

Documentation du projet de R4.01 : Gestion d'équipe de sport

API de gestion d'équipe de volley-ball avec une sécurisation grâce à JWT

Table des matières

I-	Présentation générale.....	3
II-	Présentation des ressources.....	3
	<i>Matches</i>	<i>3</i>
	<i>Joueurs.....</i>	<i>3</i>
	<i>Feuille de match.....</i>	<i>5</i>
III-	Format des réponses	5
	<i>Tableau des code de retour.....</i>	<i>5</i>
IV-	API d'authentification	6
	<i>Connexion</i>	<i>6</i>
	<i>Vérification du Token</i>	<i>6</i>
V-	API de gestion d'équipe de volley-ball.....	7
	<i>Endpoint: Gestion des Feuilles de Match.....</i>	<i>7</i>
	Récupération des joueurs d'une feuille de match	7
	Création d'une feuille de match	8
	Modification d'une feuille de match	9
	Suppression d'une feuille de match	9
	<i>Endpoint : Gestion des Joueurs.....</i>	<i>10</i>
	Récupération des joueurs.....	10
	Création d'un joueur.....	10
	Modification d'un Joueur	11
	Suppression d'un Joueur	12
	<i>Endpoint : Gestion des Matches</i>	<i>13</i>
	Récupération des matchs	13
	Création d'un match	13
	Modification d'un Match	14
	Suppression d'un Match	15
	<i>Endpoint: Statistiques.....</i>	<i>16</i>
	Récupération des Statistiques	16

I- Présentation générale

Ce projet mêle 2 API différentes et une interface web utilisateur, une première API (nommé API d'authentification) qui permet d'ajouter une couche de sécurité au reste du projet. Elle permet de gérer les login et mot de passe du projet et crée des JWT pour accéder à l'autre API. Cette seconde API est l'API de gestion d'équipe de sport, dans notre cas de volley-ball. Elle permet de gérer toute une équipe. Vous pouvez gérer les joueurs, matchs et feuilles de matchs (participation des joueurs à un match). Vous pouvez également obtenir des statistiques sur votre équipe ! Pour des raisons de rendu, il y a déjà un jeu de données pour la gestion des équipes. Vous pouvez donc exécuter des requêtes directement sur le serveur ou bien via notre client.

II- Présentation des ressources

Voici les divers champs disponibles dans les données de notre API, attention ! Les types de données affichés sont ceux enregistrés dans la base de données. Vous pouvez vous référer aux méthodes GET pour savoir les types renvoyés :

Matches

Champs	Type	Rôle
id	Entier	Identifiant unique pour chaque match
date_heure	Date heure	La date et l'heure du match
equipeadv	Chaîne	Le nom de l'équipe adverse
domicile	Booléen	Indique si le match s'est passé à domicile (vrai) ou pas (faux)
score	Chaîne	Le score final du match, notre score de chaque set étant toujours sur la gauche. Le score est sous le format : « notre score-score adverse, notre score-score adverse, ... » et doit respecter les règles officielles du volley.
gagne	Booléen	Indique si le match a été gagné (vrai) ou pas (faux)

Joueurs

Champs	Type	Rôle
id	Entier	Identifiant unique pour chaque joueur
nom	Chaîne	Le nom du joueur
prénom	Chaîne	Le prénom du joueur
taille	Décimal	La taille du joueur
poids	Entier	Le poids du joueur
commentaire	Chaîne	Commentaire de l'entraîneur sur le joueur

Feuille de match

Champs	Type	Rôle
idMatch	Entier	Identifiant unique pour chaque match
idJoueur	Entier	Identifiant unique pour chaque joueur
role_titulaire	Booléen	Identifie si le rôle est titulaire (vrai) ou pas (faux)
poste	Chaîne	Définit le rôle du joueur pour le match, ils doivent absolument être dans cette liste : « Attaquant, Passeur, Centre, Libero, Remplaçant »
note	Entier	Note pour la performance du joueur lors du match, compris entre 0 et 5

III- Format des réponses

Toutes les réponses sont renvoyées au format JSON. Le format de la réponse comportera toujours ces champs que ce soit un succès ou une erreur :

status_code : code de retour (cf tableau ci dessous)

status_message : Message personnalisé

data : null si pas de données retournées ou un tableau des données retournées par votre requête

Tableau des codes de retour

Code de statut	Signification	Description
200	OK	La requête est réussie
201	Created	Une nouvelle ressource a été créée avec succès
400	Bad request	Syntaxe non conforme/non reconnue
401	Unauthorized	Une authentification est nécessaire à l'aide d'un JWT fourni par l'API d'authentification
403	Forbidden	La requête est comprise mais ne peut pas être exécuté à cause de restrictions
404	Not found	Ressource demandée non trouvée
405	Method not allowed	Méthode non implémentée mais reconnue
418	I'm a teapot	Le serveur refuse de brasser du café avec une théière <i>Ceci est évidemment une blague, nous avons trouvé le code en lisant la documentation</i>
500	Internal server Error	Erreur d'exécution de requête côté serveur

