# Introduction

Mapping My Life is a web application based on Cozy Cloud which allows users to visualize their displacement data more easily and intuitively. The displacement data includes the geolocation data and other communications data like sms and phone calls.

This document will give you a whole concept of this application. It will introduce the environment , the functionality of the application, the framework that used and also the type of data.

# Environment

The application is running on Cozy Cloud which is a platform that brings all your web services in the same private space(<https://cozy.io/>). It is a personal server hosting web applications to manipulate personal data.

In order to develop or test Mapping My Life, you need to install the Cozy V3 environment. You can find the document with more details here (<https://cozy.github.io/cozy-docdev-> [v3/fr/intro.html#présentation-de-la-plateforme](https://cozy.github.io/cozy-docdev-v3/fr/intro.html" \l "présentation-de-la-plateforme)) about how to install the environment.

The most simple way is to use docker.

**Steps:**

1. Install Docker first on your computer

2. Pull the image:

$ docker pull cozy/cozy-app-dev

3. Run the development server:

$ cd mappingmylife

$ docker run --rm -it -p 8080:8080 -p 5984:5984 -v "$(pwd)/build":/data/cozy-app -v "$HOME/db":/usr/local/couchdb/data -v "$HOME/storage":/data/cozy-storage --name=cozydev cozy/cozy-app-dev

Now the application will run on : http://app.cozy.tools:8080/ (password default is cozy).

You can access to CouchDB with URL: http://cozy.tools:5984/\_utils/

# Architecture

Mapping My Life is a client-side application because Cozy will not support server-side application in Version 3. We can use only Cozy Data API(<https://github.com/cozy/cozy-client-js/blob/master/docs/data-api.md>) or Cozy Data System(<https://github.com/cozy/cozy-stack/blob/master/docs/data-system.md>) to interact with CouchDB.

We use React and Redux JavaScript library to build user interface of the application. Since react provides a component based structure, we can re-use the components anywhere so it makes our application scalable and much easier to developer and maintain the code.

Besides, in order to manage JavaScript modules dependencies more easily we use Webpack which is a bundler for modules to bundle JavaScript files for usage in a browser.

You can find webpack configuration files of application Mapping My Life in the folder /config and the file named webpack.config.js in root folder.

**Data Base**

There are three types of data in the data base CouchDB.

### Data format

1. GeoPoint which related to periodic geolocation (SOLO)

{

"docTypeVersion": "test\_data",

"timestamp": "2016-11-14T10:35:10Z",

"docType": "fr.orange.geopoint",

"longitude": 2.294722,

"msisdn": "33688088155",

"radius": 10000,

"latitude": 48.800556

}

”latitude” and “longitude” represent the location

“timestamp” represents when the point is located

“msisdn” represents phone number

“docType” is used to indicate the type of data

2. PhoneCommunicationLog which related to CRA( comptes rendus d’appels)

{

"msisdn": "33688088155",

"networkType": "4G",

"docType": "fr.orange.phonecommunicationlog",

"type": "SMS sortant",

"endCause": "Non significatif",

"latitude": "48.800556",

"chipType": "M",

"longitude": "2.294792",

"timestamp": "2016-11-15T08:59:59",

"partner": "0689629968",

"docTypeVersion": "test\_data",

"length": 1

}

“latitude”, “longitude”, “msisdn”, and “docType” have the same meaning like GeoPoint.

“type” represents the type of message

“partner” represents phone number with which the user contact

3. FavorisPoint which related to the favorite location we add on the map

{

"category": "sport",

"latitude": 48.890833,

"longitude": 2.326944

}

“latitude” and “longitude” represent the location

“category” represents the type of this location(like: home, work, sport, market…)

**Data permission on Cozy**

Since our application is client-side application, when a user installs it, he will be asked to accept an initial set of permission for this app. With the permission, our application can make a request to Cozy to get data from CouchDB.

We configure the permission in manifest.webapp file.

