissance": "Lille",
      "num_secu_sociale": "123490173543210",
      "id_medecin": 3
    },
    ... (autres usagers)
  ]
}
```

b. Obtenir un seul usager

GET http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/usagers/2

Permet d'obtenir un usager à partir de son identifiant.

1. Requêtes

Paramètre d'URL sur l'identifiant du patient souhaité.

2. Réponses

OK (200)

3. Exemple

```
{
  "status_code": 200,
  "status_message": "Succès",
  "data": [
    {
      "id_usager": 2,
      "civilite": "M",
      "nom": "Lefevre",
      "prenom": "Julien",
      "sexe": "H",
      "adresse": "789 Rue du Commerce",
      "ville": "Lyon",
      "code_postal": "69001",
      "date_naissance": "1990-12-10",
      "lieu_naissance": "Paris",
      "num_secu_sociale": "112233445566778",
      "id_medecin": 3
    }
  ]
}
```

c. Créer un usager

POST http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/usagers

Permet de créer un usager et de le stocker dans la base de données.

1. Requêtes

Paramètres de corps de la requête (se référer au tableau page 2) :

civilite	(obligatoire)
nom	(obligatoire)
prenom	(obligatoire)
sexe	(obligatoire)
adresse	(obligatoire)
ville	(obligatoire)
code_postal	(obligatoire)
date_naissance	(obligatoire)
lieu_naissance	(obligatoire)
num_secu_sociale	(obligatoire)
id_medecin	(obligatoire)

2. Réponses

OK (200)

3. Exemple

```
{
  "status_code": 200,
  "status_message": "Succès",
  "data": [
    {
      "id_usager": 21,
      "civilite": "M",
      "nom": "Planques",
      "prenom": "Maxime",
      "sexe": "H",
      "adresse": "1, rue Jeanne d'Arc",
      "ville": "Toulouse",
      "code_postal": "31000",
      "date_naissance": "2002-03-04",
      "lieu_naissance": "Nantes",
      "num_secu_sociale": "112233445446778",
      "id_medecin": 1
    }
  ]
}
```

d. Mettre à jour partiellement un usager

PATCH http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/usagers/21

Permet de mettre à jour les caractéristiques d'un usager. Cette mise à jour peut se faire sur une partie seulement des caractéristiques de l'utilisateur.

1. Requêtes

Paramètre d'URL obligatoire sur l'identifiant de la ressource à mettre à jour

Paramètres de corps de la requête (se référer au tableau page 2) :

civilite	(optionnel)
nom	(optionnel)
prenom	(optionnel)
sexe	(optionnel)
adresse	(optionnel)
ville	(optionnel)
code_postal	(optionnel)
date_naissance	(optionnel)
lieu_naissance	(optionnel)
num_secu_sociale	(optionnel)
id_medecin	(optionnel)

2. Réponses

OK (200)

3. Exemple

```
{
  "status_code": 200,
  "status_message": "Succès",
  "data": [
    {
      "id_usager": 21,
      "civilite": "M",
      "nom": "Planques",
      "prenom": "Maxime",
      "sexe": "H",
      "adresse": "1, rue Jeanne d'Arc",
      "ville": "Paris",
      "code_postal": "75000",
      "date_naissance": "2002-03-04",
      "lieu_naissance": "Nantes",
      "num_secu_sociale": "112233445446778",
      "id_medecin": 1
    }
  ]
}
```

e. Mettre à jour totalement un usager

PUT http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/usagers/21

Permet de mettre à jour toutes les caractéristiques d'un usager. Cette mise à jour doit se faire sur la totalité des caractéristiques.

1. Requêtes

Paramètre d'URL obligatoire sur l'identifiant de la ressource à mettre à jour

Paramètres de corps de la requête (se référer au tableau page 2) :

civilite	(obligatoire)
nom	(obligatoire)
prenom	(obligatoire)
sexe	(obligatoire)
adresse	(obligatoire)
ville	(obligatoire)
code_postal	(obligatoire)
date_naissance	(obligatoire)
lieu_naissance	(obligatoire)
num_secu_sociale	(obligatoire)
id_medecin	(obligatoire)

2. Réponses

OK (200)

3. Exemple

