

Rapport projet « Toilettes »

Programmation mobile et web services

Table des matières

| | |
|--|-----------|
| Sujet et but de l'application | 3 |
| Notice de déploiement | 4 |
| Installation serveur (API) | 4 |
| Utilisation: | 4 |
| Guide d'utilisation..... | 5 |
| Documentation de l'API | 8 |
| Get all toilets around user position..... | 8 |
| Create a new toilet..... | 9 |
| Get a specific toilet..... | 11 |
| Update a specific toilet | 11 |
| Get a specific toilet's picture..... | 13 |
| Get all comments of a specific toilet..... | 14 |
| Add new comment for a specific toilet | 14 |
| Get a specific comment of a toilet..... | 15 |
| Update a specific comment of a toilet..... | 15 |
| Register a new user | 17 |
| Authenticate a user | 17 |
| Get data of specific user account | 18 |
| Update data of specific user account..... | 18 |
| Diagramme de use cases | 20 |
| Diagramme de classes | 20 |

Sujet et but de l'application

Toilettes est une application mobile faite avec le framework Ionic 2 qui interroge une API que nous avons créée à l'aide de l'environnement de développement NodeJS et de divers modules.

Le but de l'application est de permettre aux utilisateurs de trouver les toilettes les plus proches de sa localisation.

Les fonctionnalités

- Montrer les toilettes les plus proches de ma position
- Ajouter ou mettre à jour
- Commenter et noter
- S'inscrire sur l'application
- Se connecter
- Mettre à jour son compte

Notice de déploiement

Installation client

Pour que que l'application, il faut que le serveur soit lancé.

1. Lancer la commande 'npm install' dans le dossier Toiletttes du dossier ionic_project
2. Lancer la commande 'ionic build'.
3. Lancer ensuite la commande : **ionic plugin add cordova-plugin-googlemaps --variable API_KEY_FOR_ANDROID="AlzaSyDE99utD1I0leasTivb7AuNw_Qk1DzSY2c" --variable API_KEY_FOR_IOS="AlzaSyDE99utD1I0leasTivb7AuNw_Qk1DzSY2c"**
4. Lancer la commande 'ionic run android' ou 'ionic run ios'

Installation serveur (API)

Tout d'abord, il faut que NodeJS soit installé puis il faut suivre les étapes suivantes :

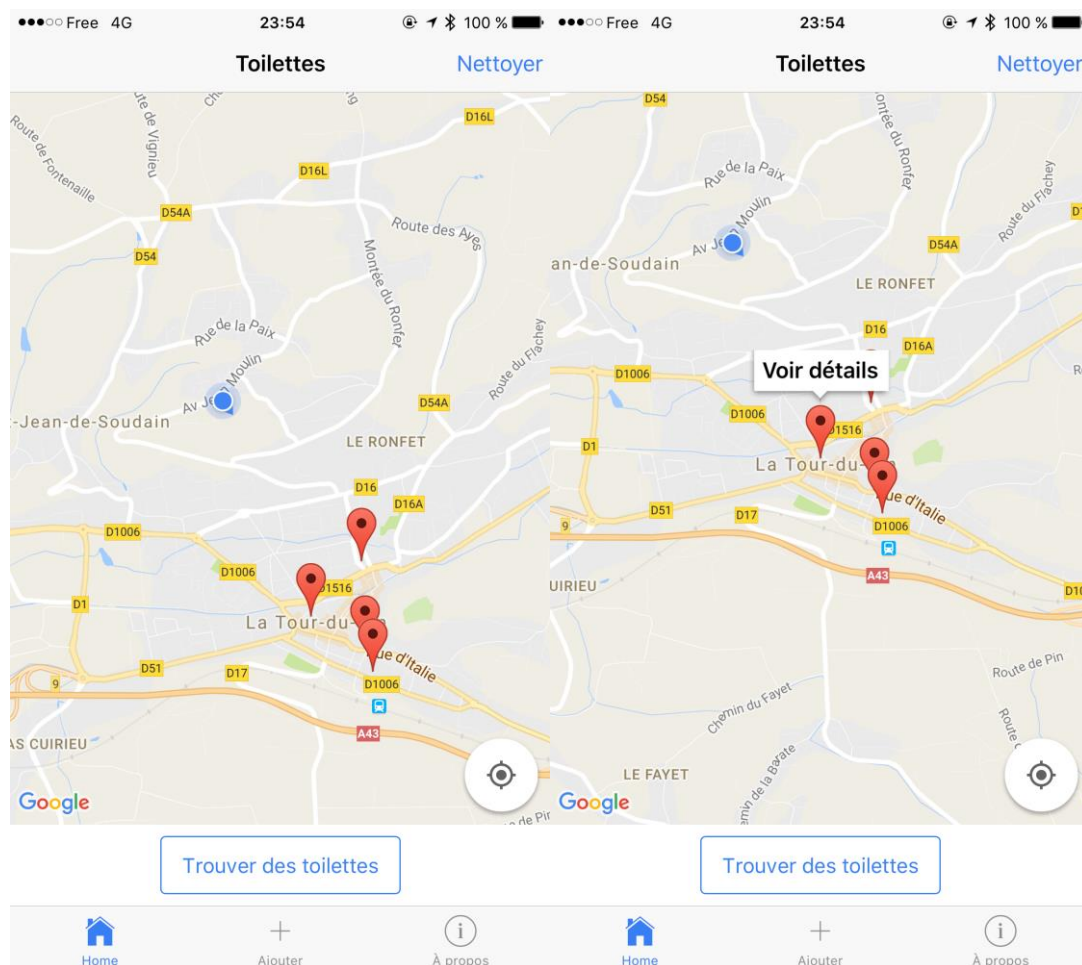
1. Cloner ou télécharger le projet <https://github.com/AhmetCelikbas/Toilettes.git>
2. Ouvrir une fenêtre de commandes à la racine du dossier nodejs_api
3. Lancer la commande 'npm install' pour récupérer les dépendances du projet
4. Lancer la commande 'sequelize db:migrate' pour créer la base de données
5. Enfin lancer la commande 'node app.js' pour démarrer le serveur

