

# **Compte Rendu**

## **Mini-Projet Rails : "Le Carnet de Voyages"**

### ***AVET Tanguy WebTech N3***

#### **Étapes du Projet**

##### **Étape 1 : Configuration de Base**

**Créer une nouvelle application Rails.**

Pour cette étape, j'ai uniquement exécuter la commande rails new Carnet\_De\_Voyage

**Créer un modèle User avec les attributs : nom, email.**

Pour cette demande, j'ai préféré utiliser rails generate scaffold User nom:string email:string

Le fait d'utiliser scaffold m'a permis d'éviter de créer un à un le model, le controller et la vue cependant si j'aurais voulu créer un à un les éléments cités ci-dessus j'aurais exécuter les commandes suivantes :

```
rails generate model User nom:string email:string
```

Ensuite j'aurais fais mes migrations

```
rails generate controller User index
```

**Créer un modèle Trip (Voyage) avec les attributs : destination, description, date de début, date de fin.**

Ici j'ai reproduis le même schéma que j'ai fais pour User en utilisant la commande suivante :

```
rails generate scaffold Trip destination:string description:string date_debut:date date_fin:date user:references
```

**Établir une relation one-to-many entre User et Trip**

J'ai ajouter les lignes suivantes dans mes deux model :

Pour user.rb :

```
trip.rb U X schema.rb U user.rb U X
app > models > user.rb
1 class User < ApplicationRecord
2   has_many :trips
3 end
4
```

Pour trip.rb :

```
trip.rb U X schema.rb U user.rb U
app > models > trip.rb
1 class Trip < ApplicationRecord
2   belongs_to :user
3 end
4
```

## Insertion d'un jeu d'essai :

User :

User.create nom:'Alice', email:'[alice@example.com](mailto:alice@example.com)'

User.create nom:'Bob', email:'[bob@example.com](mailto:bob@example.com)'

User.create nom:'Jean', email:'[jean@example.com](mailto:jean@example.com)'

Trip :

Trip.create destination:'Paris', description:'La capitale de la France', date\_debut:  
'2023-11-30', date\_fin: '2023-12-04', user: User.find(1)

Trip.create destination:'Singapour', description:'Une belle ville', date\_debut:'2023-12-04',  
date\_fin: '2024-04-12', user: User.find(2)

## Remarque :

Il y a une aide dans la caisse à outils que je n'ai pas forcément compris, c'est la commande pour les Relations : "user.trips.create(destination: 'Paris')

Dans mon cas lorsque j'essai d'utiliser cette commande j'ai un retour me disant que la méthode trips n'est pas reconnu dans user, j'ai donc associé mes voyages comme ci-dessus en passant un id d'utilisateur à mon voyage

## Étape 2 : Manipulation des Données

### Créer des routes RESTful pour User et Trip.

Pour la création des routes RESTful, j'ai ajouter dans mes routes les lignes suivantes :

```
resources :trips  
resources :users
```

Je me suis rendu compte que ces deux lignes remplacer l'entièreté des routes qu'on aurait pu faire, cela donne donc plus de clarté dans nos fichiers.

J'ai pu m'appuyer sur cette documentation sur les routes afin de savoir ce que faisait "resources :objet" les parties correspondantes sont la partie Restful et la partie Resource Notation

Lien de la documentation des routes et resources :

[https://launchschool.com/books/demystifying\\_rails/read/routes\\_and\\_resources](https://launchschool.com/books/demystifying_rails/read/routes_and_resources)

### Dans le contrôleur Trips, ajouter des actions pour afficher, créer, modifier et supprimer des voyages.

Pour afficher, créer, modifier et supprimer des voyages j'ai procéder de la manière suivante

Pour afficher j'ai voulu afficher la liste entière des voyages mais aussi afficher le détail de chaque voyage

Afficher la liste entière :

```
class TripsController < ApplicationController  
  before_action :set_trip, only: %i[ show edit update destroy ]  
  
  # GET /trips or /trips.json  
  def index  
    @trips = Trip.all  
  end  
end
```

Afficher le détail de chaque voyage :

```

# GET /trips/1 or /trips/1.json
def show
  @trip = Trip.find(params[:id])
  respond_to do |format|
    format.html # show.html.erb
    format.json { render json: @trip.as_json(include: { user: { only: [:id, :nom, :email] } }) }
  end
end

```

Ici j'ai voulu retourner par la même occasion les informations concernant l'utilisateur qui est affilié au voyage, cela donne plus de précision sur qui à réserver quel voyage.