IV- API d'authentification

Dépôt GitHub : <https://github.com/ClementReverbel/API-Authentification>

Lien vers l'API : <https://authapi.alwaysdata.net>

Connexion

- **Méthode:** POST
- **URL:** /authapi.php
- **Description:** Permet aux utilisateurs de se connecter et de recevoir un token JWT.
- **Paramètres:**
 - login (string, requis) : Identifiant de l'utilisateur.
 - password (string, requis) : Mot de passe de l'utilisateur.
- **Réponses:**
 - 200 OK : Connexion réussie, retourne un token JWT.
 - 400 Bad Request : Login ou mot de passe manquant.
 - 401 Unauthorized : Mot de passe incorrect.
- **Retour d'information (data) :**
 - Si valide : Le JWT valide et disponible de 30 minutes, exemple :

```
{
  "status_code": 200,
  "status_message": "Connexion établie",
  "data": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFkbWluIiwiaXNjaXhwIjozNzQzMDcwNzYxYQ.0sYwki6YZh5kGR8hcKF6t0HgX0S16CMqmm6tPgFKVt4"
}
```

Vérification du Token

- **Méthode:** GET
- **URL:** /authapi.php
- **Description:** Vérifie la validité du token JWT.
- **Réponses:**
 - 200 OK : Token valide.
 - 418 I'm a teapot : Token expiré ou invalide.

(normalement 401 – cf Tableau des retours code, blague)

- **Exemple :**

```
{
  "status_code": 200,
  "status_message": "Jeton valide",
  "data": null
}
```

V- API de gestion d'équipe de volley-ball

Dépôt GitHub : <https://github.com/ClementReverbel/API-VolleyTrack>

Lien vers l'API : <https://volleyapi.alwaysdata.net/ressources>

Endpoint: Gestion des Feuilles de Match

Récupération des joueurs d'une feuille de match

- **Méthode:** GET
- **URL:** /feuillematch_endpoint.php?[id=...]
- **Paramètres:**
 - id (int, requis) : Identifiant du match.
- **Réponses:**
 - 200 OK : Liste des joueurs du match.
 - 400 Bad Request : ID manquant.
 - 401 Unauthorized : Le JWT est manquant, expiré ou invalide.
 - 403 Forbidden : L'ID doit être numérique.
 - 404 Not Found : Match inexistant ou sans feuille de match.

- **Retour d'information (data) :**

- Si valide : La liste des joueurs dans la feuille du match choisi, exemple :

```
{
  "status_code": 200,
  "status_message": "Joueurs du match sélectionné récupérés avec succès",
  "data": [
    {
      "idJoueur": 1,
      "Numéro_de_licence": "VOL1234567",
      "NomComple": "Dubois Antoine",
      "Taille": 1.98,
      "Poids": 85,
      "Moyenne_note": "4.0",
      "Commentaire": "Excellent au bloc, très bon timing.",
      "Poste": "Attaquant",
      "Role_titulaire": 1
    }
  ]
}
```

Création d'une feuille de match

- **Méthode:** POST
- **URL:** /feuillematch_endpoint.php
- **Paramètres:**
 - id (int, requis) : Identifiant du match.
 - liste_joueurs (array, requis) : Liste des id des joueurs pour la feuille de match. Ils doivent être minimum 6, au maximum 12 et aucun doublon.
 - liste_roles (array, requis) : Liste des rôles des joueurs pour la feuille des matchs correspondant à chaque joueurs. Chaque rôle doit être compris dans cette liste :
« Attaquant, Passeur, Centre, Libero, Remplaçant »

Exemple :

```
1 {
2   "id": "31",
3   "liste_joueurs": ["1", "2", "3", "4", "5", "6"],
4   "liste_roles" : ["Attaquant", "Attaquant", "Centre", "Libero", "Passeur", "Centre"]
5 }
```

- **Réponses:**
 - 200 OK : Feuille créée.
 - 400 Bad Request : Données manquantes qui seront stipulées.

- 500 Internal Server Error : Erreur serveur.

Modification d'une feuille de match

- **Méthode:** PUT
- **URL:** /feuillematch_endpoint.php
- **Paramètres:**
 - id (int, requis) : Identifiant du match.
 - liste_joueurs (array, requis) : Liste des id des joueurs pour la feuille de match. Ils doivent être minimum 6, au maximum 12 et aucun doublon.
 - liste_roles (array, requis) : : Liste des rôles des joueurs pour la feuille des matchs correspondant à chaque joueurs. Chaque rôle doit être compris dans cette liste :

« Attaquant, Passeur, Centre, Libero, Remplaçant »
 - liste_notes (array, requis) : Notes des joueurs correspondant à chaque joueurs, compris entre 0 et 5.