Guide d'utilisation

L'application est simple d'utilisation, elle dispose de trois onglets : Home, Ajouter et About

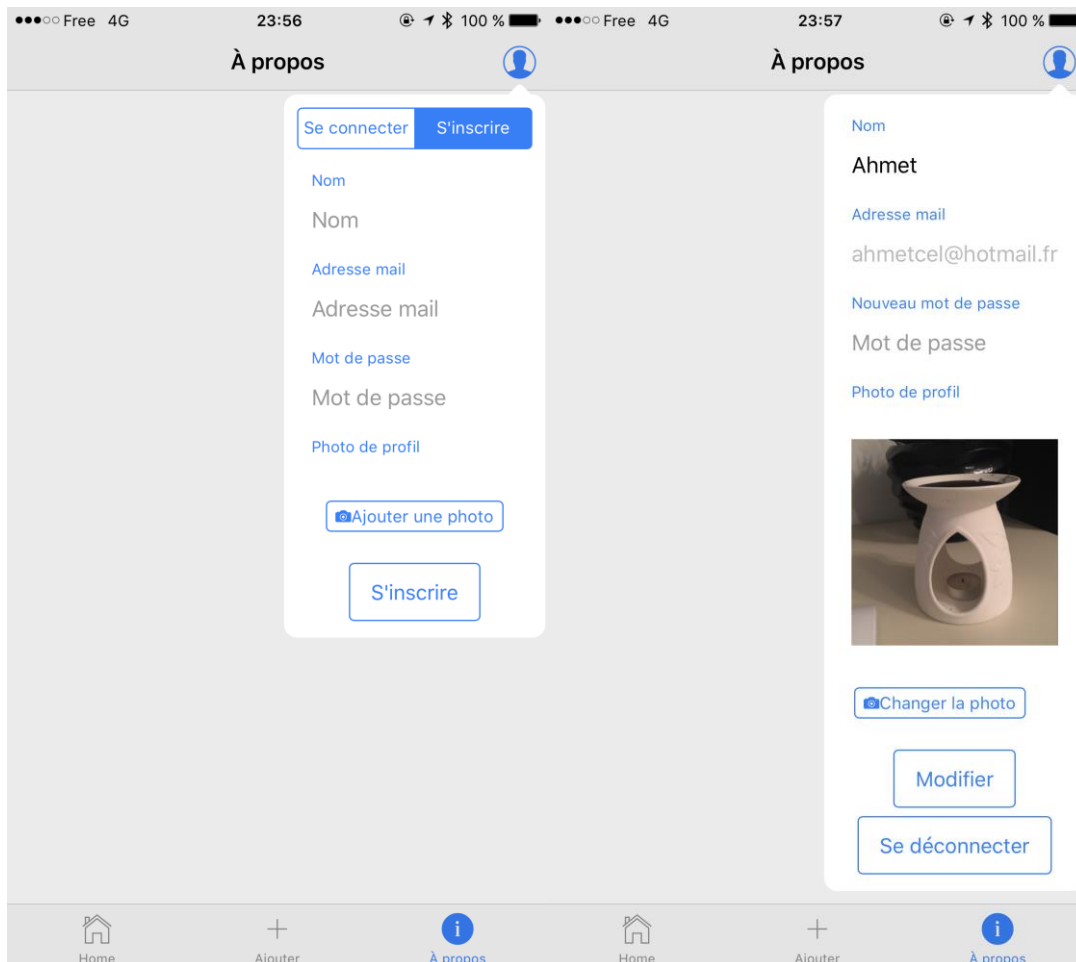
Home montre les toilettes proches de l'utilisateur sur une carte, Ajouter permet d'ajouter des nouvelles toilettes sur la position gps de l'utilisateur et About permet de se connecter ou modifier son compte