Here is an example of our permission configuration:

{

"permissions": {

"pointfavoris":{

"description": "Required for favorite place",

"type": "fr.orange.pointfavoris",

"verbs": ["GET", "POST", "PUT", "DELETE"]

},

"geopoint":{

"description": "required to manage geolocation information",

"type": "fr.orange.geopoint",

"verbs": ["GET", "POST", "PUT"]

},

"phonecommunication":{

"description": "Required to manage communications information",

"type": "fr.orange.phonecommunicationlog",

"verbs": ["GET", "POST", "PUT"]

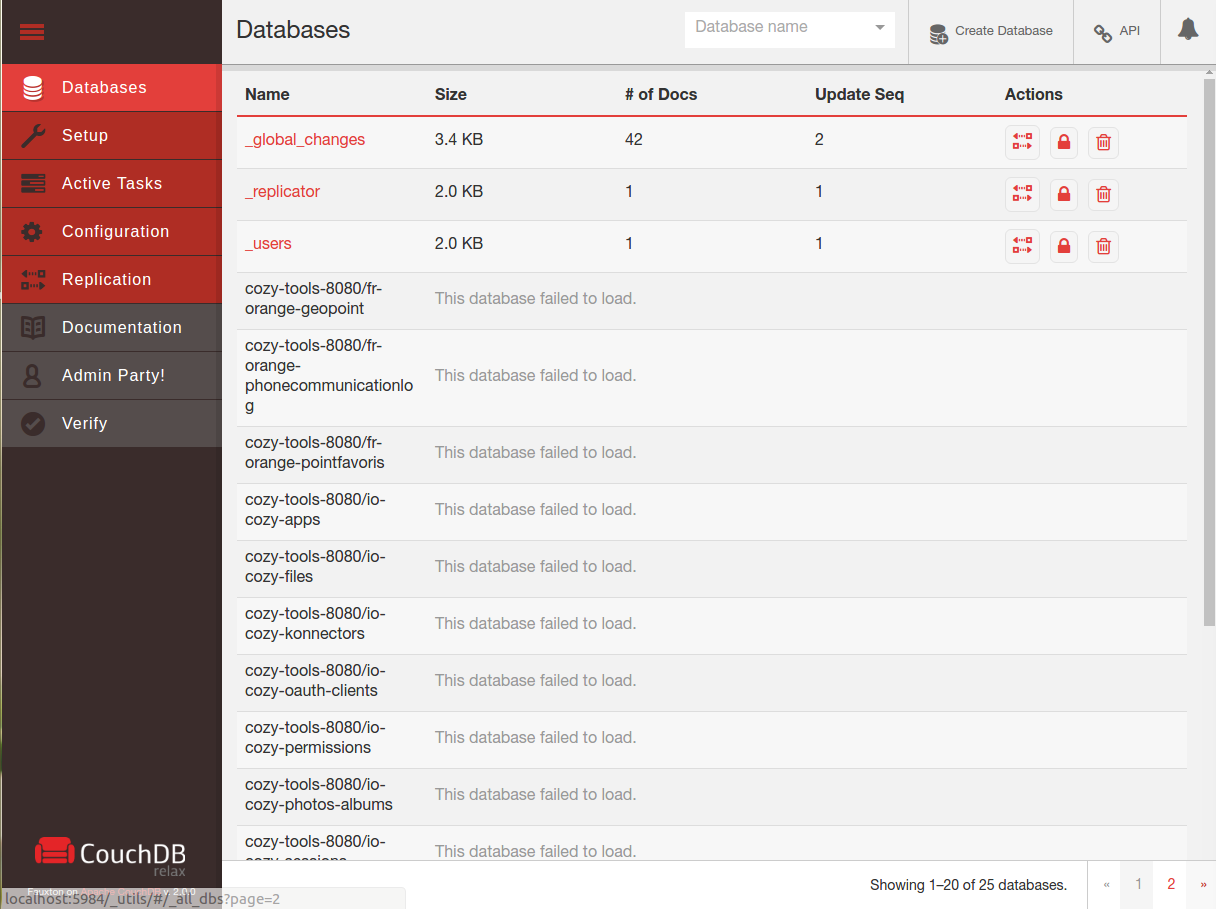
}

}

You can find more information about permission here((<https://github.com/cozy/cozy-stack/blob/master/docs/permissions.md>))

**Manage Database**

You can access to database with URL: <http://localhost:5984/_utils/> once you start to run application on Cozy.



