

# CAS-DFA 2020 – PROJET

Ufuk Arslan - 18.10.2020

<https://github.com/UfukArslan/AngularProject>

**Travel log**

## TABLE DES MATIERES

<i>Problématique</i>	3
<i>Exigence</i>	3
<b>Couche 1</b>	<b>4</b>
<i>Description</i>	4
<i>Créer un voyage (Créer une vignette)</i>	5
<i>Vignette</i>	6
<i>Vignette (Trois points)</i>	7
<i>Filtre</i>	8
<b>Couche 2</b>	<b>9</b>
<i>Formulaire</i>	10
<i>Map</i>	11
<i>Vignette</i>	12
<i>Vignette (Trois points)</i>	13
<i>Filtre</i>	14
<b>Couche 3</b>	<b>15</b>
<b>Diificultés</b>	<b>16</b>

## *Problématique*

Le travail que j'ai réalisé se base sur une interface « utilisateur » et « une expérience riche côté front-end » qui se nomme *TravelLog*. Cette fonctionnalité permet aux utilisateurs de créer des voyages et d'y ajouter des lieux. Chaque voyage permet d'avoir une description et une photo de couverture.

## *Exigence*

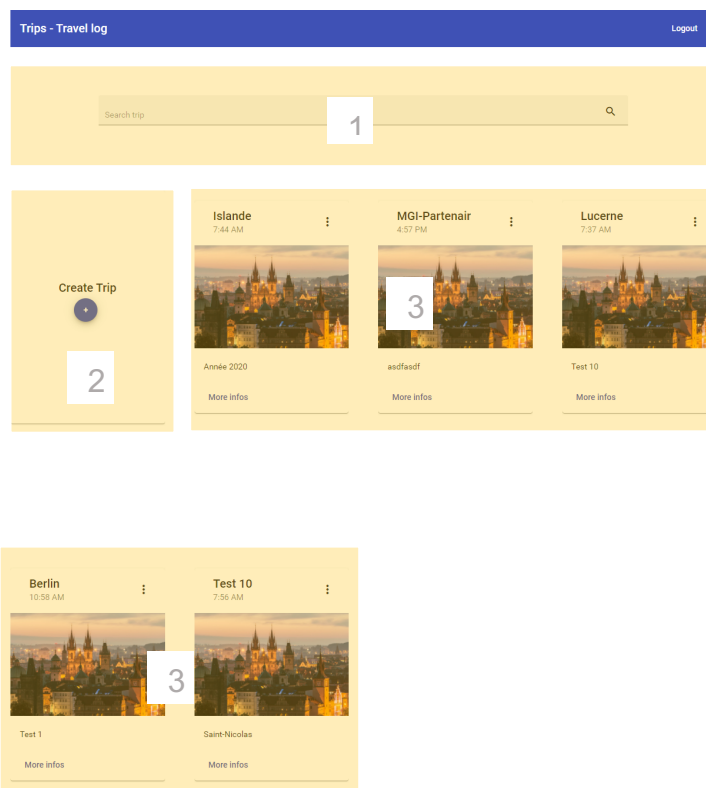
**L'application permet à l'utilisateur de:**

- **enregistrer** un nouveau compte et de se connecter / se déconnecter
- **créer** un voyage avec un titre et une description
- **ajouter** des lieux à un voyage, à un endroit précis, avec un titre, une description et une photo
- **gérer** ses voyages / lieux créés (mise à jour / suppression)
- **voir** une liste de voyages et accéder aux détails d'un voyage, y compris la liste de ses lieux
- **voir** les lieux sur une carte pour un ou tous les voyages et voir les détails de ces lieux
- **filtrer** les lieux et / ou les voyages pour ne voir que certains d'entre eux quelque part dans l'application
- **rechercher** des lieux et / ou des voyages quelque part dans l'application
- **publier** des commentaires sur des lieux et voir les commentaires quelque part dans l'application

# Couche 1

## Description

Mon application est divisée en trois couches principales. La première couche (Trips) permet principalement de créer des voyages à travers des vignettes. La deuxième couche (Places) comporte le formulaire qui récolte les informations pour déterminer un lieu qui sera affiché également dans une vignette. La dernière catégorie (EditPlace) sert à éditer les données que l'on a insérées dans la deuxième couche.



