### API

Notre application mobile doit pouvoir récupérer des données de la plateforme de covoiturage afin d’afficher différentes informations telles que les prochains trajets auxquels il est possible de s’inscrire, les trajets auxquels nous sommes déjà inscrits, etc. Le besoin de créer une api s’est donc fait sentir-> mmm y’a comme une odeur de besoin de créer une api XD. Nous pensons qu’adopter le système REST et communiquer en JSON nous semble la meilleure option : le JSON étant particulièrement bien manipulé par AngularJS et REST car il s’agit d’une architecture éprouvée et adoptée par grand nombre d’acteurs de l’informatique et qu’elle convient parfaitement à nos besoins tout en étant simple à comprendre et à mettre en place.

L’API fut élaborée en tenant évidemment compte des différentes possibilités offertes par l’application mobile, possibilités déterminées à l’aide des maquettes crées au début du projet. Cette méthode de travail nous permet, selon nous, d’éviter un grand nombre d’imprécisions ou d’oublis même si nous ne sommes pas à l’abri de bugs ou de n’avoir pas pleinement pris en compte tous les cas d’utilisation.

Le premier jet de l’api fut créé par nos soins. Nous avons ensuite demandé l’avis de Monsieur Hurni étant donné que la création d’une API REST était nouveau pour nous. Nous avons ainsi pu éviter un certain nombre d’aberrations tout en apprenant quelque chose qui nous sera probablement utile pour notre futur professionnel.

**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.

**Login**

|  |  |
| --- | --- |
| Requête | POST URLServeur/users/connexion |
| Paramètres | email : email CPNV  password : mot de passe CPNV |
| Réponse | id de l’utilisateur  token de l’utilisateur |
| Exemple | *Requête angularjs :*  $http.post('https:covoiturag.cpnv-es.ch/api/users/connexion',  {"email": "test@cpnv.ch", "password":"testtest"})  *Réponse :* |
| Erreurs |  |

|  |  |
| --- | --- |
| Requête envoyée | Réponse |
| POST URLServeur/users/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.  L’id de l’utilisateur dans le SGBD |

**Déconnexion**

|  |  |
| --- | --- |
| Requête envoyée | Réponse |
| POST URLServeur/users/deconnexion  Avec comme données :  • Token | Indication comme quoi il est déconnecté. |

**Page blog**

Utilisation de la page Facebook, ne passe pas par notre api.

**Page de données personnelles**

|  |  |  |
| --- | --- | --- |
| Cas | Requête envoyée | Réponse |
| On arrive sur la page | GET URLServeur/users/:id  Avec comme données :   * Token | Les données personnelles  (Si l’utilisateur qui demandeur n’est pas le user, renvoyer moins d’informations) |
| On met à jour ses données personnelles | PUT URLServeur/users/:id  Avec comme données :   * Token * Les nouvelles données de l’utilisateur (ex : numéro de téléphone,…) (on choisit de tout renvoyer) | Les données personnelles  (On peut uniquement modifier ses propres informations) |

**Page de données personnelles**

|  |  |  |
| --- | --- | --- |
| Cas | Requête envoyée | Réponse |
| On arrive sur la page | GET URLServeur/towns ?q=…  Avec comme données :   * Token | Une liste de villes qui correspondent à la recherche avec leur id respectif |

**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 URLServeur/rides   * Token | Le trajet  . |
| On modifie le trajet | PUT URLServeur/rides/:idtrajet   * Token * Le nouveau trajet | Le trajet |

**Page mes trajets**

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

**Page trajets**

|  |  |  |
| --- | --- | --- |
| Cas | Requête envoyée | Réponse |
| On voit tous les trajets | GET URLServeur/rides   * Token | La liste des trajets (sans les trajets qu’on conduit) |
| On effectue une recherche | GET URLServeur/rides?q=Aigle   * Token * Le texte à rechercher | Les trajets qui correspondent à la recherche (sans les trajets qu’on conduit) |

**Page trajet**

|  |  |  |
| --- | --- | --- |
| Cas | Requête envoyée | Réponse |
| On voit le trajet | GET URLServeur/rides/:idtrajet   * Token | Le trajet demandé avec les registrations de tout le monde |
| On modifie ses inscriptions à un trajet | PUT URLServeur/rides/:idtrajet/registrations   * Token * La date (ou les dates) | Les registrations |
| On supprime le trajet | DELETE URLServeur/rides/:idtrajet   * Token | Le status 200 (On ne peut évidemment pas supprimer les trajets de tout le monde) |

Requêtes :

GET rides mine : <http://covoiturag.cpnv-es.ch/api/rides?token=asdf&mine=true>

GET rides : <http://covoiturag.cpnv-es.ch/api/rides?token=asdf>

GET ride : <http://covoiturag.cpnv-es.ch/api/rides/2?token=asdf>

Search rides : <http://covoiturag.cpnv-es.ch/api/rides?token=asdf&q=Gettnau>

PUT ride registrations <http://covoiturag.cpnv-es.ch/api/rides/128/registrations?token=asdf>

Axel GET Elliott <http://covoiturag.cpnv-es.ch/api/users/2?token=asdf>

PUT données Axel <http://covoiturag.cpnv-es.ch/api/users/1?token=asdf>

GET données Axel <http://covoiturag.cpnv-es.ch/api/users/1?token=asdf>

POST connexion <http://covoiturag.cpnv-es.ch/api/users/connexion>

**Status de réponse utilisés**

* 201 : Lors de la création d’une ressource (en général un trajet)
* 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

**Représentations des ressources**

Un utilisateur lorsqu’il demande ses paramètres :



Un utilisateur lorsqu’il est demandé par un autre utilisateur (son email et téléphone est seulement transmis s’il ne les cache pas) :

