Projet

Application météo

Amine Bohi, Florian Bridoux, Nathan Lhote et Julie Parreaux

Contents

1 Le projet **1**
- 1.1 Application Météo . . . 1
  - 1.1.1 Description du projet . . . 1
  - 1.1.2 Objectif . . . 2
- 1.2 Contraintes de développement . . . 2
- 1.3 Attentes du travail à réaliser . . . 2
  - 1.3.1 Travail par itérations . . . 3
- 1.4 Ce dont vous disposez pour ce projet . . . 3
  - 1.4.1 Données au format JSON . . . 3

2 Expression fonctionnelle des besoins **3**
- 2.1 Manipulation des données . . . 3
- 2.2 Gestion des données . . . 3
- 2.3 Des interfaces . . . 4
  - 2.3.1 Interface en ligne de commande . . . 4
  - 2.3.2 Interface graphique . . . 4

3 Les livrables **4**
- 3.1 Lot 1 . . . 5
  - 3.1.1 Modèle . . . 5
  - 3.1.2 Vue/Contrôleur . . . 5
- 3.2 Lot 2 . . . 5
  - 3.2.1 Modèle . . . 5
  - 3.2.2 Vue/Contrôleur . . . 5
- 3.3 Lot 3 . . . 5
- 3.4 Lots supplémentaires . . . 5

A Exemple de fichier JSON **6**

B Exemple de GUI **7**

1 Le projet

1.1 Application Météo

1.1.1 Description du projet

Lors de ce projet, vous aurez l'occasion de participer à l'implémentation d'un logiciel de prévision météo.

Vous devrez utiliser Open Weather Map qui met à votre disposition une API permettant de récupérer des informations météorologiques mondiales.

1.1.2 Objectif

Pour mener à bien ce projet, vous travaillerez en groupes de 4 (ou 3/5 si autorisé préalablement) en utilisant les outils et concepts vus en cours afin de vous organiser et de gérer votre travail du mieux possible. Vous aurez également besoin de faire des recherches de librairies adaptées à votre problème.
Le but est d'avoir un aperçu des contraintes imposées par le monde professionnel.

1.2 Contraintes de développement

Les contraintes concernant le développement qui vous sont imposées sont les suivantes:

- Travail en groupe sur un projet en Java;
- Utilisation d'un gestionnaire de source (git en l'occurrence);
- Réaliser des tests unitaires;
- Respecter le design pattern Modèle-Vue-Contrôleur;
- Une structuration de l'interface graphique est imposée par le cahier des charges (peut être modifiée après avoir discuté avec le client !).

1.3 Attentes du travail à réaliser

Votre travail sera évalué selon plusieurs critères:

- lots qui ont été réalisés;
- respects des méthodologies de travail;
- bonne couverture du code par les tests.

Lors des différentes de séances de TP, vous pourrez poser des questions à vos enseignants. Ces questions pourront être de deux natures différentes:

- **Mode client**: questions sur le cahier des charges (ajout d'une fonctionnalité non demandée, incompréhension d'une fonctionnalité demandée, détaillé un point particulier, ...)
- **Mode enseignant**: questions sur la méthodologie de travail (est-ce que la répartition du travail est bonne, aider à choisir entre plusieurs librairies, question sur la structuration générale du code, avis sur les tests implémentés ou à implémenter, ...)

Si votre question est du type "comment lire/ouvrir/analyser un fichier/lien en Java ?" ou "comment utiliser la librairie X ?":

- Utilisez ce qui a été vu dans d'autres UE;
- Lisez des documentations;
- Faites des recherches sur des moteurs de recherche;
- Suivez des tutoriels (soit mis à disposition dans le cadre de l'UE, soit trouvés sur Internet)

1.3.1 Travail par itérations

Chaque lot étant divisé en plusieurs "Jalon", vous aurez à chaque séance de TP à vous distribuer des tâches permettant de valider ce dernier avant la prochaine séance.
Vous pourrez ainsi remplir une fiche de planification, vous permettant d'établir un planning prévisionnel pour la semaine à venir, définissant les tâches de chacun des membres de votre groupe.
Vous devrez donc progresser d'au moins un Jalon d'une semaine sur l'autre.

1.4 Ce dont vous disposez pour ce projet

Pour vous aider dans la réalisation de cette application, il vous sera fourni les contenus suivants:

- De données au format JSON contenant des données météo permettant de se passer de requêtes à l'API dans un premier temps;
- D'une base de projet Java Gradle

1.4.1 Données au format JSON

JSON est un format de données textuelles structurées, tout comme le XML par exemple.
Ces données seront récupérées grâce à l'API du site Open Weather Map. Leurs manuels d'utilisation offrent des explications détaillées sur la manière de réaliser des requêtes et sur la forme de données récupérées.
Un exemple de fichier JSON produit par Open Weather est présenté en annexe A. Ce fichier a été généré à partir d'une requête de la forme suivante: `api.openweathermap.org/data/2.5/weather?q=london&appid={APIkey}`
Dans les fichiers produits, on retrouve toutes les informations nous intéressant.