Sur l'onglet home cliquer sur le bouton trouver des toilettes. Chaque marqueur correspond à des toilettes. Le point bleu indique la position de l'utilisateur. Cliquer sur un marqueur pour sélectionner les toilettes correspondant au marqueur puis cliquer sur 'Voir détails' pour afficher les détails de ces toilettes.



Rendez-vous ensuite dans l'onglet About puis cliquer sur l'icone 'profile' en haut a droite.

Sélectionnez s'inscrire et renseignez le formulaire en y ajoutant une photo si vous le désirez. Cliquez ensuite sur le bouton s'inscrire. Lorsque vous cliquerez à nouveau sur l'icone profile vous pourrez modifier votre compte pour changer votre nom, votre mot de passe ou encore votre photo.



Une fois connecté avec votre compte vous pourrez modifier des détails de toilettes ou encore ajouter de nouvelles toilettes avec l'onglet Ajouter



Documentation de l'API

API prefix : "/v1/api"

Get all toilets around user position

GET /toilets/:southWestLat/:southWestLon/:northEastLat/:northEastLon

Parameters

| Name | Type | Description |
|------------------------------|------|--|
| southWestLat southWestLon | real | Location of the bottom left of the bounding box where are all toilets around our position |
| northEastLat northEastLon | real | Location of the top right of the bounding box where are all toilets around our position |

Response

Status: 200 OK

```
[  
  {  
    "id": 1,  
    "id_osm": 1966868061,  
    "id_user": 1,  
    "lat": 45.1974288,  
    "lng": 5.776482,
```

```
    "Details": null,  
    "Comment": []
```

```
  },  
  {  
    "id": 2,  
    "id_osm": 773538141,  
    "id_user": 1,  
    "lat": 45.1991028,  
    "lng": 5.7663749,
```

```
    "createdAt": "2017-04-12T21:31:36.869Z",  
    "updatedAt": "2017-04-12T21:31:36.869Z",
```



```
"Details": {
  "id_toilet": 1,
  "name": "",
  "access": true,
  "exist": true,
  "fee": false,
  "male": false,
  "female": false,
  "wheelchair": false,
  "drinking_water": false,
  "place_type": "restaurant",
  "address": "23 Chemin de l'Île d'Amour, 38240 Meylan, France"
```

```
"createdAt": "2017-04-12T21:31:43.607Z",
"updatedAt": "2017-04-12T21:31:43.607Z",
```

```
},
"Comment": [
  {
    "id": 1,
    "id_toilet": 3,
    "id_user": 1,
    "comment": "Mon premier commentaire.",
    "rating": 3
  },
  {
    "id": 2,
    "id_toilet": 3,
    "id_user": 2,
    "comment": "Un autre commentaire.",
    "rating": 4
  }
]
}
]
```

Create a new toilet

POST /toilets

Header: Authorization : token

Input

| Name | Type | Description |
|----------------|---------|---|
| id_osm | int | The identifier of a toilet get from overpass API. |
| lat | double | Latitude of toilet location. |
| lng | double | Longitude of toilet location. |
| Object details | | |
| name | string | The user's email. |
| access | boolean | Indicates whether the toilet is accessible |
| exist | boolean | Indicates whether it exists |
| fee | boolean | Indicates whether it's free |
| male | boolean | Indicates whether it's for male |
| female | boolean | Indicates whether it's for female |
| wheelchair | boolean | Indicates whether it's accessible for wheelchair |
| drinking_water | boolean | Indicates whether you can drink water |
| place_type | string | Indicates type of toilet, eg: 'restaurant' |

