Manuel de l’API

Projet « Train Commander » Supinfo 2015 – 2016

Sommaire

[Avant-propos 2](#_Toc452814287)

[Fonctions 3](#_Toc452814288)

[Fonctions GET 3](#_Toc452814289)

[Fonctions CREATE 4](#_Toc452814295)

[Fonctions de connexion et de recherche 5](#_Toc452814301)

# Avant-propos

Si nous utilisons la Web API au sein de notre site et de nos applications mobiles, tout le monde peut lui envoyer des requêtes. Nous discutons de ce choix dans la section dédiée à la Web API dans la documentation technique.

Pour utiliser l’API, il suffit d’aller à l’url voulue sur le sous-domaine « api » du site web.

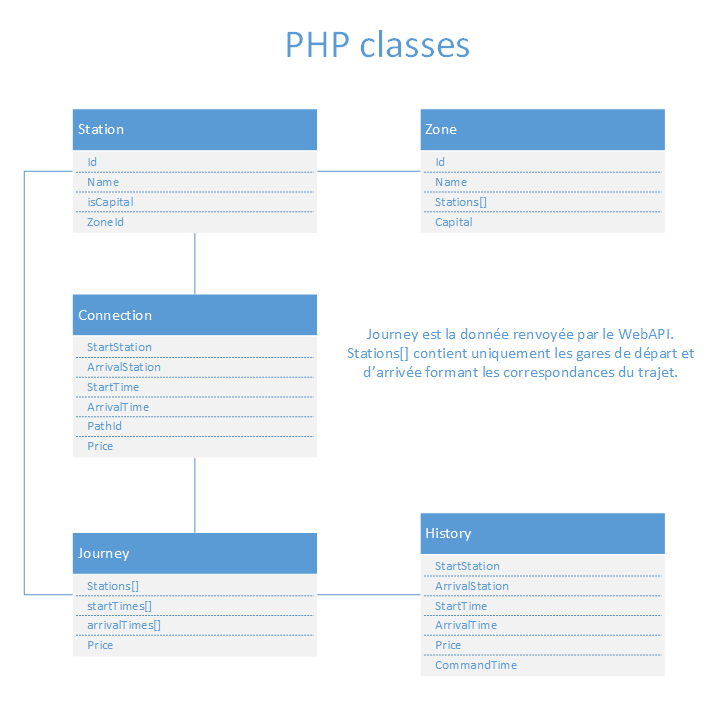
Comme le .htaccess filtre les connexions, une erreur 404 est renvoyée si l’url entrée ne correspond à rien. Une erreur 405 (Invalide request) est retournée si les paramètres passés sont invalides.

Si la requête ne retourne aucun résultat, « null » ou « 0 » sera renvoyé, de manière générale.

Voyez la description des requêtes pour avoir les informations exactes.

Tous les résultats sont renvoyés sous la forme de JSON.

Pour exploiter les résultats, référez-vous au diagramme des classes PHP.



# Fonctions

## Fonctions GET

### /stations

Renvoie une liste de toutes les gares du réseau ferroviaire.

### /user/{id}

Renvoie les données de l’utilisateur dont l’id a été passé en paramètre.

### /user/fb/{fbID}

Renvoie l’id de l’utilisateur dont l’id Facebook est passé en paramètre.

### /user/google/{googleID}

Renvoie l’id de l’utilisateur dont l’id Google est passé en paramètre.

### /history/{id}

Renvoie les commandes de l’utilisateur dont l’id a été passé en paramètre.

## Fonctions CREATE

### /create/user/{firstname}/{lastname}/{password}/{email}/{newsletter}

Crée un utilisateur dans la base de données avec les informations passées en paramètres.

Newsletter est un booléen (0 ou 1) qui définit si l’utilisateur souhaite s’abonner à la newsletter.

Non par défaut.

Le mot de passe est encrypté avant d’être passé à la base de données.

Retourne l’id de l’utilisateur créé.

Exemple

/create/user/Florent/Diez/supinfo/florent.diez@supinfo.com/1

### /create/user/fb/{firstname}/{lastname}/{FacebookID}

### /create/user/google/{firstname}/{lastname}/{GoogleID}

Crée un utilisateur dans la base de données avec les informations passées en paramètres.

À utiliser pour la connexion par Facebook ou Google.

Retourne l’id de l’utilisateur créé.

Exemple

/create/user/Florent/Diez/16845124608

### /create/history/{cost}/{startStationName}/{arrivalStationName}/{startTimestamp}/

### {arrivalTimestamp}/{userid}

Crée une commande dans la base de données avec les informations passées en paramètres.

Le dernier paramètre est l’id de l’utilisateur ayant passé la commande.

Retourne l’id de la commande créée.

Exemple

/create/history/65.0/Paris/Marseille/1467262800/1467265500/17

## Fonctions de connexion et de recherche

### /connect/{email}/{password}

Renvoie l’id de l’utilisateur si les identifiants de connexion sont corrects. Sinon, renvoie 0.

### /journey(s)/{searchMode}/{startStationID}/{endStationID}/{startTimestamp}

Renvoie le ou les meilleur(s) voyage(s) en fonction du mode de recherche choisi.

« cost » permet de chercher le(s) trajet(s) le(s) moins cher(s).

« time » permet de chercher le(s) trajet(s) le(s) plus rapide(s).

Les trajets seront compris entre une heure avant l’heure indiquée et jusqu’à deux heures après.

Renvoie une liste de voyages.

Exemple

/journeys/cost/10/34/1467262800

### /journeys/{searchMode}/{startStationID}/{endStationID}/{minTimestamp}/

### {maxTimestamp}

Renvoie les meilleurs voyages en fonction du mode de recherche choisi.

Fonctionne comme son homologue présentée ci-dessus, à l’exception des bornes temporelles.

Les voyages seront recherchés strictement dans les bornes passées en paramètres.

Renvoie une liste de voyages.

Exemple

/journeys/cost/10/34/1467262800/1467265500