You can use the command below to insert or delete data from CouchDB.

- Insert data in CouchDB

$ DB="<http://127.0.0.1:5984/dbname>"  
$ curl -X POST $DB/\_bulk\_docs -d @/path/GeoPoint.json -H "Content-type:application/json"  
  
- Delete data

$ curl -X DELETE http://127.0.0.1:5984/dbname

# Functionality

This application consists 3 view using React Router for routing.

## **Router**

## The router can be found in main.js file.

## 

## **Views**

**1. HomePage ( with URL ‘/’ )**

The HomePage component can be found in containers/HomePage.js

This page includes two components: HomeMap(components/HomeMap.js) and TimeLineView(components/TimeLineView.js).

The HomeMap displays locations information in form of markers and pop-ups on a map.

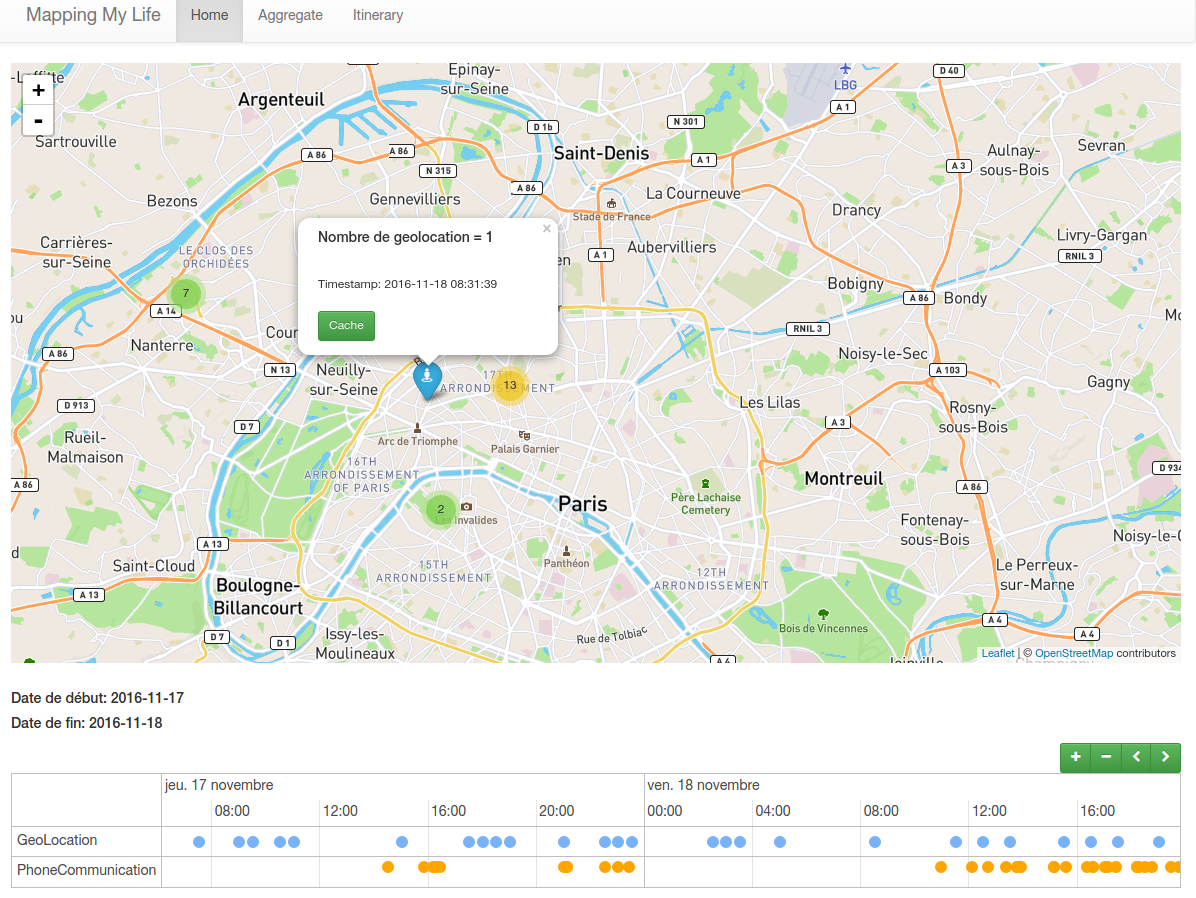
The TimeLineView displays data by date order. It is also a time selector which allows user to choose a certain period time which data will be shown.