```
{
  "status_code": 200,
  "status_message": "Succès",
  "data": [
    {
      "id_usager": 21,
      "civilite": "M",
      "nom": "Planques",
      "prenom": "Maxime",
      "sexe": "H",
      "adresse": "1, rue Jeanne d'Arc",
      "ville": "Paris",
      "code_postal": "75000",
      "date_naissance": "2002-03-04",
      "lieu_naissance": "Nantes",
      "num_secu_sociale": "112233445446778",
      "id_medecin": 2
    }
  ]
}
```

f. Supprimer un usager

DEL http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/usagers/21

Permet de supprimer un usager identifié par son id.

1. Requêtes

Paramètre d'URL obligatoire sur l'identifiant de la ressource à supprimer

2. Réponses

OK (200)

3. Exemple

```
{  
  "status_code": 200,  
  "status_message": "L'utilisateur a été supprimé avec succès",  
  "data": null  
}
```

C. Rendez-vous

a. Obtenir tous les rendez-vous

GET http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/consultations

Permet d'obtenir tous les rendez-vous planifiés.

1. Requêtes

Aucun paramètre nécessaire

2. Réponses

OK (200)

3. Exemple

```
{
  "status_code": 200,
  "status_message": "Succès",
  "data": [
    {
      "id_rendezvous": 1,
      "date_consultation": "2024-05-14",
      "heure_consultation": "15:45:00",
      "duree_consultation": 60,
      "id_usager": 9,
      "id_medecin": 8
    },
    {
      "id_rendezvous": 2,
      "date_consultation": "2024-04-08",
      "heure_consultation": "14:30:00",
      "duree_consultation": 60,
      "id_usager": 11,
      "id_medecin": 2
    },
    {
      "id_rendezvous": 3,
      "date_consultation": "2024-04-15",
      "heure_consultation": "11:15:00",
      "duree_consultation": 30,
      "id_usager": 12,
      "id_medecin": 1
    },
    (autres rendez-vous) ...
  ]
}
```


b. Obtenir un seul rendez-vous

GET http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/consultations/6

Permet d'obtenir une consultation à partir de son identifiant.

1. Requêtes

Paramètre d'URL sur l'identifiant du rendez-vous souhaité.

2. Réponses

OK (200)

3. Exemple

```
{
  "status_code": 200,
  "status_message": "Succès",
  "data": [
    {
      "id_rendezvous": 6,
      "date_consultation": "2024-03-20",
      "heure_consultation": "15:30:00",
      "duree_consultation": 60,
      "id_usager": 2,
      "id_medecin": 1
    }
  ]
}
```

c. Créer un rendez-vous

POST http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/consultations

Permet de créer un rendez-vous et de le stocker dans la base de données.

1. Requêtes

Paramètres de corps de la requête (se référer au tableau page 3) :

<code>date_consultation</code>	<i>(obligatoire)</i>
<code>heure_consultation</code>	<i>(obligatoire)</i>
<code>duree_consultation</code>	<i>(obligatoire)</i>
<code>id_usager</code>	<i>(obligatoire)</i>
<code>id_medecin</code>	<i>(obligatoire)</i>

2. Réponses

OK (200)

3. Exemple

```
{
  "status_code": 200,
  "status_message": "Succès",
  "data": [
    {
      "id_rendezvous": 17,
      "date_consultation": "2024-12-24",
      "heure_consultation": "10:00:00",
      "duree_consultation": 30,
      "id_usager": 2,
      "id_medecin": 8
    }
  ]
}
```

d. Mettre à jour partiellement un rendez-vous

PATCH http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/consultations/6

Permet de mettre à jour les caractéristiques d'un rendez-vous. Cette mise à jour peut se faire sur une partie seulement des caractéristiques.

1. Requêtes

Paramètre d'URL obligatoire sur l'identifiant de la ressource à mettre à jour

Paramètres de corps de la requête (se référer au tableau page 3) :

<code>date_consultation</code>	<i>(optionnel)</i>
<code>heure_consultation</code>	<i>(optionnel)</i>
<code>duree_consultation</code>	<i>(optionnel)</i>
<code>id_usager</code>	<i>(optionnel)</i>
<code>id_medecin</code>	<i>(optionnel)</i>

2. Réponses

OK (200)

3. Exemple

