Documentation technique

ROYAL BIO

Le Roi du bio

Table des matières

Equipe

- Christopher MARIE-ANGELIQUE
- Rubén LONGÈQUE
- Bastien BALMES
- Cédric-Alexandre PASCAL

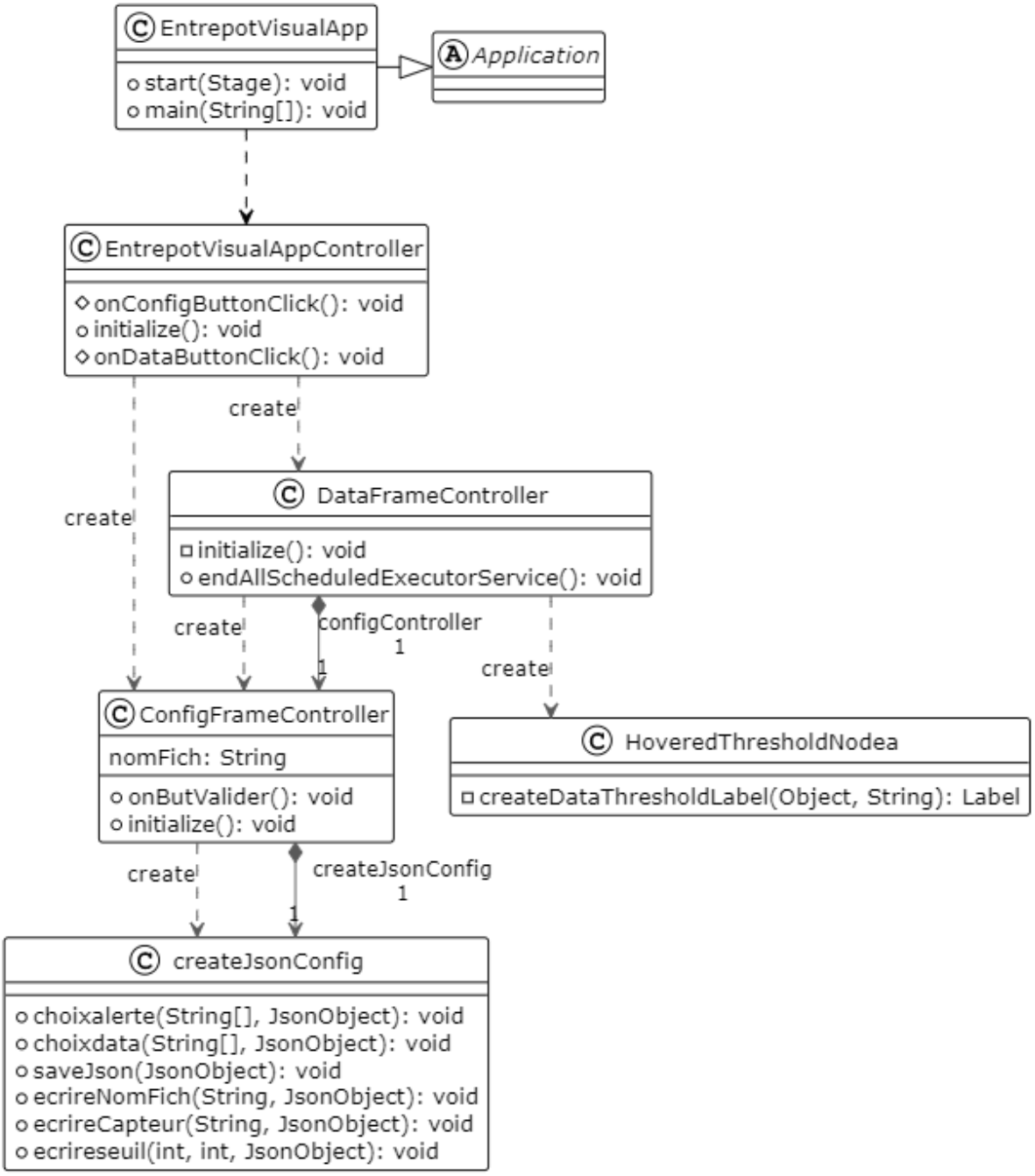
Contexte

Royal Bio souhaite s'étendre sur internet en ouvrant un site web d'e-commerce.
Cette demande est liée avec un changement des mentalités d'un grand nombre de personnes, qui veulent prioriser les aliments bio issus de circuits courts. Une autre raison pousse l'entreprise à se développer sur internet. Des grandes surfaces dans plusieurs villes menacent certaines épiceries de l'entreprise, comme celle de Blagnac et son Leclerc.
L'entreprise souhaite donc gagner en visibilité et augmenter son volume de ventes pour répondre aux besoins des utilisateurs mais aussi pallier les problèmes des hypermarchés.