Exemple :

```
1  {
2      "id": "31",
3      "liste_joueurs": ["7", "2", "3", "4", "5", "6"],
4      "liste_roles" : ["Attaquant", "Remplaçant", "Remplaçant", "Libero", "Passeur", "Centre"],
5      "liste_notes" : [10, 4, 2, 5, 6, 5]
6  }
7
```

- **Réponses:**
 - 200 OK : Feuille modifiée.
 - 400 Bad Request : Données incorrectes qui seront stipulées.
 - 500 Internal Server Error : Erreur serveur.

Suppression d'une feuille de match

- **Méthode:** DELETE
- **URL:** /feuillematch_endpoint.php
- **Réponses:**
 - 405 Method Not Allowed : Suppression non autorisée.

Endpoint : Gestion des Joueurs

Récupération des joueurs

- **Méthode** : GET
- **URL** : /joueurs_endpoint.php ?[numLic=...]/[actif=0/1]
- **Paramètres** :
 - numLic (string, optionnel) : Numéro de licence du joueur pour n'obtenir les données que ce joueur.
 - actif (booléen, optionnel) : Récupère uniquement les joueurs actifs si vrai.
 - Si aucun paramètre n'est stipulé alors on reçoit tous les joueurs
- **Réponses**:
 - 200 OK : Liste des joueurs récupérée.
 - 400 Bad Request : Numéro de licence incorrect (il doit faire 10 caractères et commencer par VOL).
 - 404 Not Found : Joueur non trouvé.
- **Retour d'information (data)** :
 - Si valide : La liste des joueurs avec l'argument choisi, exemple

Exemple :

```
{
  "status_code": 200,
  "status_message": "Liste des joueurs récupérée avec succès",
  "data": [
    {
      "idJoueur": 1,
      "Numéro_de_licence": "VOL1234567",
      "NomComplet": "Dubois Antoine",
      "Taille": 1.98,
      "Poids": 85,
      "Moyenne_note": "4.0",
      "Commentaire": "Excellent au bloc, très bon timing."
    },
    {

```

Création d'un joueur

- **Méthode**: POST
- **URL**: /joueurs_endpoint.php

- **Paramètres:**

- numLic (string, requis) : Numéro de licence.
- nom (string, requis) : Nom du joueur.
- prenom (string, requis) : Prénom du joueur.
- date_de_naissance (date, requis) : Date de naissance, dans le format :
« jour-mois-année. »
- taille (float, requis) : Taille en cm.
- poids (int, requis) : Poids en kg.
- commentaire (string, requis) : Commentaire.

Créer forcément le joueur en actif.

Exemple :

```
1  {
2      "numLic": "VOL7895632",
3      "nom" : "Arthero",
4      "prenom": "Anthony",
5      "date_de_naissance": "10-08-2003",
6      "taille": 1.60,
7      "poids":60,
8      "commentaire":"bonne réception"
9  }
```

- **Réponses:**

- 201 Created : Joueur créé.
- 400 Bad Request : Données invalides.
- 409 Conflict : Joueur déjà existant.

Modification d'un Joueur

- **Méthode:** PUT

- **URL:** /joueurs_endpoint.php

- **Paramètres:**

- numLic (string, requis) : Numéro de licence du joueur à modifier.
- nom (string, requis) : Nom du joueur.

- prenom (string, requis) : Prénom du joueur.
- date_de_naissance (date, requis) : Date de naissance.
- taille (float, requis) : Taille en cm.
- poids (int, requis) : Poids en kg.
- commentaire (string, requis) : Commentaire.
- statut (string, requis) : Statut (Actif, Blessé, Suspendu, Absent).

Exemple :

```
1  {
2      "numLic": "VOL7895632",
3      "nom" : "Arthero",
4      "prenom": "Anthony",
5      "date_de_naissance": "10-08-2003",
6      "taille": 1.60,
7      "poids":60,
8      "commentaire":"bonne réception",
9      "statut":"Blessé"
10 }
```

- **Réponses:**
 - 200 OK : Joueur modifié.
 - 400 Bad Request : Données invalides.
 - 404 Not Found : Joueur introuvable.

Suppression d'un Joueur

- **Méthode:** DELETE
- **URL:** /joueurs_endpoint.php
- **Paramètres:**
 - numLic (string, requis) : Numéro de licence (à préciser dans le body).
- **Réponses:**
 - 200 OK : Joueur supprimé.
 - 400 Bad Request : Données invalides.
 - 403 Forbidden : Impossible de supprimer un joueur ayant participé à un match.
 - 404 Not Found : Joueur non trouvé.

