

Documentation technique – API de gestion d'une équipe sportive

Table des matières

I. Présentation générale.....	2
II. Présentation des ressources.....	2
III. Url.....	2
IV. Format des réponses.....	3
V. Fonctionnalités proposées par nos API.....	4
A. Joueurs.....	4
B. Matches.....	8
C. Feuilles de match.....	12
D. Statistiques.....	16
E. Authentification.....	17

I. Présentation générale

Cette API permet à une équipe sportive de gérer les joueurs, les matchs, les feuilles de match ainsi que les statistiques associées. L'ensemble du projet est sécurisé par un système d'authentification JWT. Bien que certaines URL ne respectent pas intégralement le format REST, les opérations principales sont toutes couvertes via les méthodes HTTP classiques (GET, POST, PUT, DELETE).

II. Présentation des ressources

Les ressources exposées par l'API sont les suivantes :

- **Joueurs** : informations personnelles, numéro de licence, taille, poids, etc.
- **Matchs** : date, heure, lieu, équipe adverse, résultat.
- **Feuilles de match** : joueurs ayant participé, leurs postes, notes, titulaire/remplaçant etc.
- **Statistiques** : moyenne des notes, performances par joueur, etc.
- **Authentification** : Connexion via login/mot de passe et génération d'un token JWT.

Chaque ressource est liée à une table SQL dans `sport_db.sql` et `auth_db.sql`.

III. URL de l'interface client

L'interface graphique développée en HTML/JS est disponible ici :

<https://deploiementapiissam.alwaysdata.net/affichage/index.html>

Elle permet d'envoyer des requêtes GET, POST, PUT, DELETE aux API selon les actions de l'utilisateur.

Nous avons décidé de prendre le même type d'interface client que pour chuckapi.

Pour la connexion, les identifiants sont :

Login : coach1

Mot de passe : admin

IV. Format des réponses

Les réponses sont au format JSON.

Structure type :

```
data : [...]
```

Codes d'état HTTP utilisés :

- 200 : OK
- 201 : Created
- 400 : Bad Request
- 401 : Unauthorized
- 403 : Forbidden
- 404 : Not Found
- 500 : Internal Server Error

V. Fonctionnalités proposées par les API

A. Joueurs

Liste, création, modification et suppression des joueurs via l'endpoint
/back_end/joueurs.php

a/ Obtenir tout les joueurs

GET

Permet d'obtenir les joueurs

url : https://deploiementapiissam.alwaysdata.net/back_end/joueurs.php

1. Requête

Aucun paramètre

2. Réponse

OK (200)

3. Exemple

```
[
  {
    "status_code": 200,
    "status_message": "Succès",
    {
      "Numero_Licence": "LIC11122",
      "Nom": "Kanté",
      "Prenom": "N'Golo",
      "Date_De_Naissance": "1991-03-29",
      "Taille": "1",
      "Poids": "70.0",
      "Commentaires": "un commentaire",
      "Poste_Joueur": "Ailier",
      "Id_Statut": "2"
    }
  },
  ... (autres joueurs)
]
```

b/ Créer un joueur

POST

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

url : https://deploiementapiissam.alwaysdata.net/back_end/joueurs.php

1. Requête

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

Numero_Licence (obligatoire)

Nom (obligatoire)

Prenom (obligatoire)

Id_statut (obligatoire , [1 -> Actif , 2-> blessé , 3 -> Suspendu])

2. Réponse

OK (201)

3. Exemple

```
4. {  
5.   "Numero_Licence": "LIC77777",  
6.   "Nom": "Gobert",  
7.   "Prenom": "Rudy",  
8.   "Date_De_Naissance": "1992-06-26",  
9.   "Taille": "2.16",  
10.  "Poids": "117.0",  
11.  "Commentaires": "Pivot principal",  
12.  "Poste_Joueur": "Pivot",  
13.  "Id_Statut": 1  
14. }  
15.
```

C/ Modifier un joueur

PUT

Permet de modifier un joueur de la base de données. Cette mise à jour doit se faire sur la totalité des caractéristiques

url : https://deploiementapiissam.alwaysdata.net/back_end/joueurs.php?Numero_Licence=LIC77777

1. Requêtes

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

Numero_Licence (obligatoire)

Nom (obligatoire)

Prenom (obligatoire)

Id_statut (obligatoire , [1 -> Actif , 2-> blessé , 3 -> Suspendu])