Création d'un voyage :

```

# GET /trips/new
def new
  @trip = Trip.new
end

# POST /trips or /trips.json
def create
  @trip = Trip.new(trip_params)

  respond_to do |format|
    if @trip.save
      format.html { redirect_to trip_url(@trip), notice: "Un nouveau voyage à été créer" }
      format.json { render :show, status: :created, location: @trip }
    else
      format.html { render :new, status: :unprocessable_entity }
      format.json { render json: @trip.errors, status: :unprocessable_entity }
    end
  end
end

```

Ici def new permet d'instancier uniquement un voyage et donc permettre de rendre un visuel sur un formulaire où l'on peut compléter les informations du nouveau voyage et def create sert quant à lui à sauvegarder ce nouveau voyage en base de données avec toutes les informations complétés par l'utilisateur.

Modification d'un voyage :

```

39 # PATCH/PUT /trips/1 or /trips/1.json
40 def update
41   respond_to do |format|
42     if @trip.update(trip_params)
43       format.html { redirect_to trip_url(@trip), notice: "Le voyage à été mis à jour" }
44       format.json { render :show, status: :ok, location: @trip }
45     else
46       format.html { render :edit, status: :unprocessable_entity }
47       format.json { render json: @trip.errors, status: :unprocessable_entity }
48     end
49   end
50 end
51

```

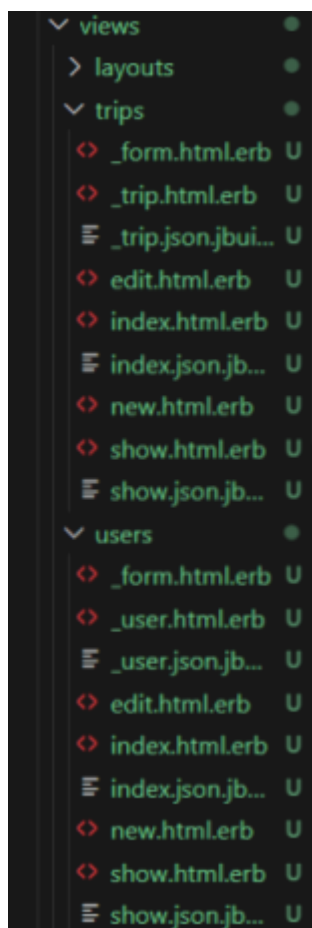
Suppression d'un voyage :

```
# DELETE /trips/1 or /trips/1.json
def destroy
  @trip.destroy!

  respond_to do |format|
    format.html { redirect_to trips_url, notice: "Le voyage à été supprimé" }
    format.json { head :no_content }
  end
end
```

**Créer des vues simples pour ces actions :**

Mes vues se décomposent en plusieurs fichiers, les voici :



Ici, j'ai un dossier trips ainsi qu'un dossier users contenant pour chacun les vues correspondantes. Je vais expliquer leur utilité pour Trip mais le fonctionnement est similaire pour User

Le fichier `_form.html.erb` affiche un formulaire avec les paramètres que prend chacun des objets par exemple pour trip on vient afficher des input pour destination, description, date\_debut, date\_fin, user\_id ainsi qu'un bouton pour soumettre le formulaire. Celui-ci me permet d'effectuer des actions CRUD sur l'objet (création, mise à jour)

Le fichier `_trip.html.erb` affiche les informations concernant mon objet Trip, en l'occurrence je viens afficher destination, description, date\_debut, date\_fin, user\_id mais aussi les informations de l'utilisateur affilié à ce voyage soit son nom d'utilisateur et son email

Le fichier `edit.html.erb` retourne le formulaire `_form.html.erb` avec les informations du voyage que l'on souhaite modifier, en dessous de ce formulaire nous avons une redirection permettant de voir les détails du voyage et une autre redirection pour retourner à la liste des voyages

Le fichier `index.html.erb` retourne la liste complète de tous les voyages existants avec pour chacun une redirection afin d'afficher les détails du voyage. En dessous de cette liste il y a une redirection pour créer un nouveau voyage.