2 Expression fonctionnelle des besoins

2.1 Manipulation des données

Une des conditions de réussite de votre projet tient du fait que le programme doit être capable d'effectuer les actions de manipulation de données suivantes:

- Récupération des données depuis Open Weather Map
- Lecture de fichiers au format JSON
- Sauvegarde des paramètres de l'utilisateur

2.2 Gestion des données

Le besoin principal est d'afficher des prévisions météo. En sélectionnant une date et un lieu, les données de prévision de température, précipitation, couverture nuageuse (*etc*) doivent s'afficher.
L'utilisateur doit également pouvoir selectionner des villes dans une liste de "favoris". Les données météo des villes dans les favoris doivent être visibles en temps réel.

2.3 Des interfaces

Le logiciel doit pouvoir être utilisé en ligne de commande (en mode dégradé) et avec une interface graphique.
Les interfaces détaillées ici devront très probablement être ajustées afin de permettre l'incorporation de toutes les fonctionnalités demandées ! Voyez avec le client (l'enseignant) dans ces cas là !

2.3.1 Interface en ligne de commande

L'interface en ligne de commande (CLI) doit permettre une utilisation basique de l'application. Elle doit permettre l'exécution des commandes suivantes:

- Afficher les prévisions météo à un lieu et une date donnés en argument (données instantanées si pas de date et lieu(x) par défault si pas de lieu)
- Modifier le(s) lieu(x) par défaut.

2.3.2 Interface graphique

L'interface graphique (GUI) doit permettre une utilisation complète de l'application. Un exemple d'interface possible est présent en annexe B. L'interface à créer est divisée en 3 sections:

1. **Zone d'affichage**

   La section principale doit afficher les prédictions météo de la dernière requête effectuée.

   Les différentes données doivent être clairement lisibles.

2. **Villes favorites**

   Cette section doit afficher les données instantanées des villes inscrites sur la liste des favoris.

   La section doit également contenir un moyen de supprimer une ville de la liste des favoris.

3. **Les champs de requête**

   Cette dernière section contient les champs où l'utilisateur pourra entrer un lieu et une date.

   La section doit aussi permettre à l'utilisateur d'ajouter la ville courante aux favoris.

3 Les livrables

Les lots devant être livrés sont au nombre de 3. Chacun d'entre eux est découpé en Jalons.

3.1 Lot 1

3.1.1 Modèle

- **Jalon 1**: Lecture des données en format JSON.
- **Jalon 1**: Implémentation des fonctionnalités autour de l'affichage de données.
- **Jalon 2**: Gestion de la liste de favoris.

3.1.2 Vue/Contrôleur

- **Jalon 2**: Affichage de données en ligne de commande.

3.2 Lot 2

3.2.1 Modèle

- **Jalon 3**: Utilisation de l'API pour récupérer les données.

3.2.2 Vue/Contrôleur

- **Jalon 4**: Affichage de données dans une interface graphique.

3.3 Lot 3

- **Jalon 5**: Implémentation d'un ou deux (ou plus suivant votre avancement) lots supplémentaires.

3.4 Lots supplémentaires

- **Pictogrammes**: affichage graphique de pictogrammes: soleil, nuage, pluie *etc*
- **Météo heure par heure**: affichage des prévisions heure par heure pour une ville (par exemple sur une journée).
- **Météo jour par jour**: affichage des prévisions jour par jour pour une ville, sur plusieurs jours (par exemple une semaine).
- **Navigation**: Navigation "jour suivant", "jour précédent", *etc*
- **Courbe de températures/précipitations** Possibilité d'afficher des courbes de températures sur une journée, des maximales sur plusieurs jours, des précipitations, *etc*
- **Autre idée à proposer au client**

A Exemple de fichier JSON


```
{
  "coord": {
    "lon":-0.1257,"lat":51.5085
  },
  "weather": [
    {
      "id":804,
      "main":"Clouds",
      "description":"overcast clouds",
      "icon":"04d"
    }
  ],
  "base":"stations",
  "main": {
    "temp":272.65,
    "feels_like":268.16,
    "temp_min":272.04,
    "temp_max":273.15,
    "pressure":999,
    "humidity":86
  },
  "visibility":10000,
  "wind": {
    "speed":3.09,
    "deg":90
  },
  "clouds": {
    "all":100
  },
  "dt":1611475110,
  "sys": {
    "type":1,
    "id":1414,
    "country":"GB",
    "sunrise":1611474590,
    "sunset":1611506106
  },
  "timezone":0,
  "id":2643743,
  "name":"London",
  "cod":200
}
```

B Exemple de GUI