

UML : Unit Tests

isFalty(Service container) :

La méthode *isFalty* de la classe *Sensor* permet de définir si un capteur en particulier est défaillant. Pour se faire, on récupère pour chaque type de mesure la moyenne de notre capteur qu'on compare avec la moyenne de chaque type des autres capteurs à proximité (rayon défini au préalable). Pour ce test, on vérifie pour une série de mesure donnée le capteur qui doit être détecté comme défaillant.

<i>Jeu de test</i>	<i>Objectif du test</i>
Vide	Vérifier que les capteurs ne sont pas marqué défaillants si ils n'ont pas de mesures enregistrées
<pre>2019-01-01 12:00:00;Sensor0;O3;50.0; 2019-01-01 12:00:00;Sensor1;O3;10.0; 2019-01-01 12:00:00;Sensor2;O3;50.0; 2019-01-01 12:00:00;Sensor3;O3;50.0; 2019-01-01 12:00:00;Sensor4;O3;50.0; 2019-01-01 12:00:00;Sensor0;NO2;50.0; 2019-01-01 12:00:00;Sensor1;NO2;50.0; 2019-01-01 12:00:00;Sensor2;NO2;50.0; 2019-01-01 12:00:00;Sensor3;NO2;50.0; 2019-01-01 12:00:00;Sensor4;NO2;10.0;</pre>	Vérifier que les capteurs sont marqués défaillant pour des types de mesures distincts
<pre>2019-01-01 12:00:00;Sensor0;O3;50.0;</pre>	Vérifier que les capteurs n'est pas marqué défaillant si il est le seul dans son type de mesure (pas de comparatif)

getSimilar(timestamp debut, timestamp fin, Service container) :

La méthode *getSimilar* permet aux utilisateurs de noter et classer tous les capteurs en fonction de leur similarité avec un capteur sélectionné. La similarité est déterminée en comparant les données générées par les capteurs au cours d'une période de temps spécifiée. Le but de cette fonctionnalité est d'identifier les zones avec une qualité de l'air similaire.

<i>Jeu de test</i>	<i>Objectif du test</i>
<pre>2019-01-01 12:00:00;Sensor0;O3;10.0; 2019-01-01 12:00:00;Sensor1;O3;15.0; 2019-01-01 12:00:00;Sensor2;O3;20.0; 2019-01-01 12:00:00;Sensor3;O3;25.0; 2019-01-01 12:00:00;Sensor4;O3;30.0; 2019-01-01 12:00:00;Sensor0;NO2;20.0; 2019-01-01 12:00:00;Sensor1;NO2;25.0; 2019-01-01 12:00:00;Sensor2;NO2;30.0; 2019-01-01 12:00:00;Sensor3;NO2;35.0; 2019-01-01 12:00:00;Sensor4;NO2;40.0;</pre>	<i>Vérifier qu'on obtient bien en retour la note et les classement de chaque autre capteur par rapport à notre Sensor0 de référence</i>
<pre>2019-01-01 12:00:00;Sensor0;O3;10.0; 2019-01-01 12:00:00;Sensor1;O3;15.0; 2019-01-01 15:00:00;Sensor1;O3;100.0; 2019-01-01 12:00:00;Sensor2;O3;20.0; 2019-01-01 15:00:00;Sensor2;O3;90.0; 2019-01-01 12:00:00;Sensor3;O3;25.0; 2019-01-01 15:00:00;Sensor3;O3;80.0; 2019-01-01 12:00:00;Sensor4;O3;30.0; 2019-01-01 15:00:00;Sensor4;O3;70.0; 2019-01-01 12:00:00;Sensor0;NO2;20.0; 2019-01-01 12:00:00;Sensor1;NO2;20.0; 2019-01-01 15:00:00;Sensor1;NO2;200.0; 2019-01-01 12:00:00;Sensor2;NO2;30.0; 2019-01-01 15:00:00;Sensor2;NO2;190.0; 2019-01-01 12:00:00;Sensor3;NO2;35.0; 2019-01-01 15:00:00;Sensor3;NO2;180.0; 2019-01-01 12:00:00;Sensor4;NO2;40.0; 2019-01-01 15:00:00;Sensor4;NO2;170.0;</pre>	<i>Vérifier qu'on obtient bien en retour le classement qui dépend de la timezone spécifiée en paramètre ici on prendra entre 12h et 13h. Pour se faire, on prend un classement ordonné dans la timezone et le classement inverse hors de la timezone pour voir si on récupère le bon en sortie.</i>

userParticipation() :

La méthode *userParticipation* gère la participation des particuliers à la génération de données sur la qualité de l'air. Les particuliers sont incités à contribuer aux données en gagnant des points pour l'utilisation de leurs données. Chaque fois que les données de leur capteur sont utilisées dans une requête, ils reçoivent un point. Si un des capteurs d'un utilisateur est marqué comme défaillant (cf. fonction décrite précédemment), celui-ci ne pourra plus gagner de points.

<i>Jeu de test</i>	<i>Objectif du test</i>
<pre>User0;Sensor0; User1;Sensor1; + requête sur Sensor0</pre>	<i>Vérifier qu'on obtient bien les scores des utilisateurs en prenant aucun capteur défaillant</i>
<pre>User0;Sensor0; User1;Sensor1; + requête sur Sensor0</pre>	<i>Vérifier que le score n'évolue pas si un des capteurs est défaillant (Sensor0 défaillant)</i>

calculateQuality(double longi, double lat, Service container) :

La méthode *calculateQuality* a pour objectif de fournir la valeur de la qualité de l'air à une position géographique précise sur le territoire à un moment donné. Elle prend en entrée les coordonnées en tant que paramètres et renvoie les mesures des quatre types de données fournies par le capteur le plus proche de la position spécifiée. De plus, elle renvoie la distance jusqu'à ce capteur.