Présentation de l'application

Cette application est une application permettant de visualiser la température, le taux de CO2 et le taux d'humidité dans l'entrepôt où sont situés les produits vendus par Royal Bio. Elle sert donc à s'assurer que tout se passe bien dans l'entrepôt. Si la température ou le taux d'humidité est trop élevé alors une alerte sera donnée afin d'avertir l'administrateur du problème.

Diagramme de classe



Architecture de l'application

1. Packages

app.g2b11 : Contient toute l'application ainsi que ses contrôleurs
app.g2b11.func : contient les fonctions utilisés par les contrôleurs

2. Rôle de chaque classe par package

app.g2b11:

- **ConfigFrameController** : gère la partie graphique de la fenêtre de modification du fichier de configuration de l'application Python. Affiche les différents champs à cocher, à remplir, à sélectionner.
- **DataFrameController** : gère la partie graphique de la fenêtre d'affichage des données. lis les données du fichier fourni par l'application Python et les affiche dans un graphique.
- **EntrepotVisualApp** : Ouvre la fenêtre principale de l'application
- **EntrepotVisualAppController** : gère la partie graphique de la fenêtre principale de l'application. Peut ouvrir la fenêtre de modification du fichier de configuration et celle d'affichage des données.
- **HoveredThresholdNodea** : Crée un noeud pour le graphique. Si on passe la souris sur ce nœud, il nous affiche la valeur numérique du nœud.

app.g2b11.func :

- **createJsonConfig** : contient toutes les fonctions afin de créer le fichier de configuration.

3. Fonctionnalités

Modification du fichier de configuration

Pour modifier le contenu du fichier de configuration, tout se passe dans *ConfigFrameController.java*. Tout d'abord il affiche les différents capteurs disponibles dans une ListView appelée *lvCapteurs* puis met en place deux Listener sur le slider du seuil maximum de température (*slideSeuilTemp*) et sur le slider du seuil maximum d'humidité (*slideSeuilHum*) . Ensuite une fois que l'utilisateur entre toutes les informations désirées il vérifie si l'utilisateur a bien sélectionné des données, a bien sélectionné un capteur et s'il a bien entré un nom de fichier. Enfin si tout est bon il crée ou modifie le json avec les informations désirés par l'utilisateur sinon un message d'erreur s'affiche avec l'erreur en question.

Affichage des données

Ici l'application récupère les données du fichier .txt et rentre toutes ces données dans un dictionnaire. Ensuite elle récupère l'heure, les minutes et les secondes dans *simpleDateFormat* mais elle récupère aussi le jour et le mois dans *dayDateFormat*. Ensuite elle crée 3 séries, une pour chaque données, donc une série pour entrer les valeurs de température, une autre pour l'humidité et enfin une dernière pour le taux de co2. On ajoute ces séries au graphique actuel si le fichier fourni par le programme Python les contient. Enfin on met en place un scheduler toutes les 10 mins (temps d'actualisation du fichier donné par le programme Python) qui lis le fichier et ajoute les données aux séries et si dans le fichier il est écrit "AlerteTemperature:1.0" le programme Java va afficher une AlertBox

avec le l'heure et la date où l'alerte s'est déclenchée en précisant que le seuil de température est trop élevé, et de même si il est écrit "AlerteHumidite:1.0".

4. Installer le projet dans votre IDE

Libraries

Afin que l'application puisse fonctionner correctement vous aurez tout d'abord besoin d'installer 2 librairies:

- Gson : https://search.maven.org/remotecontent?filepath=com/google/code/gson/gson/2.10/gson-2.10.jar
- Json Simple : https://search.maven.org/remotecontent?filepath=com/google/code/gson/gson/2.10/gson-2.10.jar

Intellij:

Afin d'installer des librairies dans votre projet intellij il va falloir procéder de la manière suivante:

- Faites soit CTRL + ALT + MAJ + S soit cliquer sur File -> ProjectStructure
- Ensuite allez dans l'onglet Modules puis cliquez sur Dependencies
- Enfin cliquez sur le bouton + et sélectionnez "JARs or Directories…" et choisissez le JAR de votre librairie

Eclipse :

Afin d'installer des librairies dans votre projet Eclipse il va falloir procéder de la manière suivante:

- Tout d'abord faites clic droit sur votre projet
- Allez dans propriété puis Java Build Path puis dans Librairies et enfin sélectionnez le JAR de votre librairie