Le fichier `new.html.erb` retourne un formulaire vide où il est possible d'entrer des valeurs correspondantes à la nouvelle instance de Trip, on peut donc créer un nouveau voyage avec

ce formulaire. En dessous on a une nouvelle redirection pour retourner à la liste complète des voyages.

Le fichier show.html.erb affiche les détails d'un voyage, il y a 2 redirections en dessous, une redirection pour modifier ce voyage et une autre pour retourner à la liste des voyages. On a également un bouton en dessous permettant de supprimer le voyage

**\_form.html.erb :**

```
app > views > trips > <_form.html.erb
1  <%= form_with(model: trip) do |form| %>
2  <% if trip.errors.any? %>
3    <div style="color: red">
4      <h2><%= pluralize(trip.errors.count, "error") %> prohibited this trip from being saved:</h2>
5
6      <ul>
7        <% trip.errors.each do |error| %>
8          <li><%= error.full_message %></li>
9        <% end %>
10     </ul>
11   </div>
12 <% end %>
13
14 <div>
15   <%= form.label :destination, style: "display: block" %>
16   <%= form.text_field :destination %>
17 </div>
18
19 <div>
20   <%= form.label :description, style: "display: block" %>
21   <%= form.text_field :description %>
22 </div>
23
24 <div>
25   <%= form.label :date_debut, style: "display: block" %>
26   <%= form.date_field :date_debut %>
27 </div>
28
29 <div>
30   <%= form.label :date_fin, style: "display: block" %>
31   <%= form.date_field :date_fin %>
32 </div>
33
34 <div>
35   <%= form.label :user_id, style: "display: block" %>
36   <%= form.text_field :user_id %>
37 </div>
38
39 <div>
40   <%= form.submit %>
41 </div>
42 <% end %>
43
```

## **\_trip.html.erb :**

```
app > views > trips > <> _trip.html.erb
1  <div id="<%= dom_id trip %>">
2    <p>
3      <strong>Destination:</strong>
4      <%= trip.destination %>
5    </p>
6
7    <p>
8      <strong>Description:</strong>
9      <%= trip.description %>
10   </p>
11
12   <p>
13     <strong>Date debut:</strong>
14     <%= trip.date_debut %>
15   </p>
16
17   <p>
18     <strong>Date fin:</strong>
19     <%= trip.date_fin %>
20   </p>
21
22   <p>
23     <strong>User:</strong>
24     <%= trip.user_id %>
25   </p>
26
27   <p>
28     <strong>Nom d'utilisateur :</strong>
29     <%= trip.user.nom %>
30   </p>
31
32   <p>
33     <strong>Email utilisateur :</strong>
34     <%= trip.user.email %>
35   </p>
36
37 </div>
38
```

## **edit.html.erb :**

```
app > views > trips > <> edit.html.erb
1  <h1>Modifier le voyage</h1>
2
3  <%= render "form", trip: @trip %>
4
5  <br>
6
7  <div>
8    <%= link_to "Voir le voyage", @trip %> |
9    <%= link_to "Retournez à la liste de voyage", trips_path %>
10  </div>
11
```

index.html.erb :

```
app > views > trips > <> index.html.erb
1   <p style="color: green"><%= notice %></p>
2
3   <h1>Voyages</h1>
4
5   <div id="trips">
6     <% @trips.each do |trip| %>
7       <%= render trip %>
8       <p>
9         <%= link_to "Voir le voyage", trip %>
10      </p>
11    <% end %>
12  </div>
13
14  <%= link_to "Nouveau voyage", new_trip_path %>
15
```