<i>Jeu de test</i>	<i>Objectif du test</i>
<pre>Sensor0;44;-0.3; Sensor1;50;2; 2019-01-01 12:00:00;Sensor0;O3;10.0; 2019-01-01 12:00:00;Sensor1;O3;100.0; 2019-01-01 12:00:00;Sensor0;NO2;20.0; 2019-01-01 12:00:00;Sensor1;NO2;110.0;</pre>	<i>Vérifier qu'on obtient en sortie les valeurs souhaitées pour la position initiale de (-44, -1). On retourne la moyenne de Sensor0 qui est plus proche.</i>

<pre>Sensor0;44;-0.3; Sensor1;44;-0.3;</pre>	Vérifier qu'on obtient en sortie les valeurs souhaitées pour la position initiale de (-44, -1) avec deux capteurs situés à la même distance de la position initiale
<pre>2019-01-01 12:00:00;Sensor0;O3;10.0; 2019-01-01 12:00:00;Sensor1;O3;100.0; 2019-01-01 12:00:00;Sensor0;NO2;20.0; 2019-01-01 12:00:00;Sensor1;NO2;110.0;</pre>	

meanByZones(Service container) :

La méthode *meanByZones* calcule la valeur moyenne mesurée par tous les capteurs dans chaque zone géographique pour chaque type de données.

<i>Jeu de test</i>	<i>Objectif du test</i>
<pre>Sensor0;44;-0.3; Sensor1;44;-0.3;</pre> <pre>2019-01-01 12:00:00;Sensor0;O3;10.0; 2019-01-01 12:00:00;Sensor1;O3;20.0;</pre>	Vérifier qu'on obtient bien la moyenne des capteurs situés dans une même zone pour un même type
<pre>Sensor0;44;-0.3; Sensor1;44;-0.3;</pre> <pre>2019-01-01 12:00:00;Sensor0;O3;10.0; 2019-01-01 12:00:00;Sensor1;NO2;20.0;</pre>	Vérifier qu'on obtient bien la moyenne des capteurs situés dans une même zone pour des types différents
<pre>Sensor0;44;-0.3; Sensor1;50;2;</pre> <pre>2019-01-01 12:00:00;Sensor0;O3;10.0; 2019-01-01 12:00:00;Sensor1;O3;20.0;</pre>	Vérifier qu'on obtient bien la moyenne des capteurs situés dans des zones différentes pour un même type
<pre>Sensor0;44;-0.3; Sensor1;50;2;</pre> <pre>2019-01-01 12:00:00;Sensor0;O3;10.0; 2019-01-01 12:00:00;Sensor1;NO2;20.0;</pre>	Vérifier qu'on obtient bien la moyenne des capteurs situés dans des zones différentes pour des types différents

impactCleaner() :

La méthode *impactCleaner* facilite l'observation de l'impact des Cleaners d'air fournis par diverses entreprises sur la qualité de l'air comme le rayon d'impact ou le pourcentage moyen d'amélioration.

<i>Jeu de test</i>	<i>Objectif du test</i>
<pre>Sensor0;44;-0.3; //Cleaner Sensor1;45;-0.3; //Dans le rayon Sensor2;46;-0.3; //Hors du rayon 2019-01-01 12:00:00;Sensor0;O3;5.0; 2019-01-01 12:00:00;Sensor1;O3;7.0; 2019-01-01 12:00:00;Sensor2;O3;35.0;</pre>	<i>Vérifier que le rayon d'amélioration est fonctionnel par rapport à la position des autres capteurs pour un type précis</i>
<pre>Sensor0;44;-0.3; //Cleaner Sensor1;45;-0.3; //Dans le rayon Sensor2;46;-0.3; //Hors du rayon 2019-01-01 12:00:00;Sensor0;O3;5.0; 2019-01-01 12:00:00;Sensor1;O3;7.0; 2019-01-01 12:00:00;Sensor2;O3;35.0; 2019-01-01 12:00:00;Sensor2;NO2;1.0; 2019-01-01 12:00:00;Sensor2;NO2;3.0; 2019-01-01 12:00:00;Sensor2;NO2;12.0; 2019-01-01 12:00:00;Sensor2;SO2;3.0; 2019-01-01 12:00:00;Sensor2;SO2;8.0; 2019-01-01 12:00:00;Sensor2;SO2;22.0; 2019-01-01 12:00:00;Sensor2;PM10;0.5.0; 2019-01-01 12:00:00;Sensor2;PM10;1.1.0; 2019-01-01 12:00:00;Sensor2;PM10;5.0;</pre>	<i>Vérifier que le rayon d'amélioration est fonctionnel par rapport à la position des autres capteurs pour la globalité des paramètres de la qualité de l'air</i>

distance(Sensor s2) :

La méthode *distance* est une méthode de la classe Sensor qui permet de calculer la distance séparant deux capteurs.

<i>Jeu de test</i>	<i>Objectif du test</i>
<pre>Sensor0;48.88;2.34; //Paris Sensor1;43.66;1.39; //Toulouse</pre>	<i>Vérifier la distance entre deux capteurs placés stratégiquement pour qu'on puisse vérifier la distance.</i>
<pre>Sensor0;48.88;2.34; Sensor1;48.88;2.34;</pre>	<i>Vérifier que la distance est à 0 si deux capteurs sont à la même position</i>