
Manuel utilisateur

Version 3.0

API Challenge M306

Table des matières

API Challenge M306	1
Table des matières	2
Introduction	4
Concepts de base	4
User	4
Challenge	4
Game	4
Historique des modifications	4
Instructions concernant l'utilisation de l'API depuis JavaScript	5
Effectuer une requête GET	5
Effectuer une requête POST	6
Utilisation de jetons d'accès à l'API	6
Fonctionnalités	7
Créer un challenge	7
Description	7
Paramètres	7
Réponses	7
Succès	7
Erreur	7
Créer un utilisateur	8
Description	8
Paramètres	8
Réponse	8
Succès	8
Erreur	8
Se connecter	9
Description	9
Paramètres	9
Réponses	9
Succès	9
Erreur	9
Sauver un score	10
Description	10
Paramètres	10
Réponse	10
Succès	10
Erreurs	10

Obtenir tous les challenges	11
Description	11
Réponse	11
Succès	11
Obtenir le classement d'un challenge	12
Description	12
Paramètres	12
Réponses	12
Succès	12
Erreur	12
Ajouter un utilisateur à un challenge	13
Description	13
Paramètres	13
Réponses	13
Succès	13
Erreur	13
Obtenir tous les jeux	14
Description	14
Paramètres	14
Réponse	14
Succès	14
Erreur	14
Créer un jeu	15
Description	15
Paramètres	15
Réponse	15
Succès	15
Erreur	15
Obtenir un jeu	16
Description	16
Paramètres	16
Réponse	16
Succès	16
Erreur	16
Ajouter un jeu à un challenge	17
Description	17
Paramètres	17
Réponse	17
Succès	17
Erreur	17

Introduction

Cette documentation vous permettra de vous servir facilement de l'API challenge, qui sert à créer des challenges avec des utilisateurs et des jeux.

Concepts de base

User

Les utilisateurs sont des joueurs qui peuvent participer aux tournois et interagir avec ceux-ci après y avoir été liés. Les utilisateurs ont des jetons d'accès qui leur permettent de s'authentifier.

Challenge

C'est un tournoi à durée définie (avec une date et heure de début et de fin) durant lequel les joueurs doivent obtenir le maximum de points.

N.B : Il est nécessaire de lier des jeux et joueurs (*user*) au challenge après la création de celui-ci.

Game

Correspond à un jeu en ligne créé par la classe I.FA-P1B. Les jeux créés pourront être intégrés à un challenge.

Historique des modifications

Version	Date	Auteurs	Modifications
1.0	04.03.2019	Amandine Pouly < amandine.pl@eduge.ch >	Version initiale du document.
2.0	12.03.2019	Nicolas Ettlin < nicolas.ettlin@eduge.ch > Adamo Cannone < adamo.cnnn@eduge.ch >	Finalisation du document pour qu'il comprenne toutes les fonctionnalités.
3.0	25.03.2019	Nicolas Ettlin < nicolas.ettlin@eduge.ch > Adamo Cannone < adamo.cnnn@eduge.ch >	Clarification des instructions pour l'utilisation de l'API

Instructions concernant l'utilisation de l'API depuis JavaScript

L'API *Challenge M306* est conçue pour être utilisée depuis le code des jeux en ligne créés par la classe I.FA/DA-P1B. Voici quelques exemples d'utilisation d'une API externe depuis JavaScript.

Il existe deux types principaux de points d'accès : les points d'accès GET et les points d'accès POST.

- Les points d'accès GET permettent d'**obtenir** des informations. Ils ne produisent pas d'effets secondaires (cela veut dire qu'utiliser une API en GET n'effectue jamais des modifications dans la base de données).
- Les points d'accès en POST permettent d'**effectuer une action**.

La documentation des fonctionnalités vous indiquera pour chaque point d'accès de quel type il fait partie.