2. Réponse

```
3. {  
4.   "message": "Joueur modifié avec succès"  
5. }
```

3. Exemple

```
{  
  "Nom": "Gobert",  
  "Prenom": "Rudy",  
  "Date_De_Naissance": "1992-06-26",  
  "Taille": "2.15",  
  "Poids": "115.5",  
  "Commentaires": "Modifié en test",  
  "Poste_Joueur": "Pivot",  
  "Id_Statut": 2  
}
```

d/ Supprimer un joueur

DELETE

Permet de supprimer un joueur identifié par son id.

url : https://deploiementapiissam.alwaysdata.net/back_end/joueurs.php?Numero_Licence=LIC77777

1. Requêtes

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

2. Réponse

OK (200)

3. Exemple

```
{  
  "message": "Joueur supprimé avec succès"  
}
```

B. Matches

Gestion complète des matchs via /back_end/matches.php

a/ Obtenir tout les matchs

GET

Permet de recuperer tout les joueurs

url : https://deploiementapiissam.alwaysdata.net/back_end/matches.php

1. Requête

Aucun paramètre

2. Réponse

OK (200)

3. Exemple

```
4. [  
5.   {  
6.     "Id_Match": "1",  
7.     "Date_Heure_Match": "2024-03-01 15:00:00",  
8.     "Lieu": "Accor Arena",  
9.     "Adversaire": "Team USA",  
10.    "Score_Equipe": 78,  
11.    "Score_Adversaire": 72,  
12.    "Victoire": 1,  
13.    "Egalite": 0  
14.  },
```

(...autre match)

```
]
```


b/ Créer un Match

POST

Permet de créer un match et de le stocker dans la base de données

url : https://deploiementapiissam.alwaysdata.net/back_end/matches.php

1. Requête

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

Id_Match (obligatoire)
Date_Heure_Match (obligatoire)
Lieu (obligatoire)
Adversaire (obligatoire)

2. Réponses

OK (200)

3. Exemple

```
4. {  
5.   "Id_Match": 4,  
6.   "Date_Heure_Match": "2024-05-10 19:00:00",  
7.   "Lieu": "Toulouse Arena",  
8.   "Adversaire": "Italie U20",  
9.   "Score_Equipe": 81,  
10.  "Score_Adversaire": 76,  
11.  "Victoire": 1,  
12.  "Egalite": 0  
13. }  
14.
```

C/ Modifier un match

PUT

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

url : https://deploiementapiissam.alwaysdata.net/back_end/matches.php?Id_Match=4

1. Requête

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

Date_Heure_Match (obligatoire)

Lieu (obligatoire)

Adversaire (obligatoire)

Score_Equipe (obligatoire)

Score_Adversaire (obligatoire)

Victoire (obligatoire)

Egalite (obligatoire)

2. Réponse

OK (200)

3. Exemple

```
4. {  
5.   "Date_Heure_Match": "2024-05-10 20:00:00",  
6.   "Lieu": "Toulouse Arena",  
7.   "Adversaire": "Italie U20",  
8.   "Score_Equipe": 83,  
9.   "Score_Adversaire": 79,  
10.  "Victoire": 1,  
11.  "Egalite": 0  
12. }
```

d/ Supprimer un match

DELETE

Permet de supprimer un usager identifié par son id.

url : https://deploiementapiissam.alwaysdata.net/back_end/matches.php?Id_Match=4

1. Requête

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

2. Réponse

OK (200)

3. Exemple

```
4. {  
5.   "message": "Match supprimé avec succès"  
6. }
```

C. Feuilles de match

a/ Récupérer les feuilles de matchs

GET

Permet d'obtenir toutes les feuilles de matchs .

url : https://deploiementapiissam.alwaysdata.net/back_end/feuilles.php?Id_Match=3

1. Requête

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

2. Réponse

OK (200)

