



Dev Documentation

Voyage Voyage - End Project - Coding Academy by Epitech

Vladimira GROSSEAN - Samira TAIBI - Fanny THIBERT

=> A website like "TripAdvisor"

I - Dev part

Create backend :

Using expressjs/nodejs :

```
mkdir API_voyages
cd API_voyages
npm init -y
npm install express
```

Create folders and file for API routes :

```
mkdir routes
cd routes
touch files.js...

mkdir config
cd config
touch db.js => enter parameters to access to the database
```

Install dependencies :

```
npm i nodemon
npm i axios
npm i bcrypt
npm i body-parser
npm i cookie-parser
npm i cors
npm i crypto-js
npm i debug
npm i dotenv
npm i express
npm i fs

npm i http-errors
npm i jsonwebtoken
npm i morgan
npm i multer
npm i mysql
npm i nodemailer
npm i pug
npm i swagger-jsdoc
npm i swagger-ui-express

=> do the same for all APIs
```

tests with mocha chai

```
npm install --save mocha chai
mkdir test

touch 'files-test'.js...
npm run test
```

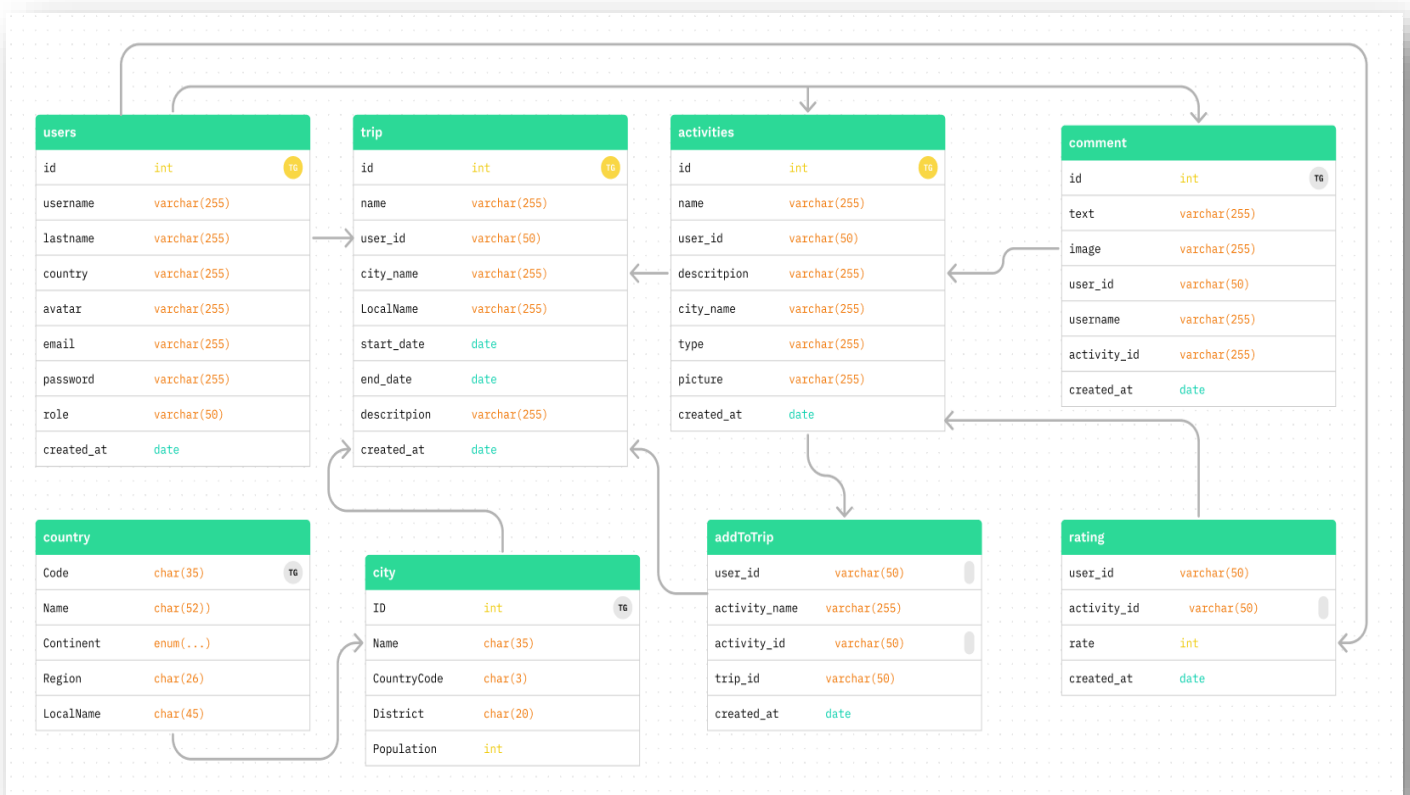
Using mysql :

```
sudo /etc/init.d/mysql start  
sudo mysql
```

- > CREATE DATABASE voyage
- > USE voyage
- > CREATE TABLES...