Response

Status: 200 OK

```
{
  "id": 3,
  "id_osm": null,
  "id_user": 1,
  "lat": 45.1974276,
  "lng": 5.736442,
  "Details": null
}
```

Get a specific toilet

```
GET /toilet/:id
```

Response

Status: 200 OK

```
{
  "id": 2,
  "id_osm": 773538141,
  "id_user": 1,
  "lat": 45.1991028,
  "lng": 5.7663749,
  "createdAt": "2017-04-12T21:31:36.869Z",
  "updatedAt": "2017-04-12T21:31:36.869Z",
  "Details": {
    "id_toilet": 1,
    "name": "",
    "access": true,
    "exist": true,
    "fee": false,
    "male": false,
    "female": false,
    "wheelchair": false,
    "drinking_water": false,
    "place_type": "restaurant",
    "address": "23 Chemin de l'Île d'Amour, 38240 Meylan, France",
    "createdAt": "2017-04-12T21:31:43.607Z",
    "updatedAt": "2017-04-12T21:31:43.607Z",
  }
}
```

Update a specific toilet

```
PUT /toilet/:id
```

Header: Authorization : token

Input

| Name | Type | Description |
|----------------|---------|--|
| name | string | The user's email. |
| access | boolean | Indicates whether the toilet is accessible |
| exist | boolean | Indicates whether it exists |
| fee | boolean | Indicates whether it's free |
| male | boolean | Indicates whether it's for male |
| female | boolean | Indicates whether it's for female |
| wheelchair | boolean | Indicates whether it's accessible for wheelchair |
| drinking_water | boolean | Indicates whether you can drink water |

Response

Status: 200 OK

```
{
  "id": 2,
  "id_osm": 773538141,
  "id_user": 1,
  "lat": 45.1991028,
  "lng": 5.7663749,
  "createdAt": "2017-04-12T21:31:36.869Z",
  "updatedAt": "2017-04-12T21:41:12.165Z",
  "Details": {
    "id_toilet": 1,
    "name": "",
    "access": true,
    "exist": true,
    "fee": false,
    "male": true,
    "female": true,
    "wheelchair": true,
    "drinking_water": false,
    "place_type": "restaurant",
  }
}
```

```

    "address": "23 Chemin de l'Île d'Amour, 38240 Meylan, France"
    "createdAt": "2017-04-12T21:31:43.607Z",
    "updatedAt": "2017-04-12T21:41:14.632Z",
  }
}

```

Get a specific toilet's picture

```
GET /toilet/:id/picture
```

Response

Status: 200 OK

```
{  
  "picture":  
    "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAMCAgMCAGMDAwMEAwMEBQgFBQQE  
BQoHBWYIDAOmDAsKCwsNDhIQDQ4RDgsLEBYQERMUFRUVDA8XGBYUGBIUFRT/2wBDAQMEBAUEBQkFBQkUDQs  
NFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQU..."  
}
```

Comments endpoints:

Get all comments of a specific toilet

GET /toilet/:id/comments

Response

Status: 200 OK

```
[
  {
    "id": 1,
    "id_toilet": 2,
    "id_user": 1,
    "comment": "Mon premier commentaire.",
    "rating": 3
  },
  {
    "id": 2,
    "id_toilet": 2,
    "id_user": 2,
    "comment": "Un autre commentaire.",
    "rating": 4
  }
]
```

Add new comment for a specific toilet

POST /toilet/:id/comments

Header: Authorization : token

Input

| Name | Type | Description |
|---------|--------|-----------------------------|
| comment | string | A comment about the toilet. |
| rating | int | Give a note for the toilet. |

Response

Status: 200 OK

```
{
  "id": 1,
  "id_toilet": 2,
  "id_user": 1,
  "comment": "Mon premier commentaire.",
  "rating": 3
}
```

Get a specific comment of a toilet

GET /toilet/:id_toilet/comment/id

Response

Status: 200 OK

```
{
  "id": 1,
  "id_toilet": 2,
  "id_user": 1,
  "comment": "Mon premier commentaire.",
  "rating": 3
}
```

Update a specific comment of a toilet

PUT /toilet/:id_toilet/comment/:id

Header: Authorization : token

Input

| Name | Type | Description |
|---------|------|-----------------------------|
| comment | text | A comment for the toilet. |
| rating | int | Give a note for the toilet. |

Response

Status: 200 OK

```
{  
  "id": 1,  
  "id_toilet": 2,  
  "id_user": 1,  
  "comment": "Mon nouveau commentaire modifié.",  
  "rating": 4  
}
```


