### Authentification

Système de token généré lors de la connexion et possédant un timeout. À chaque requête, le token est envoyé, c’est pourquoi celles-ci doivent être en https. Le timeout est « reseté » à chaque requête.

### Page de login

|  |  |
| --- | --- |
| Requête envoyée | Réponse |
| POST URLBackend/connexion  Avec comme données :   * L’adresse email CPNV * Le mot de passe CPNV | Le token à utiliser. Celui qui est stocké sur la DB propre à covoiturage et correspond à un utilisateur. |

### Déconnexion

|  |  |
| --- | --- |
| Requête envoyée | Réponse |
| POST URLBackend/deconnexion  Avec comme données :   * Token | Indication comme quoi il est déconnecté. |

### Page blog

Utilisation de la page Facebook, pas d’api perso

### Page de données personnelles

|  |  |  |
| --- | --- | --- |
| Cas | Requête envoyée | Réponse |
| On arrive sur la page | GET URLBackend/users/:id   * Token | Les données personnelles  Si l’utilisateur qui demandeur n’est pas le user, renvoyer moins d’info |
| On met à jour les data | PUT URLBackend/users/:id  Avec comme données :   * Token * Les nouvelles data (on choisit de tout renvoyer) | Les données personnelles |

### Page about

Pas d’échanges. Ecrit en dur dans l’application

### Page d’accueil

Pas d’échanges. Ecrit en dur dans l’application

### Page création de trajet

Sert aussi pour la modification d’un trajet

|  |  |  |
| --- | --- | --- |
| Cas | Requête envoyée | Réponse |
| On créée un trajet | POST URLBackend/rides   * Token | Le ride ?  L’URL du trajet  Une telle URL doit faire pointer l’application sur la page des trajets. |
| On modifie le trajet | PUT URLBackend/rides/:idtrajet   * Token * Le nouveau trajet |  |

### Page mes trajets

|  |  |  |
| --- | --- | --- |
| Cas | Requête envoyée | Réponse |
| On voit tous les trajets qui nous impliquent | GET URLBackend/rides?mine=true   * Token | Les trajets qui nous impliquent (conducteur ou inscrit) |

### Page trajets

|  |  |  |
| --- | --- | --- |
| Cas | Requête envoyée | Réponse |
| On voit tous les trajets | GET URLBackend/rides   * Token | Tous les trajets (combien ?) sauf ceux qu’on conduit |
| On effectue une recherche | GET URLBackend/rides?q=Sainte-Croix   * Token | Les trajets qui correspondent à la recherche (sans les trajets qu’on conduit) |
| On récupère 1 trajet | GET URLBackend/rides/:idtrajet   * Token | Le trajet en question |

### Page trajet

|  |  |  |
| --- | --- | --- |
| Cas | Requête envoyée | Réponse |
| On voit le trajet | GET URLBackend/rides/:idtrajet   * Token | Le trajet demandé |
| On modifie ses inscriptions à un trajet | PUT URLBackend/rides/ :idtrajet/registrations/   * Token * La date (ou les dates ?)   Un seul appel pour la modification et la désinscription ? |  |
| On supprime le trajet | DELETE URLBackend/rides/:idtrajet   * Token (on ne peut pas supprimer n’importe quel trajet) |  |

### Gestion d’erreurs

Renvoyer le code d’erreur correspondant.

401 pour un problème d’authentification

403 lorsque la personne essaie de réaliser des actions qui lui sont interdites

404 lorsque la ressource n’existe pas

Un utilisateur en tant que l’utilisateur lui-même

[

Nom

Prénom

Email

Phone

Confidentialité [

ShowEmail

ShowPhone

]

Notification [

Notif1

Notif2

…

]

]

Un utilisateur en tant qu’un autre utilisateur

[

Nom

Prénom

Email (si l’utilisateur veut le montrer)

Phone (si l’utilisateur veut le montrer)

]

Un trajet en tant que monsieur tout le monde

(Seulement pour la page du ride ou aussi pour la page du listing des trajets ?)

Il faudrait savoir si le conducteur a tout le temps le droit d’avoir accès aux infos personnelles des utilisateurs inscrits à son trajet (téléphone et email personnel)

[

driver

departuretown

[

Id

name

]

departure

arrivaltown

[

Id

name

]

arrival

startdate

enddate

description

recurrence [

monday

tuesday

…

]

registrations [

15-12-2014 [(la liste des utilisateurs)

[

firstname

lastname

],

[

firstname

lastname

],

…

],

…

]

]

Un trajet en tant que conducteur ressemble furieusement au trajet présenté précédemment sauf que les utilisateurs possèdent également leur numéro de téléphone et leur email.

Plan de travail pour savoir quelles sont les informations à récupérer lors d’un GET de trajets :

1. Créer 100 utilisateurs
2. Créer 200 trajets
3. Créer 500 enregistrements
4. Tester la récupération de 10, 50, 100, 200 trajets
   1. Mesurer les temps de réponse
   2. Mesurer le poids des réponses
   3. Evaluer l’augmentation des temps et poids en fonction du nombre d’utilisateurs, trajets et enregistrements
5. En fonction des données récoltées précédemment, déterminer les informations à indiquer par trajet et le nombre qu’il faut en charger
6. S’arranger pour que ces paramètres soit facilement modifiables (indiqués dans le fichier config par exemple)