

Collège Ahuntsic
Département d'informatique

420-289-AH
Programmation web côté
serveur
hiver 2022

TP1

Développement d'une application météorologique

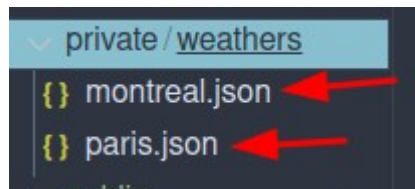
Énoncé

Vous êtes engagé par la firme MétéoWiki pour développer une application permettant d'accéder aux conditions météorologiques de différentes villes. Cette application sera rendue sur le serveur (server side rendering) à l'aide de Handlebars et de Node.js.

Travail à faire

Étape 1: Création des templates [10 points]

La première étape de la création du site web consiste à créer les templates .hbs pour votre application. Dans le gabarit fourni par MétéoWiki, Express est configuré pour utiliser le dossier *templates/views* pour les vues et *templates/partials* pour les partiels. Pour vous simplifier la vie, le service *src/services/staticWeather.service.ts* a été créé. Il permet de simuler un appel à l'API de wttr.in. Ce service retourne sous format JSON le contenu de fichier .json disponible dans le dossier *private* seulement deux villes sont disponibles: *montreal.json* et *paris.json*.



Ce JSON est disponible depuis le controller *src/controllers/weather.controller.ts*. Ce controller est configuré pour recevoir toutes les routes commençant par */weather*.

```
18     router.use(async (req: Request, res: Response) => {
19         const json = await this._weatherService.readWeather(this._defaultLocation);
20         console.log(json);
21         res.render('index');
22     });
23
```

Dans *src/app.ts* la route racine (/) est redirigée vers la route */weather*. Assurez-vous de configurer correctement vos dossiers statiques dans ce fichier (voir le */TODO*)

```
//La route par défaut est /weather
this.app.get('/', (req, res) => {
    res.redirect('/weather');
});
```

Les onglets suivants devront être implémentés:

- **Conditions actuelles:** Onglet résumant les conditions actuelles d'une ou de plusieurs villes (voir le champ *current_condition*).
- **Prochaines heures:** Onglet résumant les conditions pour les prochaines. Chaque champ *weather* contient un champ *hourly*. Ce champ *hourly* indique les prévisions à une certaine heure. L'heure est disponible sous le champ *time* qui représente une certaine heure sous le format [militaire](#).
- **Prochains jours:** Onglet résumant les conditions moyennes des prochains jours. Utilisez les champs dans *weather*
- **Astronomie:** Onglet résumant les informations en lien avec l'astronomie pour les prochains jours. Utilisez le champ *astronomy* dans *weather*. Cet onglet doit se trouver sous la route */astronomy* et non */weather*. Vous devrez donc ajouter un nouveau *controller*.

On désire également pouvoir visualiser les informations de chaque onglet pour plusieurs villes en même temps dans le but de comparer plus facilement les informations. Pour cela, la barre de recherche permettant d'entrer le nom de la ville doit pouvoir supporter **plusieurs villes séparées par des virgules**. Un exemple d'interface est disponible en annexe.

Étape 2: Connexion à l'API de wttr.in [5 points]

La deuxième étape consiste à faire le lien avec l'API. Cette communication doit être faite dans le service *src/services/wttrWeather.service.ts*. Ce service implémente la même interface que le service *src/services/staticWeather.service.ts*, il pourra donc le remplacer facilement une fois fonctionnel.

```
readWeather(location: string): Promise<JSON> {  
  // TODO Extraire le JSON à l'aide du service wttr  
  console.log(location);  
  throw new Error('Method not implemented.');
```

Pour cela, il suffit de modifier le binding dans le fichier *src/inversify.config.ts* (voir le TODO).

```
5 import { Server } from '../server';  
6 // TODO Utiliser wttr comme provider au lieu des fichiers statiques  
7 import { StaticWeatherService as weatherProvider } from '../services/staticWeather.service';  
8 // import { wttrWeatherService as weatherProvider } from '../services/wttrWeather.service';  
9
```

Le module `inversify` permet l'injection des dépendances dans notre application ([inversify](#)). La documentation sur l'API de `wtrr.in` est disponible sur leur [Github](#). Voici deux exemples d'utilisation:

wtrr.in/montreal?format=j1	Retourne la météo sous format .json pour la ville de Montréal.
wtrr.in/@github.com?format=j1	Retourne la météo sous format .json pour la position de l'adresse ip associée à github.com.
wtrr.in/~kilimanjaro?format=j1	Retourne la météo sous format .json d'une position près du mont Kilimanjaro.

Assurez-vous que votre application supporte ces trois types de requêtes et qu'il soit possible de les combiner avec des virgules.

Étape 3: Intégration de MongoDB [5 points]

Cette étape consiste à utiliser MongoDB pour augmenter les performances de votre application en conservant le résultat des requêtes faites par l'utilisateur. Vous devez donc:

- Sauvegarder le résultat de chaque requête faite par un utilisateur dans MongoDB.
- Lorsque l'utilisateur fait une requête présente dans MongoDB, vérifiez que la date et le temps de l'observation sont toujours pertinents (on ne veut pas retourner des données qui ne sont plus d'actualité). Si les données sont assez récentes, les retourner, sinon assurez-vous de les mettre à jour.

La complexité de MongoDB devra être encapsulée dans le service `src/services/mongodb.service.ts`. Utilisez une base de données nommée ***tp1*** et une collection nommée ***weathers***.

Évaluation

Pour chacune des étapes, les critères suivants seront évalués:

- Qualité du code
 - Respect des règles du linter (*npm run lint*)
 - Documentation du code
- Fonctionnalité
 - Respect des exigences
 - Dépassement des attentes
- Appréciation globale
 - UI (interface utilisateur)
 - UX (expérience utilisateur)
 - Effort

Remise

La remise de votre code se fera sur LÉA sous le format .zip avant le 23 janvier 23h59. Assurez-vous de ne pas inclure les dossiers **out** et **node_modules** au .zip. Si le temps le permet, une présentation des projets aura lieu en classe.

Annexe



MétéoWiki

Condition actuelle

Prochaines heures

Prochain jour

Astronomie

Trois-Rivieres, Quebec, Canada

Température: -4°C (ressentie
-12°C)

Conditions: Clear

Vents: 6Km/h WNW

Visibilité: 0 km

UV: 1

Humidité: 99%

Précipitations: 0.1 mm

Dernière observation: 2022-01-09 07:06 PM
(heure locale)

Greenwich Village, New York,
United States of America

Température: 4°C (ressentie 1°C)

Conditions: Light rain

Vents: 19Km/h WSW

Visibilité: 16 km

UV: 1

Humidité: 76%

Précipitations: 0.0 mm

Dernière observation: 2022-01-09 07:15 PM
(heure locale)