new.html.erb :

```
app > views > trips > <> new.html.erb
1   <h1>Nouveau voyage</h1>
2
3   <%= render "form", trip: @trip %>
4
5   <br>
6
7   <div>
8     <%= link_to "Retournez à la liste de voyage", trips_path %>
9   </div>
10
```

show.html.erb :

```
app > views > trips > <> show.html.erb
1   <p style="color: green"><%= notice %></p>
2
3   <%= render @trip %>
4
5   <div>
6     <%= link_to "Modifier ce voyage", edit_trip_path(@trip) %> |
7     <%= link_to "Retournez à la liste de voyage", trips_path %>
8
9     <%= button_to "Supprimer ce voyage", @trip, method: :delete %>|
10  </div>
11
```

## Étape 3 : Requêtes et Affichage

### Créer une page d'accueil listant tous les voyages

Ici l'url de la page d'accueil affichant la liste complète de tous les voyages est la suivante dans mon cas : <http://localhost:3000/trips>

Pour afficher la liste complète de mes voyages j'ai simplement bouclé grâce à un foreach sur ma liste de voyages @trips, cette variable d'instance récupère l'entièreté de mes voyages dans le controller Trips grâce à une requête sur ma base de données



J'affiche ensuite chacun des voyages avec l'utilisateur qui leur est affilié et je leur associe pour chaque voyage un lien afin de rediriger sur le détail du voyage. J'ai également une redirection pour créer un nouveau voyage. Voici mon code :

```
app > views > trips > index.html.erb
1 <p style="color: green"><%= notice %></p>
2
3 <h1>Voyages</h1>
4
5 <div id="trips">
6   <% @trips.each do |trip| %>
7     <%= render trip %>
8     <p>
9       <%= link_to "Voir le voyage", trip %>
10    </p>
11  <% end %>
12 </div>
13
14 <%= link_to "Nouveau voyage", new_trip_path %>
15
```

## Étape 5 : Finitions et Style

### Utiliser CSS pour améliorer l'apparence des pages

(Voir le projet pour la stylisation), par ailleurs il y a quelques choses où je n'ai pas réussi à trouver la réponse, c'est sur le fait que toutes mes stylisations s'attachent à mon fichier css de base (application.css), de ce que j'ai vu cela pouvait provenir du fichier manifest.js où j'aurais du changer mon lien vers mon dossier stylesheet et au lieu de mettre link\_directory mettre link\_tree avec les noms de fichiers css mais dans mon cas ca n'a pas fonctionner !

### Points Bonus

#### Formulaires de création de Trip

```
app > views > trips > new.html.erb
1 <%= stylesheet_link_tag 'new' %>
2 <section class="new-container">
3   <h1 class="new-title">Nouveau voyage</h1>
4
5   <%= render "form", trip: @trip %>
6
7   <br>
8
9   <div class="new-button">
10    <%= link_to "Retournez à la liste de voyage", trips_path %>
11  </div>
12 </section>
```

Voici mon formulaire de création de Trip, ici j'ai une section contenant mon titre de formulaire, un rendu de mon formulaire en lien avec Trip puis j'ai une redirection sur ma liste de voyage.

Et voici le rendu de ce formulaire de création :

## Nouveau voyage

Destination

---

Description

---

Date debut

jj/mm/aaaa



Date fin

jj/mm/aaaa



User

---

Create Trip

Retournez à la liste de voyage

## Conclusion

En conclusion, ce TP m'a permis de découvrir le framework Ruby On Rails. J'ai pu avoir un aperçu de sa puissance, et je trouve ce framework très intéressant, car il permet d'aborder rapidement des problématiques complexes tout en conservant un code assez simple à comprendre. Ce TP m'a également beaucoup plu, car c'était une manière de s'initier au framework sans avoir l'impression de travailler. De mon côté, je vais continuer d'approfondir mes connaissances sur cette technologie, et je me questionne maintenant sur l'idée d'utiliser ce framework pour mon projet CDA !