User endpoints:

Register a new user

POST /signup

Input

| Name | Type | Description |
|----------|--------|----------------------|
| name | string | The user's name. |
| email | string | The user's email. |
| password | string | The user's password. |

Response

Status: 200 OK

```
{
  "token":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiZW1haWwiOiJhQGluY29tliwicGFzc3dvcmQiOiJhem
  UiLCJpYXQiOiJlMDIwMDIyMDYsImV4cCI6MTQ5MjA4ODYwNn0.0zF9r6v4dOJjqxQaV7BOATQ6j5U64zbsEca
  LdR5K22g"
}
```

Authenticate a user

POST /authenticate

Input

| Name | Type | Description |
|----------|--------|----------------------|
| email | string | The user's email. |
| password | string | The user's password. |

Response

Status: 200 OK

```
{
  "token":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiZW1haWwiOiJhQGluY29tIiwicGFzc3dvcmQiOiJhem
  UiLCJpYXQiOiJEOTIwMDIyMDYsImV4cCI6MTQ5MjA4ODYwNn0.0zF9r6v4dOJjqxQaV7BOATQ6j5U64zbsEca
  LdR5K22g"
}
```

Get data of specific user account

GET /user

Header: Authorization : token

Response

Status: 200 OK

```
{
  "id": 1,
  "name": "Batman",
  "email": "mail@gmail.com",
  "password": "azerty",
  "createdAt": "2017-04-12 10:43:51.000 +00:00",
  "updatedAt": "2017-04-12 10:43:51.000 +00:00"
}
```

Update data of specific user account

PUT /user

Header: Authorization : token

Input

| Name | Type | Description |
|----------|--------|----------------------|
| name | string | The user's name. |
| password | string | The user's password. |

Response

Status: 200 OK

```
{
  "id": 1,
  "name": "Superman",
  "email": "mail@gmail.com",
  "password": "qwerty",
  "createdAt": "2017-04-12 10:43:51.000 +00:00",
  "updatedAt": "2017-04-12 11:03:22.177 +00:00"
}
```

Diagramme de use cases

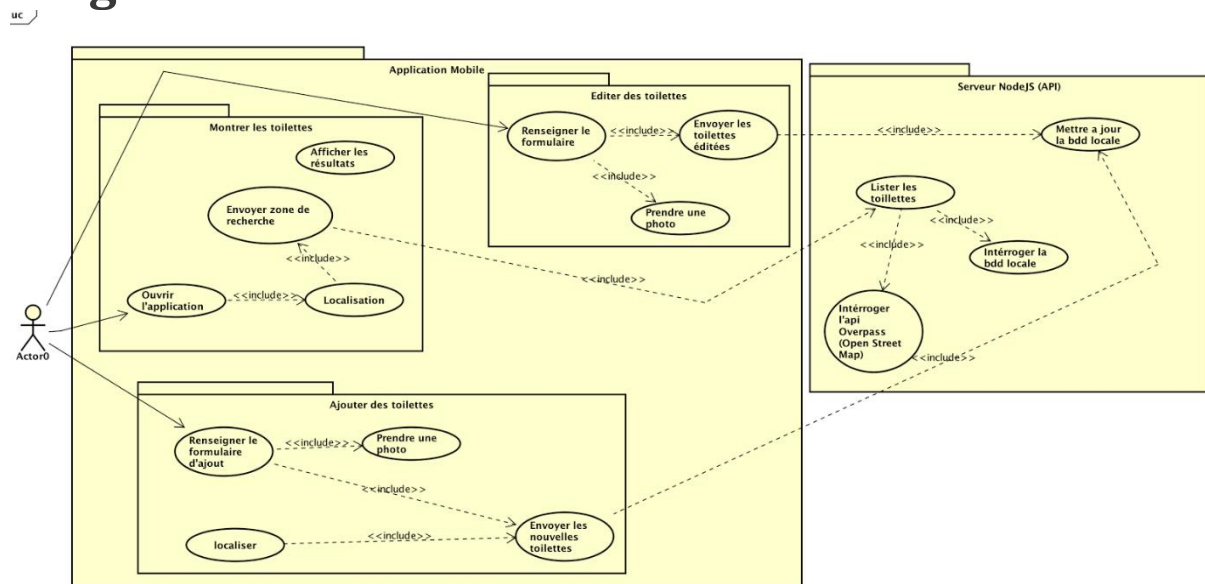


Diagramme de classes