The date information can be found just below the map.

For example, you can first click on timeline and then use mouse to zoom in/out or move left/right the timeline, the date time will be chosen according to the start and the end of timeline. At the same time, HomeMap will display also the data according to the date chosen.

In the screen shot below, you can see we chose the date from 2016-11-17 to 2016-11-18.

The pop-up shows geolocation informations of date 2017-11-18.



**2. FavorisPage ( with URL ‘/map’ )**

The FavorisPage component can be found in containers/FavorisPage.js.

This view contains two parts: FavorisMap(components/FavorisMap.js) and FavorisForm(components/FavorisForm.js)

FavorisMap shows the five points which is the most frequent in the database for data type GeoPoint and PhoneCommunicationLog.

FavorisForm allows users to add, modify or delete a point favorite to database. After adding a favorite point, the style of the marker on FavorisMap will be changed according to the type of favorite point.

Here, the fields latitude and longitude are read-only which mean that you can only set values by clicking the marker on the map. The three buttons with different functions can make user to mange a favorite point. After clicking button, it will send a request by using Data API to Cozy and application will receive a response. The response will be displayed to indicate users the result.

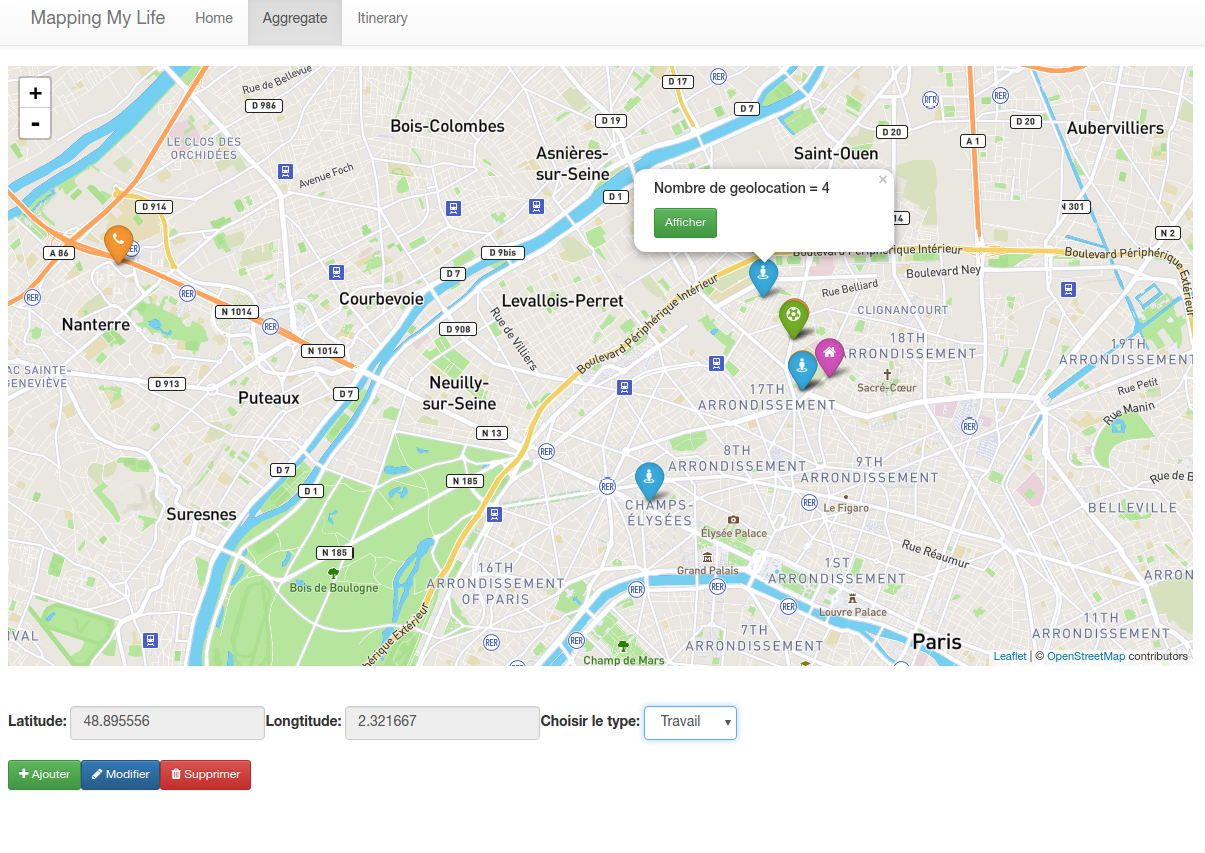
For example, when you click on a marker, the value latitude and longitude will be filled and then you need to select the type of this point like it is your home or work place. You can also customer the type by select ‘Autres’ fields. Once you click on ‘Ajouter’ button, a request will be sent in form of like :