3. Exemple

Renvoie tout les joueur du match 3

```
[
  {
    "Numero_Licence": "LIC11122",
    "Nom": "Kanté",
    "Prenom": "N'Golo",
    "Statut_Participation": "Titulaire",
    "Poste_Match": "Ailier",
    "Note": "3.00"
  },
  {
    "Numero_Licence": "LIC12345",
    "Nom": "Mbappé",
    "Prenom": "Kylian",
    "Statut_Participation": "Titulaire",
    "Poste_Match": "Arrière",
    "Note": "5.00"
  },
  {
    "Numero_Licence": "LIC33344",
    "Nom": "Lloris",
    "Prenom": "Hugo",
    "Statut_Participation": "Titulaire",
    "Poste_Match": "Pivot",
    "Note": "4.00"
  },
  (... autre joueur)
```

b/ Ajouter une participation à un match

POST

Permet d'ajouter un joueur a un match qui existe déjà

url : https://deploiementapiissam.alwaysdata.net/back_end/feuilles.php

1. Requête

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

Numero_Licence (obligatoire)
Id_Match (obligatoire)
Statut_Participation (obligatoire)
Poste_Match (obligatoire)
Note (obligatoire)

2. Réponse

OK (200)

3. Exemple

```
4. {  
5.   "Numero_Licence": "LIC77777",  
6.   "Id_Match": 3,  
7.   "Statut_Participation": "Titulaire",  
8.   "Poste_Match": "Pivot",  
9.   "Note": "4"  
10. }  
11.
```

c/ Modifier participation

PUT

permet de modifier un match et de mettre à jour toutes les caractéristiques. Cette mise à jour doit se faire sur la totalité des caractéristiques.

url : https://deploiementapiissam.alwaysdata.net/back_end/feuilles.php?Numero_Licence=LIC12345&Id_Match=3

1. Requête

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) :

Numero_Licence (obligatoire)
Id_Match (obligatoire)
Statut_Participation (obligatoire)
Poste_Match (obligatoire)
Note (obligatoire)

2. Réponse

OK (200)

3. Exemple

```
4. {  
5.   "Statut_Participation": "Titulaire",  
6.   "Poste_Match": "Arrière",  
7.   "Note": "5"  
8. }  
9.
```

d/ Supprimer une participation

DELETE

Permet de supprimer une participation identifié par son id.

url : https://deploiementapiissam.alwaysdata.net/back_end/feuilles.php?Numero_Licence=LIC77777&Id_Match=3

1. Requête
Paramètre d'URL obligatoire sur l'identifiant de la ressource à supprimer
2. Réponse
OK (200)
3. Exemple

```
{  
  "message": "Participation supprimée"  
}
```

D. Statistiques

Accès aux statistiques calculées (moyenne de notes, etc.) via /back_end/stats.php

a/ Statistiques

GET

Permet de récupérer toutes statistiques pour les matchs et les joueurs

url : https://deploiementapiissam.alwaysdata.net/back_end/stats.php

1. Requête

Aucun paramètre

2. Réponse

OK (200)

3. Exemple

```
4. {  
5.   "moyenne_notes": [  
6.     {  
7.       "Numero_Licence": "LIC54321",  
8.       "Nom": "Griezmann",  
9.       "Prenom": "Antoine",  
10.      "Moyenne_Note": "4.50"  
11.    },  
12.  ]  
13. }
```

(...autres statistiques)

}

E. Authentification

Connexion sécurisée et validation du token JWT via /auth_api/login.php et /auth_api/profile.php

a/ S'authentifier

POST

Permet d'obtenir un jeton d'authentification

url : https://deploiementapiissam.alwaysdata.net/auth_api/login.php

1. Requête

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

2. Réponse

OK (200)

3. Exemple

```
4. {  
5.   "token":  
   "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJhdXRoX2FwaSI6ImNvYWN0Mzk2MDc3NCwiZmxwIjozNzQzOTY0Mzc0LCJ1c2VyIjp7Im1kIjozLCJsb2dpbiI6ImNvYWN0MSIsInJvbGUiOiJlbnRyYWluZXVzIn19.eysYzJjq5LPBWQwn1T0c2egRGx73XWLM16hC0_Mj1f4"  
6. }
```

Détail :

Pour chaque token généré, il faut modifier le Bearer token dans le dossier « Projet » pour que celui-ci soit récupérer par toutes les autres méthodes

b/ Valider un jeton

GET

Permet de vérifier la validité de son jeton. En récupéreront le profile du login.

url : https://deploiementapiissam.alwaysdata.net/auth_api/profile.php

1. Requête

Paramètre Authorization avec le JWT Token nécessaire à la connexion

2. Réponse

OK (200)

3. Exemple

```
4. {  
5.     "id": 1,  
6.     "login": "coach1",  
7.     "role": "entraîneur"  
8. }
```