adcleek
Partenaire Innovation du Retail

Test technique

Objectif

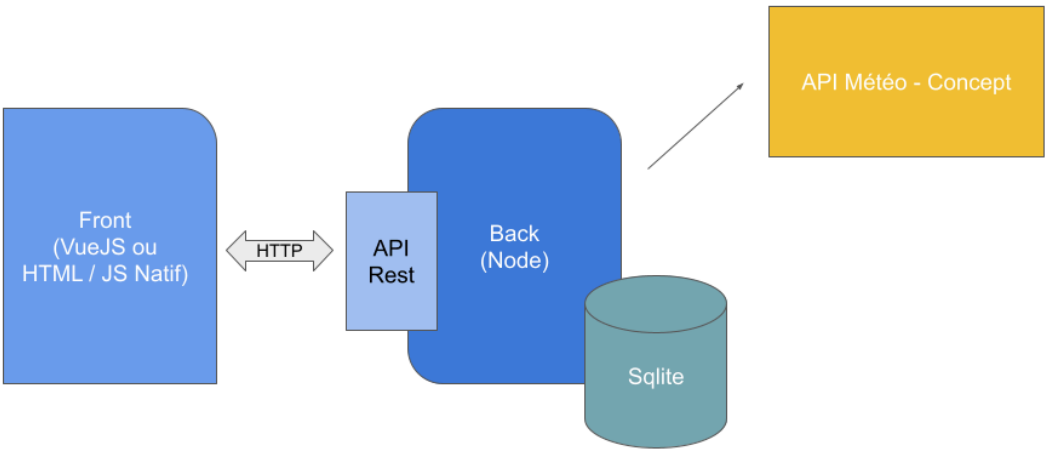
L'objectif est de développer une application qui permet d'afficher des informations de météo à partir d'une liste de villes.
L'application se décompose en une partie front-end et une partie back-end.

Architecture

La partie front-end sert d'interface entre l'utilisateur et l'application back-end. Cette partie du code est exécutée côté navigateur.

La partie back-end a pour rôle d'accéder aux données et de les proposer sous forme d'une API Rest à la partie front-end. Ce module doit aussi aller chercher les informations de météo sur une API de météo externe si on ne les a pas en base de données.

Le stockage des données se fait dans une base de données sqlite. Le code pour y accéder est dans le fichier Database.ts / database.js.

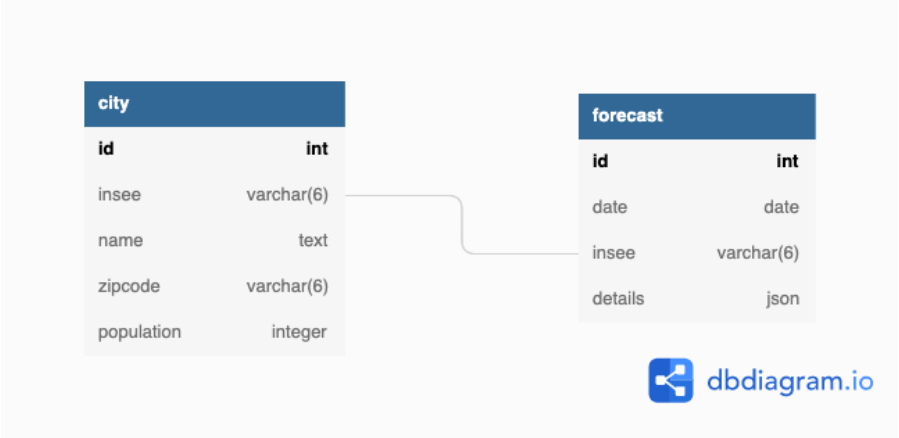


Modèles de données

La base est constituée de deux tables :

- city : la liste des villes
- forecast : la prévision de la météo à une date donnée pour une ville donnée. L'ensemble des données de prévision sont stockées dans *details* sous forme d'une structure JSON.

C'est le code insee de la ville qui fait office d'identifiant unique dans les deux tables.



Accès API météo

Les données de météo sont à récupérer sur l'API de météo : *Météo Concept* : https://api.meteo-concept.com/documentation. Un token est fourni pour pouvoir se connecter à l'API.

Back-end

Le back-end est à développer en node. Il est recommandé d'utiliser express. Il faut proposer une API Rest composée de trois routes :

- Liste des villes: retourne la liste des villes et leurs informations
- Prévision météo: retourne les prévisions météos pour une ville donnée à partir de son code INSEE
- Ajouter une ville à partir de son code INSEE.

De manière générale, dans le retour des routes, on ne veut pas voir apparaître l'id des tables en base (exemple pour une city on veut uniquement insee, name, zipcode et population)

Route liste des villes :

Cette route renvoie un tableau de la liste des villes présentes en base avec leur code insee, leur nom, le code postal et la population.