Effectuer une requête GET

Voici comment effectuer une requête GET (ici, nous utilisons le point d'accès `/games/all.php` documenté plus bas dans le document) :

```
fetch('https://api.motak.ch/games/all.php') // Indiquez ici le point
d'accès de l'API
.then(response => response.json()) // Cette ligne indique que le code
s'attend à récupérer des données en format JSON depuis l'API
.then((data) => {
    // Ce code est exécuté dès que la réponse est récupérée depuis le
    serveur.
    // La variable "data" contient les données obtenues.
    // Par exemple, voici comment récupérer les jeux :
    var games = data.games;
});
```

Si vous avez besoin d'indiquer des paramètres dans une requête GET, vous devez le faire directement dans l'URL, comme ceci :

```
fetch('https://api.motak.ch/games/get.php?name=Un jeu 1')
.then(response => response.json())
.then((data) => {
    // ...
});
```

Effectuer une requête POST

La syntaxe utilisée pour effectuer une requête POST est un petit peu différente que celle pour faire une requête GET. Vous devez préciser que vous voulez effectuer une requête POST manuellement, et passer les paramètres dans un objet FormData :

```
var parametres = new FormData();
parametres.append('username', 'TotoLagrenouille'); // Ajoutez ici tous
les paramètres nécessaires...
parametres.append('password', 'Super2012');
parametres.append('firstName', 'Toto');
parametres.append('lastName', 'Lagrenouille');

fetch('https://api.motak.ch/users/create.php', {
  method: 'POST',
  body: parametres,
})
.then(response => response.json())
.then((data) => {
  // Après la création d'un utilisateur, vous pouvez récupérer
  // son jeton d'accès à l'API comme ceci :
  var jeton = data.accessToken;
});
```

Utilisation de jetons d'accès à l'API

Certains points d'accès à l'API nécessitent l'utilisation d'un paramètre "accessToken" : ce paramètre correspond au jeton d'accès d'un utilisateur. Un jeton d'accès est une chaîne de caractères aléatoires qui permet à l'utilisateur de s'authentifier à l'API.

Le jeton d'accès est **unique** pour un utilisateur et ne change jamais.

Vous pouvez obtenir un jeton d'accès de deux manières :

- Lorsque vous vous enregistrez en tant que nouvel utilisateur (avec l'API `users/create/`), vous obtenez un jeton d'accès.
- Lorsque vous utilisez l'API `users/login`, vous obtenez également un jeton d'accès à partir de son nom d'utilisateur et de son mot de passe.

Astuce : vous pouvez stocker en mémoire le jeton d'accès d'un utilisateur, pour ne pas devoir obtenir le jeton à chaque fois. Nous vous recommandons d'utiliser la propriété `localStorage` fournie par le langage JavaScript. Celle-ci vous permet de stocker des chaînes de caractères qui seront conservées même après un rafraîchissement de la page.

La documentation sur l'utilisation de `localStorage` est disponible sur MDN :

<https://developer.mozilla.org/fr/docs/Web/API/Window/localStorage>

Fonctionnalités

Créer un challenge

Endpoint (POST) : <https://api.motak.ch/challenges/create.php>

Description

Vous pouvez créer votre propre challenge grâce à cette fonction.

Paramètres

name : string

accessToken : string

startDate: string (au format : "2019-12-28 10:09:30")

endDate: string (au format : "2019-12-28 10:09:30")

Réponses

Note : le format de toutes les réponses est JSON.

Succès

```
{"status" : "success"}
```

Erreur

Paramètre startDate/endDate manquant:

```
{"status": "error", "errors":["Missing startDate o  
endDate parameter"]}
```

Paramètre startDate invalide (commence avant la date/heure actuelle):

```
{"status": "error", "errors":["Invalid startDate"]}
```

Paramètre token invalide:

```
{"status": "error", "errors":["Invalid token"]}
```

Paramètre name/token manquant :

```
{"status": "error", "errors":["Missing name or token"]}
```

Créer un utilisateur

Endpoint (POST) : <https://api.motak.ch/users/create.php>

Description

Vous pouvez créer votre propre utilisateur grâce à cette fonction. Elle vous retournera votre access token.

Attention gardez votre access token quelque part, vous devrez le réutiliser !

Paramètres

username : string

password : string

firstName : string

lastName : string

Réponse

Note : le format de toutes les réponses est JSON.