*Trips - Première Couche 1*

1. Filtre
2. Créer un voyage
3. Vignette

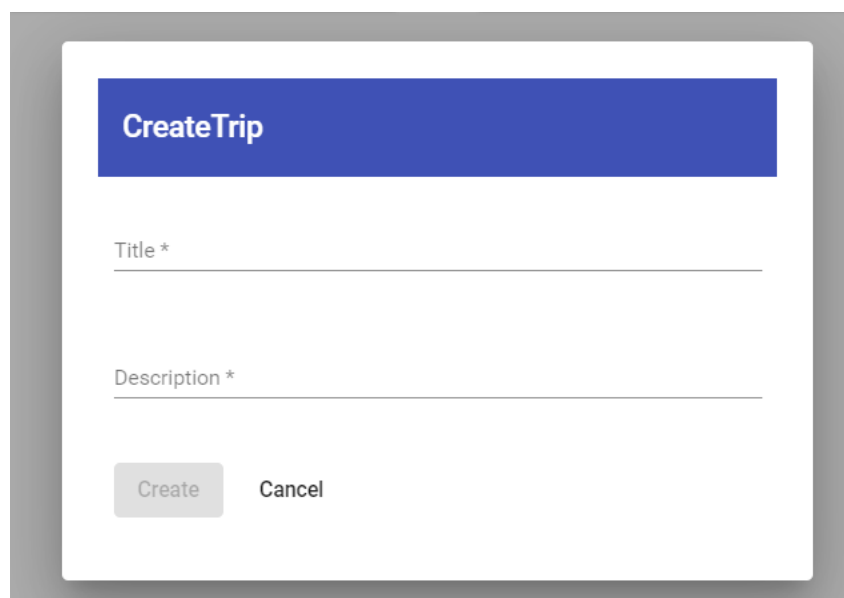
### *Créer un voyage (Créer une vignette)*

Le bouton «Create Trip » va générer une popup afin de récupérer les informations pour le titre et la description du voyage.