Route des prévisions de météo :

On doit d'abord regarder en base si on a l'information demandée, si c'est le cas la retourner, si ce n'est pas le cas aller la chercher sur l'API météo, la stocker en base et la retourner. On peut par exemple utiliser la librairie *axios* pour requêter l'API météo.

L'API météo renvoie les prévisions pour plusieurs jours, on souhaite uniquement garder les prévisions pour le jour même et pour le lendemain. On souhaite aussi avoir :

- La somme des prévisions de pluie sur l'ensemble des jours retournés par l'API de météo.
- La moyenne de la température min et max sur l'ensemble des jours retournés par l'API de météo
- Le nombre de jours qui ont permis de faire ces calculs.

Route d'ajout d'une ville :

Cette route doit permettre à partir d'un code INSEE d'ajouter une nouvelle ville en base. On ne passe que le code INSEE et on récupère le maximum d'informations pour insérer en base via l'API météo.

Si la ville existe déjà en base, on ne l'ajoute pas.

Fichiers fournis :

Deux versions du projet sont disponibles: en TypeScript ou en js.

Le projet contient les fichiers:

- Database.ts / database.js :
  - run : fonction pour exécuter une requête sans retour de données (create, insert, update …)
  - all : récupère un tableau de données suite à une requête (select par exemple)
  - get : récupère une donnée suite à une requête (select sur un élément)
  - init : initialise la base de données avec les bonnes tables. Il faut garder son appel au lancement de l'application (déporté dans InitDb.ts dans la version TypeScript)

- WeatherUtils.ts / weatherUtils.js : contient une fonction qui permet d'obtenir un nom d'icône cohérent pour un code weather de l'API météo.

Il est nécessaire de laisser la ligne initDb.*initialize() / db.init* présente dans le fichier server.ts / index.js (elle sert à initialiser les tables de la base et à peupler la table city).

Il est nécessaire de mettre en place du “cors” pour pouvoir communiquer entre le front et le back. La ligne permettant de le faire est en commentaire dans le fichier server.ts / index.js.

Attention l’échappement du caractère “ (double quote) avec *sqlite* se fait en le doublant.

Front-end

Le front-end peut être développé au choix en VueJS, ReactJS ou en HTML / Javascript natif.
Il est recommandé d’utiliser la librairie *Bootstrap* ou tout autre framework.
Les noms d’icône pour les prévisions renvoyés par la fonction présente en back sont ceux de bootstrap.

Il doit ressembler à la maquette présente en fin de document (l’apparence peut être adaptée au framework).
Lorsque l’on clique sur une ligne du tableau de gauche, on affiche les informations de prévision météo de la ville sélectionnée dans les 4 cadres de droite.

La liste des villes est chargée au démarrage de la page via l’API Rest du back. Les prévisions météo sont récupérées au moment du clic via l’API Rest du back.

En plus de ce qui est présent sur la maquette, il faut ajouter un champ pour saisir un code INSEE et un bouton “ajouter”, pour permettre d’ajouter une ville à la liste à la fois côté back et côté front.

Dans le cas où vous développez le front en JS / HTML natif vous pouvez servir vos fichiers HTML à partir de node (voir lignes en fin du fichier index.js du back).

Bonus

Si il reste du temps vous pouvez ajouter au choix les fonctionnalités suivantes :

- Ajouter une barre de recherche au tableau
- Ajouter une pagination au tableau
- Trier le tableau par chacune des colonnes

Code Insee	City	Population
06029	CANNES	74152
06088	NICE	342636
10387	TROYES	60641
13001	AIX-EN-PROVENCE	143006
13208	MARSEILLE 8E ARRONDISSEMENT	80725
13209	MARSEILLE 9E ARRONDISSEMENT	74521
13212	MARSEILLE 12E ARRONDISSEMENT	60800
13213	MARSEILLE 13E ARRONDISSEMENT	91758
13214	MARSEILLE 14E ARRONDISSEMENT	62199
13215	MARSEILLE 15E ARRONDISSEMENT	76419
14118	CAEN	105400
17300	LA ROCHELLE	75735
18033	BOURGES	65557
21231	DIJON	155095
25056	BESANÇON	116465
26362	VALENCE	62475
29019	BREST	139339
29232	QUIMPER	63401
2A004	AJACCIO	69073
30189	NÎMES	150999
31555	TOULOUSE	475442
33063	BORDEAUX	252044
33281	MÉRIGNAC	70318
33318	PESSAC	61860
34032	BÉZIERS	76495
34172	MONTPELLIER	281611
35238	RENNES	216268
37261	TOURS	136565
38185	GRENOBLE	158182

Probabilité de pluie

-

50 %

Min **5 °C**

Max **15 °C**

Probabilité de pluie

-

80 %

Min **1 °C**

Max **11 °C**

Probabilité de pluie

-

10 %

Min **9 °C**

Max **18 °C**

Probabilité de pluie

-

33 %

Min **7 °C**

Max **20 °C**