Succès

```
{
  "successes" : ["User successfully registered"],
  "accessToken" : "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "username" : "Test"
}
```

Erreur

Username déjà utilisé :

```
{"errors":["Username already taken"]}
```

Paramètres manquants :

```
{"errors":["Missing parameter password"]}
```

Veuillez spécifier une valeur pour le paramètre mot de passe.

```
{"errors":["Missing parameter first_name"]}
```

Veuillez spécifier une valeur pour le paramètre prénom.

```
{"errors":["Missing parameter last_name"]}
```

Veuillez spécifier une valeur pour le paramètre nom.

```
{"errors":["Missing parameter username"]}
```

Veuillez spécifier une valeur pour le paramètre nom d'utilisateur.

Se connecter

Endpoint (POST) : <https://api.motak.ch/users/login.php>

Description

Vous pouvez vous loguer grâce à cette fonction. Elle vous retourne le même access token que lors de l'inscription.

Attention gardez votre access token quelque part, vous devrez le réutiliser !

Paramètres

username : string

password : string

Réponses

Note : le format de toutes les réponses est JSON.

Succès

```
{
  "successes":["User successfully logged"],
  "access_token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
}
```

Erreur

Paramètres manquants :

```
{"errors":["Missing parameter username"]}
```

Veuillez spécifier une valeur pour le paramètre nom d'utilisateur

```
{"errors":["Missing parameter password"]}
```

Veuillez spécifier une valeur pour le paramètre mot de passe

```
{"errors":["Missing parameter username",
```

```
 {"Missing parameter password"]}
```

Veuillez spécifier les valeurs pour les paramètres nom d'utilisateur et mot de passe

Paramètres invalides:

```
{"errors":["Wrong username or password"]}
```

Les identifiants fournis ne correspondent à aucun compte présent dans la base de données.

Sauver un score

Endpoint (POST) : <https://api.motak.ch/games/save-score.php>

Description

Vous pouvez sauver un score grâce à cet endpoint.

Paramètres

score : int

challengeId : string

gameId : string

accessToken : string

Réponse

Note : le format de toutes les réponses est JSON.

Succès

```
{"successes":["Score has been registred"]}
```

Erreurs

Paramètres manquants :

```
{"errors":["Missing parameter score"]}
```

Veuillez spécifier une valeur pour le paramètre score.

```
{"errors":["Missing parameter challengeId"]}
```

Veuillez spécifier une valeur pour le paramètre challengeId.

```
{"errors":["Missing parameter gameId"]}
```

Veuillez spécifier une valeur pour le paramètre gameId.

```
{"errors":["Missing parameter accessToken"]}
```

Veuillez spécifier une valeur pour le paramètre accessToken.

Paramètres invalides :

```
{"errors":["Cannot find challenge from challengeId value"]}
```

```
{"errors":["Cannot find challenge from gameId value"]}
```

```
{"errors":["Cannot find user from accessToken value"]}
```

Obtenir tous les challenges

Endpoint (GET) : <https://api.motak.ch/challenges/all.php>

Description

Vous pouvez récupérer tous les challenges grâce à cette fonction.

Réponse

Note : le format de toutes les réponses est JSON.

Succès

```
{
  "challenges" : [
    {
      "id": 42,
      "name": "Mon super challenge",
      "startDate": "2019-12-28 10:09:30",
      "endDate": "2019-12-30 09:41:42",
    }
    // ...
  ],
}
```

Obtenir le classement d'un challenge

Endpoint (GET) :

[https://api.motak.ch/challenges/ranking.php?challengeId=\[id\]&accessToken=\[VotreToken\]](https://api.motak.ch/challenges/ranking.php?challengeId=[id]&accessToken=[VotreToken])

Description

Vous pouvez obtenir le classement concernant un challenge grâce à cette fonction.

Paramètres

accessToken: string

challengeId: string

Réponses

Note : le format de toutes les réponses est JSON.

Succès

```
{
  "successes":["Ranking received"],
  "ranking": [{
    "username" : "Test",
    "first_name" : "John",
    "last_name" : "Appleseed",
    "score" : 42
  }]
}
```

Erreur

Paramètres manquants :

```
{"errors":["Missing parameter accessToken"]}
```

Veuillez spécifier une valeur pour le token

```
{"errors":["Missing parameter challengeId"]}
```

Veuillez spécifier une valeur pour le paramètre challengeId.