⇒ use db [world](#) for countries and cities

UML Diagram :



```
app.js  
config  
├─ db.js  
├─ config.json  
├─ package-lock.json  
├─ package.json  
├─ routes  
│   ├── CountryCity.js  
│   ├── activities.js  
│   ├── addToTrip.js  
│   ├── allcity.js  
│   ├── cities.js  
│   ├── cityActivity.js  
│   ├── comment.js  
│   ├── commentActiv.js  
│   ├── countries.js  
│   ├── lang.js  
│   ├── login.js  
│   ├── myActivity.js  
│   ├── myTrips.js  
│   ├── rating.js  
│   ├── register.js  
│   ├── trip.js  
│   ├── updateUser.js  
│   └── users.js  
└─ test  
    ├── database-entry.js  
    ├── login.js  
    ├── trip.js  
    └── users.js  
voyage.sql
```

API documentation :

swagger : <http://localhost:3020/api-docs/#/>

Activities ^	
GET	/activities v
POST	/activities v 🔒
GET	/activities/{id} v
DELETE	/activities/{id} v
GET	/activities/{city_name} v
AddToTrip ^	
GET	/addToTrip/{id} v
DELETE	/addToTrip/{id} v
POST	/addToTrip v
Cities ^	
GET	/allcity/{CountryCode} v
GET	/cities v
GET	/cities/{id} v
Comments ^	
GET	/comment v
POST	/comment v 🔒
GET	/comments/{activity_id} v
DELETE	/comment/{id} v
Countries ^	
GET	/countries v
GET	/countries/{id} v
CountryCity ^	
GET	/countryCity v
Countrylanguage ^	
GET	/lang v
GET	/lang/{id} v
Authentication ^	
POST	/login v
POST	/register v
my Trips ^	
GET	/myTrips v
Ratings ^	
GET	/rating v
POST	/rating v 🔒
GET	/rating/{activity_id} v
Trips ^	
GET	/trip v
POST	/trip v 🔒
GET	/trip/{id} v
PATCH	/trip/{id} v 🔒
DELETE	/trip/{id} v
Users ^	
PATCH	/users/{id} v 🔒
GET	/users/{id} v
DELETE	/users/{id} v
GET	/users/a/{id} v
GET	/users v

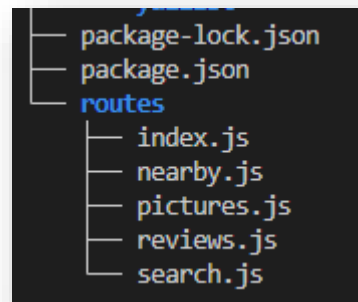
API Tripadvisor

use Tripadvisor API :

<https://www.tripadvisor.com/developers>

```
mkdir TripAdvisor_API
cd TripAdvisor_API
npm init -y
npm install express

mkdir routes
cd routes
touch files.js...
```



```
├── package-lock.json
├── package.json
├── routes
│   ├── index.js
│   ├── nearby.js
│   ├── pictures.js
│   ├── reviews.js
│   └── search.js
```

Create frontend :

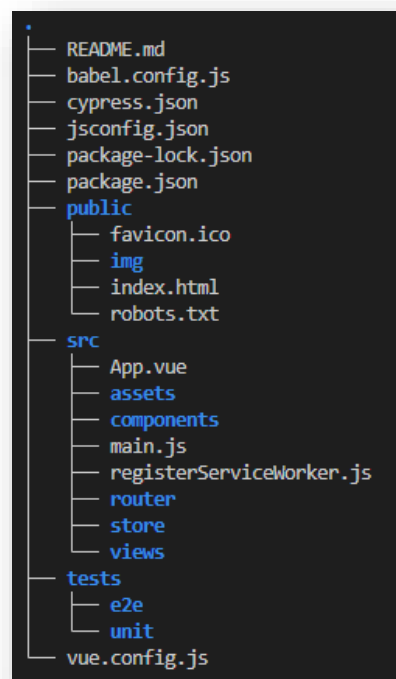
Create a vue.js app :

```
npm install -g @vue/cli
vue create front-v
=> select your options

cd front-v
npm install
npm run serve
```

Install dependencies :

```
npm i bootstrap
npm i bootstrap-vue
npm i ol
npm i mdb-vue-ui-kit
npm i parcel
npm i pug
npm i register-service-worker
npm i vue
npm i vue-axios
npm i vue-router
npm i vue3-openlayers
npm i vuex
```



```
├── README.md
├── babel.config.js
├── cypress.json
├── jsconfig.json
├── package-lock.json
├── package.json
├── public
│   ├── favicon.ico
│   ├── img
│   ├── index.html
│   └── robots.txt
├── src
│   ├── App.vue
│   ├── assets
│   ├── components
│   ├── main.js
│   ├── registerServiceWorker.js
│   ├── router
│   ├── store
│   └── views
├── tests
│   ├── e2e
│   └── unit
└── vue.config.js
```

Create folders and files for the front :

```
mkdir 'folders'...components/views/router...
touch 'files'...
```

To manage uploading photos : use an image download website : <https://fr.imgbb.com/>