```
{
  "status_code": 200,
  "status_message": "Succès",
  "data": [
    {
      "id_rendezvous": 6,
      "date_consultation": "2024-12-12",
      "heure_consultation": "12:30:00",
      "duree_consultation": 45,
      "id_usager": 3,
      "id_medecin": 8
    }
  ]
}
```

e. Mettre à jour totalement un rendez-vous

PUT http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/consultations/6

Permet de mettre à jour toutes les caractéristiques d'un rendez-vous. Cette mise à jour doit se faire sur la totalité des caractéristiques.

1. Requêtes

Paramètre d'URL obligatoire sur l'identifiant de la ressource à mettre à jour

Paramètres de corps de la requête (se référer au tableau page 3) :

<code>date_consultation</code>	<i>(obligatoire)</i>
<code>heure_consultation</code>	<i>(obligatoire)</i>
<code>duree_consultation</code>	<i>(obligatoire)</i>
<code>id_usager</code>	<i>(obligatoire)</i>
<code>id_medecin</code>	<i>(obligatoire)</i>

2. Réponses

OK (200)

3. Exemple

```
{
  "status_code": 200,
  "status_message": "Succès",
  "data": [
    {
      "id_rendezvous": 6,
      "date_consultation": "2024-11-12",
      "heure_consultation": "12:30:00",
      "duree_consultation": 45,
      "id_usager": 3,
      "id_medecin": 4
    }
  ]
}
```

f. Supprimer un rendez-vous

DEL http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/consultation/7

Permet de supprimer un rendez-vous identifié par son id.

1. Requêtes

Paramètre d'URL obligatoire sur l'identifiant de la ressource à supprimer

2. Réponses

OK (200)

3. Exemple

```
{
  "status_code": 200,
  "status_message": "La consultation a été supprimée avec succès",
  "data": null
}
```

D. Statistiques

a. Obtenir les statistiques pour les usagers

GET http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/statistiques/usagers

Permet d'obtenir les statistiques pour tous les usagers.

1. Requêtes

Aucun paramètre

2. Réponses

OK (200)

3. Exemple

```
{
  "status_code": 200,
  "status_message": "Succès",
  "data": [
    [
      "Cheng",
      "Li",
      0.5
    ],
    (autres statistiques) ...
  ]
}
```

b. Obtenir les statistiques pour les médecins

GET http://beaujouripiceanu.alwaysdata.net/api_cabinet_medical/statistiques/medecins

Permet d'obtenir les statistiques pour tous les médecins.

1. Requêtes

Aucun paramètre

2. Réponses

OK (200)

3. Exemple

```
{
  "status_code": 200,
  "status_message": "Succès",
  "data": {
    "Entre 25 et 50 ans": [
      15,
      0
    ],
    "Moins de 25 ans": [
      3,
      0
    ]
  },
  (autres statistiques) ...
}
```

E. Authentication

a. S'authentifier

POST http://api-authentication.alwaysdata.net/api_authentication/api_authentication.php

Permet d'obtenir un jeton d'authentification

1. Requêtes

Paramètres de corps de la requête (se référer au tableau page 3) :

user (obligatoire)

mdp (obligatoire)

2. Réponses

OK (200)

3. Exemple

```
{
  "token":
  "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjozLCJyb2xlIjoic2VjcmV0YWlyZSIsImV4cCI6MTcxMjUwMDA3N30.3J7R2YRi6UfrEwQV7ZgwZe4iuIdM6Vb00KdVrgpOgzU"
}
```

Détail : Automatiquement, le paramètre Authorization est mis à jour avec le jeton stocké dans une variable appelée "token". Cette opération est effectuée dans la section **Tests** du client Postman :
`pm.collectionVariables.set("token", pm.response.json().token);`

Ce script s'exécute automatiquement dès qu'une requête est envoyée et que les données sont reçues. Il récupère le jeton contenu dans le champ "token" du JSON de la réponse, puis le place dans la variable "token" précédemment créée. Ensuite, le paramètre Authorization du header est initialisé comme suit:

```
pm.request.headers.add({
  key: 'Authorization',
  value: 'Bearer ' + pm.collectionVariables.get("token")
});
```

b. Valider un jeton

GET http://api-authentication.alwaysdata.net/api_authentication/api_authentication.php

Permet de vérifier la validité de son jeton.

1. Requêtes

Paramètre Authorization avec le JWT Token nécessaire à la connexion

2. Réponses

OK (200)

3. Exemple

```
{
  "message": "Le jeton est valide",
  "statut": 200
}
```