Paramètres invalides:

```
{"errors":["Cannot find challenge from challengeId value"]}
```

```
{"errors":["Cannot find user from accessToken value"]}
```

Ajouter un utilisateur à un challenge

Endpoint (POST) : <https://api.motak.ch/users/add-to-challenge.php>

Description

Vous pouvez obtenir le classement concernant un challenge grâce à cette fonction.

Paramètres

accessToken: string

challengeId: string

Réponses

Note : le format de toutes les réponses est JSON.

Succès

```
{
  "status": "success",
}
```

Erreur

Paramètres invalides :

```
{
  "status": "error",
  "errors":["Invalid token"]
}
```

Veuillez spécifier une valeur pour le token

Paramètres manquants :

```
{
  "status": "error",
  "errors":["challengeId and accessToken are required"]
}
```

Veuillez spécifier une valeur pour le token et challengeId

Paramètres invalides:

```
{"status": "error", "errors":["Invalid token"]}
{"status": "error", "errors":["Database error occurred."]}
```

Obtenir tous les jeux

Endpoint (GET) : <https://api.motak.ch/games/all.php>

Description

Vous pouvez obtenir tous les jeux de la base de données avec cette API.

Paramètres

Aucun paramètre

Réponse

Note : le format de toutes les réponses est JSON.

Succès

```
{
  "games": [
    {
      id: 42,
      name: "Mon super jeu",
    },
    // ...
  ],
}
```

Erreur

Ce point d'accès ne retourne jamais d'erreur.

Créer un jeu

Endpoint (POST) : <https://api.motak.ch/games/create.php>

Description

Ce point d'accès permet de créer un jeu.

Paramètres

gameName : string

accessToken : string

Réponse

Note : le format de toutes les réponses est JSON.

Succès

```
{
  "status": "success"
}
```

Erreur

```
{"status": "error", "errors": ["Invalid access token"]}
```

Veillez utiliser un jeton d'accès valide.

```
{"status": "error", "errors": ["Missing game name or token"]}
```

Veillez vous assurer que vous utilisez les paramètres demandés.

Obtenir un jeu

Endpoint (GET) : <https://api.motak.ch/games/get.php>

Description

Ce point d'accès permet d'obtenir les informations d'un jeu spécifique.

Paramètres

name : string

Réponse

Note : le format de toutes les réponses est JSON.

Succès

```
{
  "successes": ["Game found"],
  "game": {
    "id": 42,
    "name": "Mon Super Jeu"
  },
}
```

Erreur

```
{"errors": ["Missing parameter name"]}
```

Veuillez renseigner le paramètre name.

```
{"errors": ["Cannot find game from name value"]}
```

Impossible de trouver un jeu à partir de ce nom.

Ajouter un jeu à un challenge

Endpoint (POST) : <https://api.motak.ch/games/add-to-challenge.php>

Description

Ce point d'accès permet d'associer un jeu à un challenge spécifique.

Paramètres

accessToken : string

challengeId : int

gameId : int

Réponse

Note : le format de toutes les réponses est JSON.

Succès

```
{
  "successes": ["Game added to challenge"]
}
```

Erreur

```
{"errors":["Missing parameter accessToken"]}
```

Veuillez renseigner une valeur pour le paramètre accessToken.

```
{"errors":["Missing parameter gameId"]}
```

Veuillez renseigner une valeur pour le paramètre gameId.

```
{"errors":["Missing parameter challengeId"]}
```

Veuillez renseigner une valeur pour le paramètre challengeId.

```
{"errors":["Cannot find user from accessToken value"]}
```

Le token d'accès n'existe pas dans la base de données. Veuillez renseigner une valeur valide.

```
{"errors":["Cannot find game from gameId value"]}
```

Le jeu n'existe pas dans la base de données. Veuillez renseigner un ID de jeu existant.

```
{"errors":["Cannot find challenge from challengeId value"]}
```

Le challenge n'existe pas dans la base de données. Veuillez renseigner un ID de challenge existant.