Create Trip

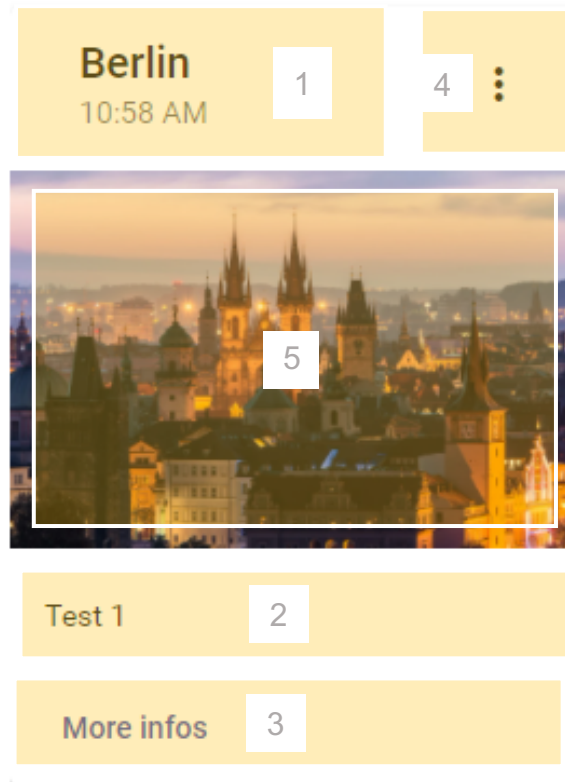


*CreateTrip - Button*



*CreateTrip - Popup*

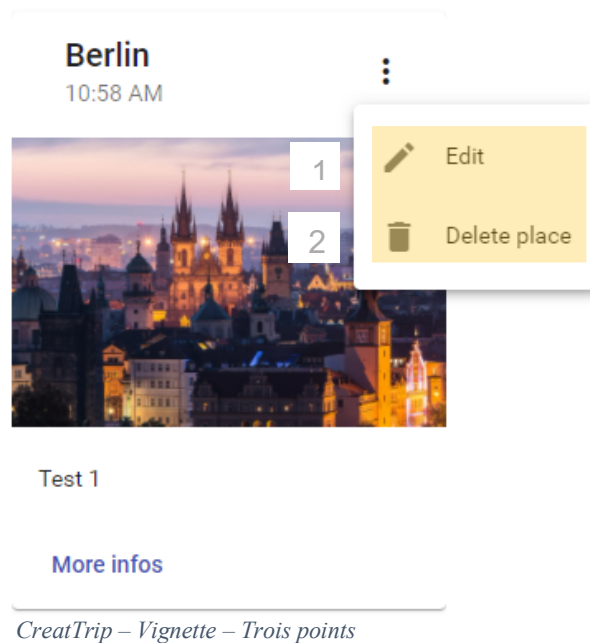
## Vignette



CreateTrip - Vignette

1. Titre
2. Description
3. Button *More infos* permet d'accéder à la deuxième couche (Places)
4. Les trois points ouvrent une fenêtre contextuelle qui permet d'éditer ou de supprimer la vignette.
5. Une image qui représente le voyage

## Vignette (Trois points)

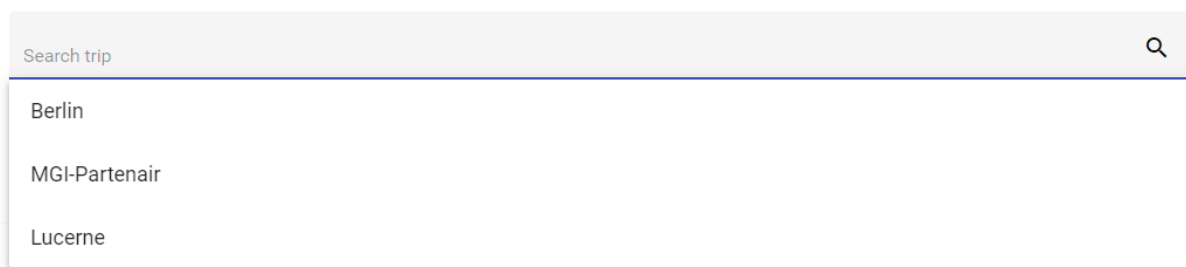


1. *Edit* génère une popup qui nous invite à porter des modifications sur le titre et la description du voyage.
2. *Delete place* permet de supprimer la vignette.

The screenshot shows the 'Edit Trip' popup form. It has a blue header with the title 'Edit Trip'. Below the header are two text input fields labeled 'Title \*' and 'Description \*'. At the bottom are two buttons: 'Edit' (blue) and 'Cancel' (gray).

*EditTrip - Popup*

## Filtre



The image shows a search bar with the placeholder text "Search trip" and a magnifying glass icon. Below the search bar, a dropdown menu is open, displaying a list of travel suggestions: "Berlin", "MGI-Partenair", and "Lucerne".

*CreateTrip - Filtre*

La barre qui se trouve au-dessus des vignettes donne la possibilité de filtrer les voyages qui ont été insérés dans l'application par les utilisateurs. En cliquant dans l'espace grisé, une liste des voyages disponibles sur l'application est proposée dans un format liste.

De plus, en cliquant sur l'une des propositions, cette dernière va afficher le voyage sous une vignette.



## Couche 2

La deuxième couche donne accès à une interface qui permet de souscrire des informations afin de pouvoir créer un emplacement dans le voyage que l'on effectue.

The screenshot shows the 'Create Places' interface in the 'Trip - Islande' application. The interface is divided into four numbered sections:

- 1**: A form for entering place details. It includes a 'Name place \*' field, a 'Description' field, a 'Location' field, and a 'Done' button.
- 2**: A map showing the location of the place. The map is a topographic map of Iceland, with a blue pin indicating the location of the place.
- 3**: A thumbnail image of the place. The image shows a cityscape with a church spire, and the text 'Lac myvtan' and '7:45 AM' is visible above it.
- 4**: A search bar for finding places. It includes a 'Search place' label and a magnifying glass icon.