Endpoint : Gestion des Matches

Récupération des matchs

- **Méthode** : GET
- **URL** : /matches_endpoint.php ?[id=...]
- **Paramètres** :
 - id (int, optionnel) : Numéro de licence du joueur pour n'obtenir les données que ce joueur..
 - Si aucun paramètre n'est stipulé alors on reçoit tous les matchs
- **Réponses**:
 - 200 OK : Liste des joueurs récupérée.
 - 404 Not Found : Match non trouvé.
- **Retour d'information (data)** :
 - Si valide : La liste des matchs choisis, exemple :

Exemple :

```
{
  "status_code": 200,
  "status_message": "Matches récupérés avec succès",
  "data": [
    {
      "id": 1,
      "date_heure": "15-01-2024 18:30",
      "equipeadv": "Tigers Volley",
      "domicile": "OUI",
      "score": "25-20, 22-25, 25-18, 23-25, 15-12",
      "gagne": "GAGNÉ"
    }
  ]
}
```

Création d'un match

- **Méthode**: POST
- **URL**: /matches_endpoint.php
- **Paramètres**:
 - date (string, requis) : date du match au format :
« jour-mois-année »
 - heure (string, requis) : heure du match au format :
« heure:minute »

- equipeadv (string, requis) : Nom de l'équipe adverse pour ce match.
- domicile (boolean, requis) : 1 pour dire que le match a été joué à domicile et 0 pour dire l'inverse

Créer un match sans score car pas encore joué

Exemple :

```
1  {
2      "date": "26-04-2025",
3      "heure" : "20:00",
4      "equipeadv": "Les méchants",
5      "domicile": 1
6  }
```

- **Réponses:**
 - 201 Created : Joueur créé.
 - 400 Bad Request : Données invalides.

Modification d'un Match

- **Méthode:** PUT
 - **URL:** /matches_endpoint.php
 - **Paramètres:**
 - id (int, requis) : l'id du match que vous souhaitez modifier.
 - date (string, requis) : date du match au format :
« jour-mois-année »
 - heure (string, requis) : heure du match au format :
« heure:minute »
 - equipeadv (string, requis) : Nom de l'équipe adverse pour ce match.
 - domicile (boolean, requis) : 1 pour dire que le match a été joué à domicile et 0 pour dire l'inverse
 - score (string, requis) : le score du match, les points sur la gauche sont toujours de notre équipe, il faut respecter ce format :

« **notre score-score adverse, notre score-score adverse, ...** »
- De plus, il faut respecter les règles officielles du volley-ball :

- Les sets sont en 25 points gagnants dont 2 points d'écart minimum.
- Cela se joue en 3 sets gagnants et au maximum de 5 sets.
- Le dernier set se joue en 15 points gagnants dont 2 points d'écart.

Exemple :

```
1  {
2      "id" : 32,
3      "date": "26-04-2025",
4      "heure" : "20:00",
5      "equipeadv": "Les méchants",
6      "domicile": 1,
7      "score": "25-20,25-20,25-20"
8  }
9
```

- **Réponses:**
 - 200 OK : Match modifié.
 - 400 Bad Request : Données invalides.

Suppression d'un Match

- **Méthode:** DELETE
 - **URL:** /match_endpoint.php
 - **Paramètres:**
 - id (int, requis) : Identifiant du match à supprimer (à préciser dans le body). Le match ne doit pas avoir sa date dépassée (donc joué) et une feuille de match pour pouvoir être supprimé
 - **Réponses:**
 - 200 OK : Match supprimé.
 - 400 Bad Request : Données invalides.
 - 403 Forbidden : Impossible de supprimer un match déjà joué ou prévu.
 - 404 Not Found : Match non trouvé.
-

Endpoint: Statistiques

Récupération des Statistiques

- **Méthode:** GET
- **URL:** /stats_endpoint.php ?[stats_joueur/stats_match]=[0/1]
- **Paramètres:**
 - stats_joueur (booléen, 1 des deux est requis) : Récupère les statistiques des joueurs.
 - stats_match (booléen, 1 des deux est requis) : Récupère les statistiques des matchs.
- **Réponses:**
 - 200 OK : Statistiques récupérées.
 - 400 Bad Request : Paramètre invalide.
 - 405 Method Not Allowed : Seul GET est autorisé.

```
{
  "status_code": 200,
  "status_message": "Statistiques du joueur",
  "data": [
    {
      "idJoueur": 1,
      "Nom": "Dubois",
      "Prenom": "Antoine",
      "Statut": "Actif",
      "Poste_preferé": "Attaquant",
      "Total_titulaire": 10,
      "Total_remplacant": 0,
      "Moyenne_note": "4.0".
    }
  ]
}
```