{ “latitude”: 48.890833,

“longitude”: 2.326944,

“category”: “maison” }

and it will be add to FavorisPoint in CouchDB.

**3. ItineraryPage ( with URL ‘/trace’ )**

The ItineraryPage component can be found in containers/ItineraryPage.js.

This view includes two parts: TraceMap (components/TraceMap.js) and Calendar (components/Calendar.js).

TraceMap shows a itinerary which permit users know the movement of location data intuitively. The itinerary consists with the point start, the point end and an arrow moved to show the direction.

Calendar allows users to choose the date which he wants to display the itinerary.

For example, we chose 2016-11-14 by clicking on calendar and then click ‘Display’ button. A itinerary will be shown on the map. The point start is the first location point ( data type GeoPoint) of the day 2016-11-14 in CouchDB and the point end is the last location point ( data type GeoPoint) of 2016-11-14.

## Configuration

## There are several configuration files with different uses.

## package.json : this is the configuration file for the whole application which includes basic information of the application like: the name of application, the version, the scripts commands, and all the dependencies packages needed…

## webpack.config.js: this is the configuration file for Webpack. It is a place to put all of hte configuration, loaders, and other specific information relating to the build. And there is also a config/ folder includes configuration file for webpack.

## Manifest.webapp: this is the configuration file for Cozy. In order to install this application on Cozy, it must includes a manifest file which named manifest.webapp. It need to be placed to the root of the application. The file contains the name of the application, the description of the application and the permissions.

## .babelrc : this is the configuration file for babel.

## Running application

**Run it inside the VM**

In order to run this application on Cozy, we need use cozy-stack docker image which provide an environment allowing to run application on the cozy-stack. ([https://github.com/cozy/cozy-stack/blob/master/docs/client-app-dev.md#with-docker](https://github.com/cozy/cozy-stack/blob/master/docs/client-app-dev.md" \l "with-docker))

So you need first install Docker and then pull cozy-stack docker image.

This image contains the development script and all its dependencies.

In a terminal, run the app in watch mode

$ yarn run watch

In another terminal, run the docker container

$ cd mappingmylife-react-v3

$ docker run --rm -it -p 8080:8080 -p 5984:5984 -v "$(pwd)/build":/data/cozy-app -v "$HOME/db":/usr/local/couchdb/data -v "$HOME/storage":/data/cozy-storage --name=cozydev cozy/cozy-app-dev

Your app is available at http://app.cozy.tools:8080.

You can check your data on CoucheDB at http://localhost:5984

**Build application**

Build the application for production mode to the build/ folder.

$ npm run build

or

$ yarn build

You can find all the files needed to be deployed in build/ folder.