*Create Places - Couche 2*

1. Le formulaire comporte les champs pour le nom de la place, la description, les coordonnées qui seront ensuite enregistrées dans la base de données.
2. La map donne la possibilité de sélectionner un emplacement. Ce dernier va s'enregistrer automatiquement dans le formulaire.
3. Une vignette est générée avec les informations récupérées depuis le formulaire.
4. Filtre

## Formulaire

Le formulaire nous invite dans un premier temps à entrer un nom concernant l'emplacement où l'on se trouve. Une fois le champ validé, le point numéro deux va à son tour nous inviter à insérer une description et le dernier point permet d'obtenir les coordonnées de la place.

Ces informations auront un nombre de caractère à respecter (de trois à quatorze caractères afin de respecter la taille de la vignette). Tant que ces derniers ne sont pas valides, un message d'erreur s'affichera et le bouton *Next* ou *Save* seront inactifs (grisés).

### 1 Fill out your name place

Name place \*

Name must be between 3 and 14 characters

Next

### 2 Fill out your description

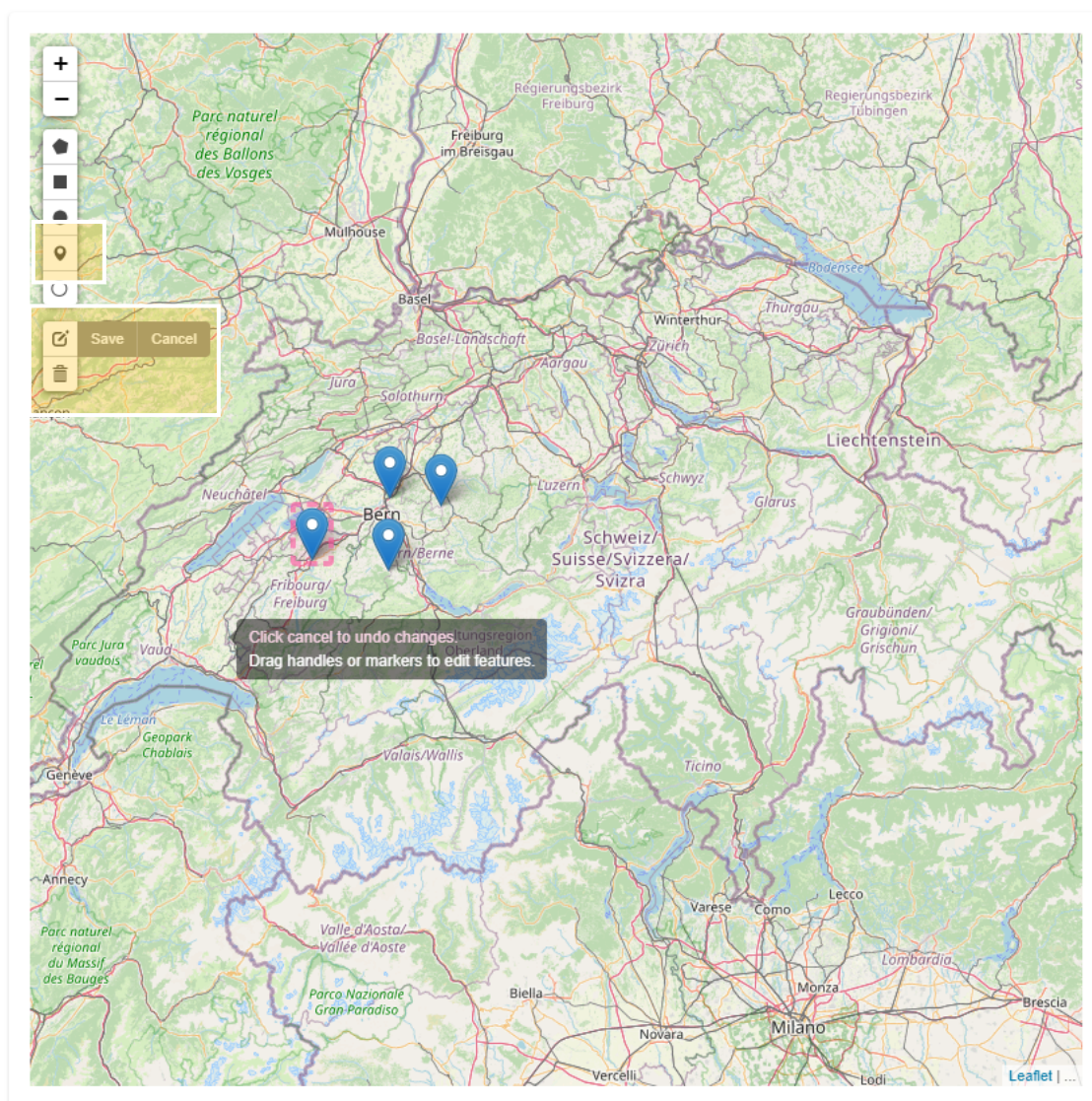
### 3 Add location since the map

### 4 Done

Create Places - Formulaire

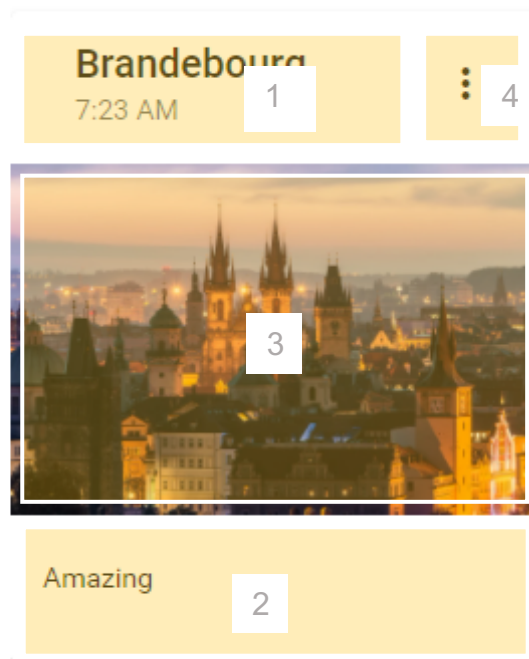
## Map

Une barre d'outils se trouve en haut à gauche de la map. Cette barre permet de sélectionner le marqueur afin de définir une position sur la carte. Les coordonnées du marqueur seront transférées directement dans la formulaire. De plus, l'outil *EditLayer* accorde la modification de l'emplacement du marker. Le *DeleteLayer*, quant à lui, sert à supprimer le maker.



CreatePlace - Map

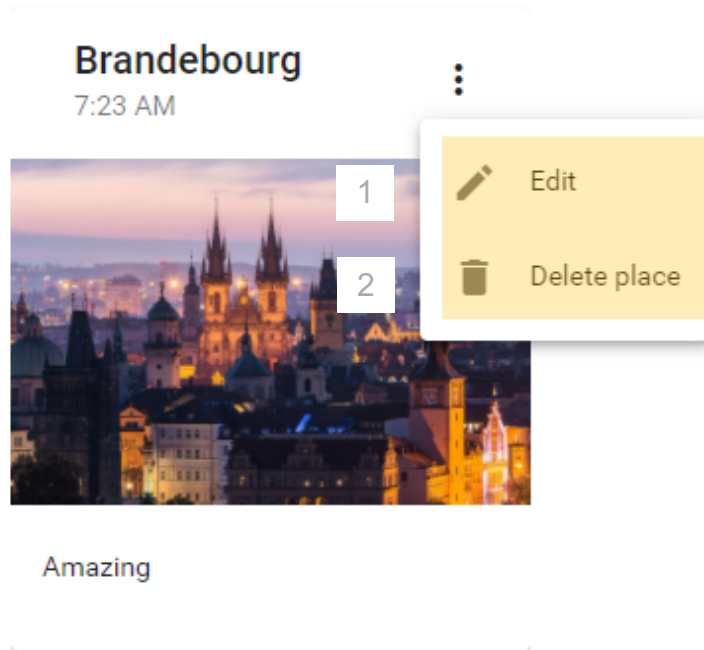
## Vignette



CreatePlace - Vignette

1. Titre
2. Description de le place
3. L'image décrit l'emplacement
4. Les trois points ouvrent une fenêtre contextuelle qui permet d'éditer ou de supprimer la vignette.


### *Vignette (Trois points)*



*CreatePlace – Vignette – Trois points*

1. *Edit* accède à la couche trois qui permet de redéfinir le titre, la description ainsi que l'emplacement du marker.
2. *Delete place* permet de supprimer la vignette.

## Filtre



Search trip

- Berlin
- MGI-Partenair
- Lucerne

*CreatePlace - Filtre*

La barre qui se trouve au-dessus des vignettes donne la possibilité de filtrer les emplacements qui ont été insérés dans l'application par les utilisateurs. En cliquant dans l'espace grisé, une liste des places qui sont disponibles sur l'application sera proposée dans un format liste. De plus, en cliquant sur l'une des propositions, ce dernier va afficher l'endroit sous une vignette.



## Couche 3

L'interface pour éditer une place est similaire à celle de la création d'une vignette pour un emplacement. Cependant, les champs du formulaire sont préalablement remplis avec les informations existantes (grisé) et doivent être passés avec les nouvelles informations.

Trip - Islande

Edit - Logout

1 Change name

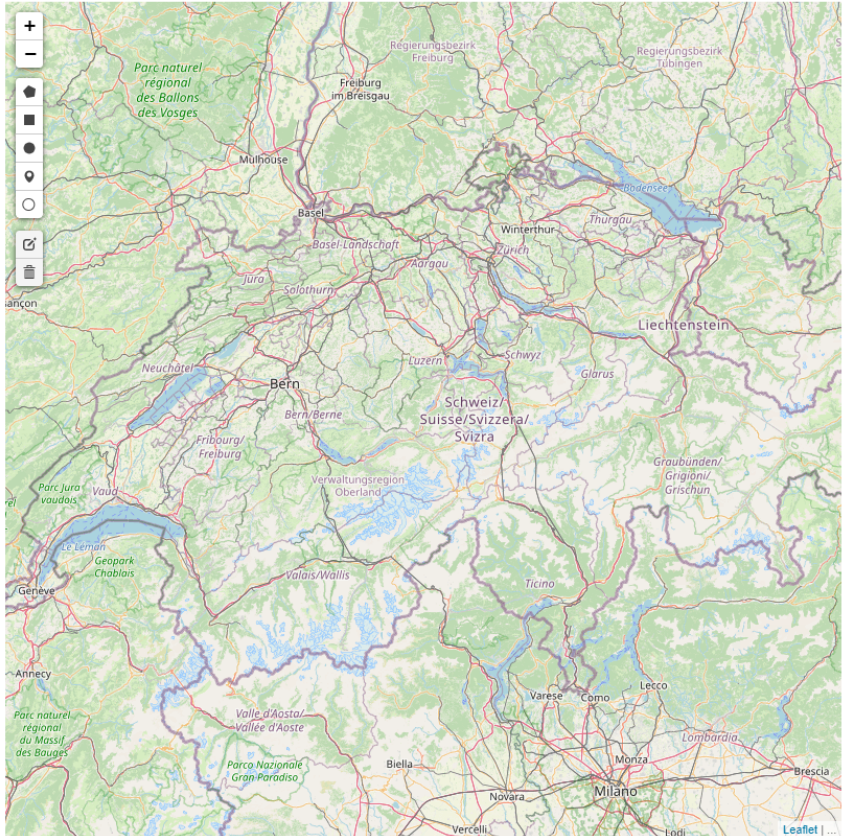
Brandebourg \*

Next

2 Change description

3 Change location

4 Done



*EditPlace*

## Difficultés

Les difficultés rencontrées durant l'élaboration de ce projet ont été dans un premier temps, de comprendre comment faire interagir les composants entre eux.

Le premier service qui m'a permis d'établir cette interaction entre deux composants a été un simple *assesseur*. Je me suis dirigé ensuite sur la technologie de *RXJS* afin de profiter de quelques fonctionnalités utiles (ex : ajouter une vignette sans rafraîchir la page).

Après avoir vu comment faire pour passer d'un composant à un autre, la deuxième partie a été de comprendre comment faire passer *la donnée* (tel que l'id) à travers ces composants. Il y a avait les *composants enfants* où l'on passait les informations avec `@Input` et d'autres données à travers des services.

J'ai utilisé la technologie *Leaflet* pour la localisation de la map. Elle m'a paru abstraite tout au long du projet.

Les fonctions des filtres se sont révélées être de plus en plus compliquées.

Et pour finir, par manque de temps, la base de donnée pour les images n'a pas été établie.